# Užduotis

Sujungti 𝑚 1 ir 𝑚 2 masių objektai iššaunami vertikaliai į viršų pradiniu greičiu 𝑣 0 . Oro pasipriešinimo koeficientas sujungtiems kūnams lygus 𝑘 . Praėjus laikui 𝑢 , objektai pradeda judėti atskirai. Oro pasipriešinimo koeficientai atskirai judantiems objektams atitinkamyra 𝑘 1 ir 𝑘 2 . Oro pasipriešinimas proporcingas objekto greičio kvadratui. Raskite, kaip kinta objektų greičiai nuo 0 s iki

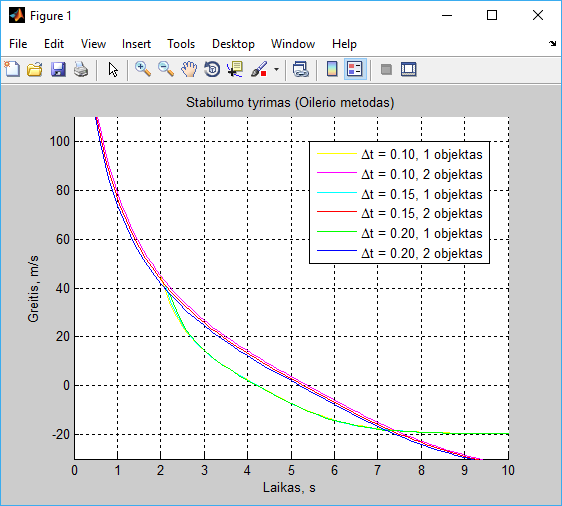
𝑡𝑚𝑎𝑥. Kada kiekvienas objektas pasieks aukščiausią tašką ir pradės leistis?

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Var. Nr. | 𝑚1, 𝑘𝑔 | 𝑚2, 𝑘𝑔 | 𝑣0, 𝑚/𝑠 | 𝑘𝑠, 𝑘𝑔/𝑚 | 𝑡𝑠, 𝑠 | 𝑘1, 𝑘𝑔/𝑚 | 𝑘2, 𝑘𝑔/𝑚 | 𝑡𝑚𝑎𝑥, 𝑠 |
| **5** | 0,8 | 0,8 | 200 | 0,01 | 2 | 0,02 | 0,005 | 15 |

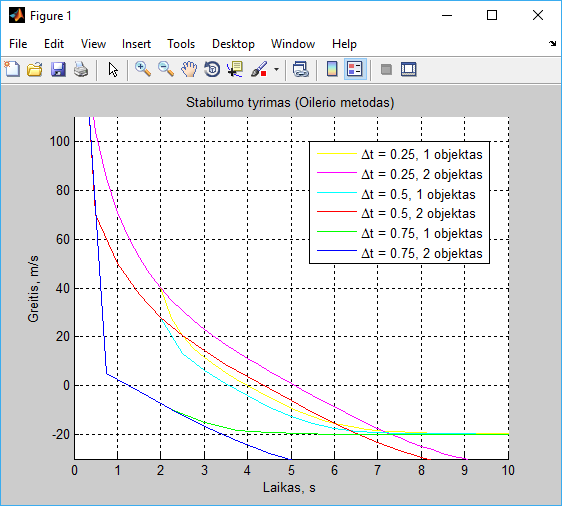
# Sprendimas

Sprendimas Oilerio metodu:

Žingsniai 0.1, 0.15 ir 0.2



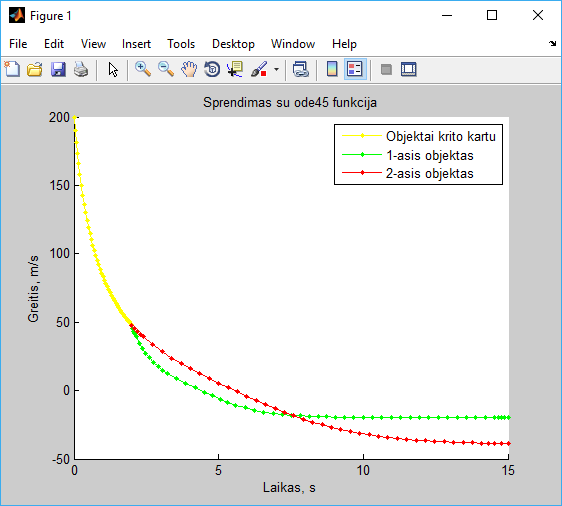
Žingsniai 0.25, 0.5 ir 0.75:



Grafikuose matomas krentančių kūnų greičio kitimas laike. Panaudotas “Oilerio” metodas. Kaip matoma, didesni žingsniai lemia tai, jog metodas pasidaro nebestabilus.

Didžiausias galimas žingsnis, kad metodas išliktų stabiles, yra ~0.25.

Sprendimas su ode45 funkcija:



Grafike matomas krentančių kūnų greičio kitimas laike. Panaudotas “Ode45” metodas.

# Išvados

Laboratorinis darbas padėjo parodyti, kaip “matlab” aplinkoje pavaizduoti Niutono dėsnį veikiančių objektų veiksmus.

Grafikuose matomas krentančių kūnų greičio kitimas laike. Kaip matoma naudojant “Oilerio” metodą, didesni žingsniai lemia tai, jog metodas pasidaro nebestabilus. Didžiausias galimas žingsnis, kad metodas išliktų stabilus, yra ~0.25.

Laboratorinis darbas atliktas naudojant “Oilerio” metodu bei standartine “Ode45” funkcija.

# Programų tekstai

**Oilerio mtodas:**

function oileris

clc, clear all,

close all

spalva=['y'; 'm'; 'c'; 'r'; 'g'; 'b']; %spalvos

m1 = 0.8; % Pirmojo objekto masė (kg)

m2 = 0.8; % Antrojo objekto masė (kg)

v0 = 200; % Pradinis greitis (m/s)

ks = 0.01; % Sujungtų kūnų oro pasipriešinimo koef. (kg/m)

ts = 2; % Judėjimo kartu laikas (s)

k1 = 0.02; % Oro pasipriešinimo koef. 1-am objektui (kg/m)

k2 = 0.005; % Oro pasipriešinimo koef. 2-am objektui (kg/m)

tmax = 15; % Objektų buvimo ore laikas

% kg \* m/s^2

x=0;% pradinės reikšmės x - laikas, y - greitis

step\_variations = 3; % Kiek yra skirtingų žingsnių

dx=[0.25 0.5 0.75]; % žingsnis

%disp(dx(1))

figure(1), hold on, grid on, axis([0,10,-30,110]);

title('Stabilumo tyrimas (Oilerio metodas)')

ylabel('Greitis, m/s')

xlabel('Laikas, s')

set(gca,'XTick', 0:1:15);

plot(x,v0,'ro');

h = 0; % Piesti legendom

hend = 1;

% pntx=0;

% pnty1=v0;

% pnty2=v0;

% v1(1) = v0;

% v2(1) = v0;

for a=1:step\_variations

nsteps=tmax/dx(a); % žingsnių kiekis

pntx=0;

pnty1=v0;

pnty2=v0;

v1(1) = v0;

v2(1) = v0;

x=0;

for i=1:nsteps+1

if(x < ts)

v1(i+1) = v1(i) + dx(a)\*DY((m1+m2), ks, v1(i));

v2(i+1) = v1(i+1);

else

v1(i+1) = v1(i) + dx(a)\*DY(m1, k1, v1(i));

v2(i+1) = v2(i) + dx(a)\*DY(m2, k2, v2(i));

end

plot(x,v1(i),[spalva(2\*a-1),'.'],'MarkerSize',2);

plot(x,v2(i),[spalva(2\*a),'.'],'MarkerSize',2);

h(hend) = plot([pntx,x],[pnty1,v1(i)],[spalva(2\*a-1),'-']);

if (i < 2)

hend = hend+1;

end

h(hend) = plot([pntx,x],[pnty2,v2(i)],[spalva(2\*a),'-']);

if (i < 2)

hend = hend+1;

end

pntx = x;

x=x+dx(a);

pnty1 = v1(i);

pnty2 = v2(i);

end

end

%hh = h([1, 3, 5]);

%legend('labs', 'krabs', 'maps', 'Location', 'Best');

legend(h([1 2 3 4 5 6]),'\Deltat = 0.10, 1 objektas',...

'\Deltat = 0.10, 2 objektas',...

'\Deltat = 0.15, 1 objektas',...

'\Deltat = 0.15, 2 objektas',...

'\Deltat = 0.20, 1 objektas',...

'\Deltat = 0.20, 2 objektas',...

'Location', 'Best');

return

function dy=DY(m, k, v)

g = 9.8; % Laisvojo kritimo pagreitis

if (v < 0)

dy = -(m.\*g - k.\*v.^2)/m;

else

dy = -(m.\*g + k.\*v.^2)/m;

end

return, end

end

**Ode45 metodas:**

function ode

clc, clear all,

close all

global m m1 m2 v0 v1 v2 ks k1 k2;

m1 = 0.8; % Pirmojo objekto masė (kg)

m2 = 0.8; % Antrojo objekto masė (kg)

m = m1 + m2;

v0 = 200; % Pradinis greitis (m/s)

ks = 0.01; % Sujungtų kūnų oro pasipriešinimo koef. (kg/m)

ts = 2; % Judėjimo kartu laikas (s)

k1 = 0.02; % Oro pasipriešinimo koef. 1-am objektui (kg/m)

k2 = 0.005; % Oro pasipriešinimo koef. 2-am objektui (kg/m)

tmax = 15; % Objektų buvimo ore laikas

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% sprendimas ode45 funkcija

[T,X]=ode45(@DY,[0 ts], v0);

v1 = X(end);

v2 = v1;

[T1,X1]=ode45(@DY1,[ts tmax], v1);

[T2,X2]=ode45(@DY2,[ts tmax], v2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(1); hold on;

plot(T,X,'y.-');

plot(T1,X1,'g.-');

plot(T2,X2,'r.-');

title('Sprendimas su ode45 funkcija')

ylabel('Greitis, m/s')

xlabel('Laikas, s')

legend('Objektai krito kartu', '1-asis objektas', '2-asis objektas')

function dy=DY(t, v)

g = 9.8; % Laisvojo kritimo pagreitis

if (v < 0)

dy = -(m.\*g - ks.\*v.^2)/m;

else

dy = -(m.\*g + ks.\*v.^2)/m;

end

return, end

function dy1=DY1(t, v)

g = 9.8; % Laisvojo kritimo pagreitis

if (v < 0)

dy1 = -(m1.\*g - k1.\*v.^2)/m1;

else

dy1 = -(m1.\*g + k1.\*v.^2)/m1;

end

end

function dy2=DY2(t, v)

g = 9.8; % Laisvojo kritimo pagreitis

if (v < 0)

dy2 = -(m2.\*g - k2.\*v.^2)/m2;

else

dy2 = -(m2.\*g + k2.\*v.^2)/m2;

end

end

end