# Užduotis

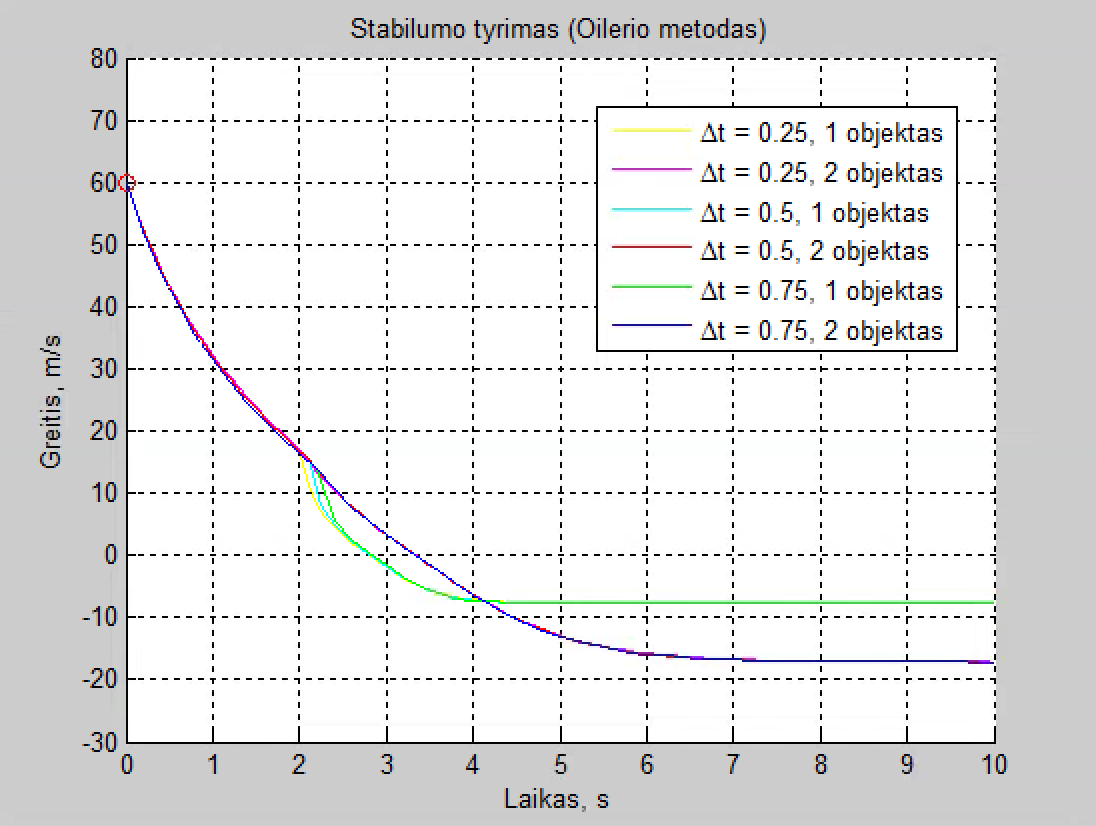
Sujungti 𝑚 1 ir 𝑚 2 masių objektai iššaunami vertikaliai į viršų pradiniu greičiu 𝑣 0 . Oro pasipriešinimo koeficientas sujungtiems kūnams lygus 𝑘 . Praėjus laikui 𝑢 , objektai pradeda judėti atskirai. Oro pasipriešinimo koeficientai atskirai judantiems objektams atitinkamai yra 𝑘 1 ir 𝑘 2 . Oro pasipriešinimas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

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

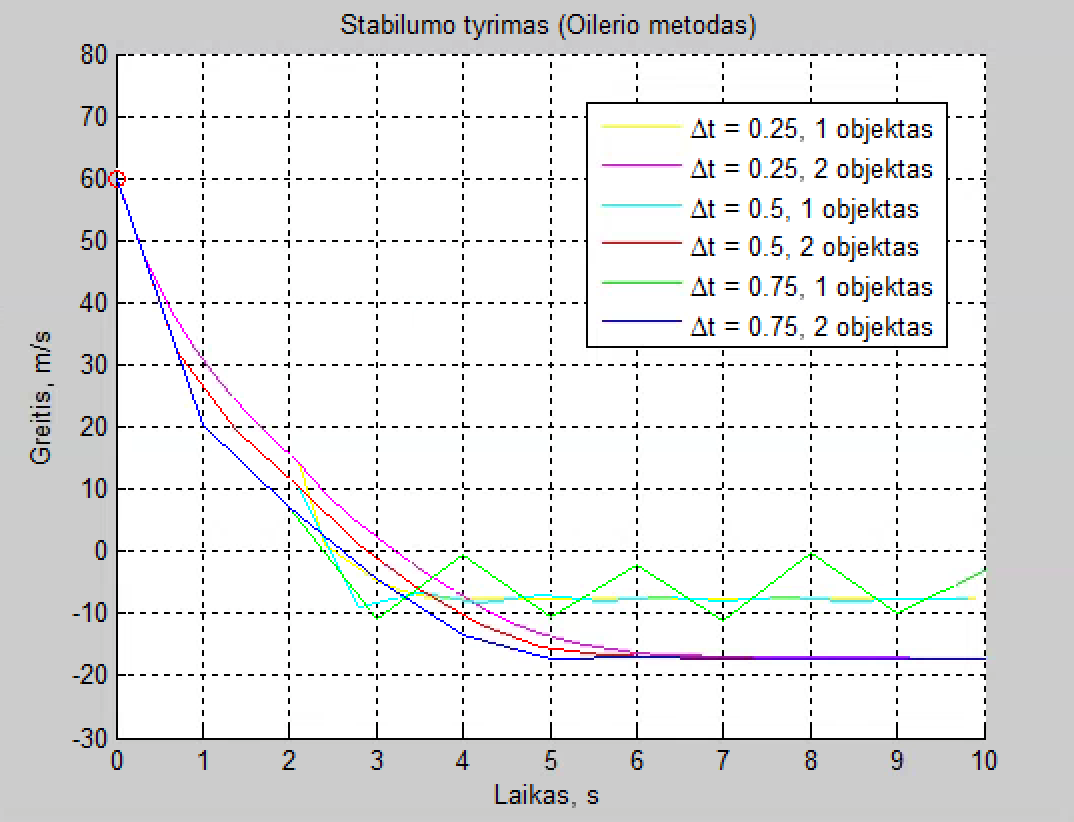
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Var. Nr. | 𝑚1, 𝑘𝑔 | 𝑚2, 𝑘𝑔 | 𝑣0, 𝑚/𝑠 | 𝑘𝑠, 𝑘𝑔/𝑚 | 𝑡𝑠, 𝑠 | 𝑘1, 𝑘𝑔/𝑚 | 𝑘2, 𝑘𝑔/𝑚 | 𝑡𝑚𝑎𝑥, 𝑠 |
| **2** | 0,3 | 0,3 | 60 | 0,005 | 2 | 0,05 | 0,01 | 10 |

# Sprendimas

# 2.1. Oilerio metodu



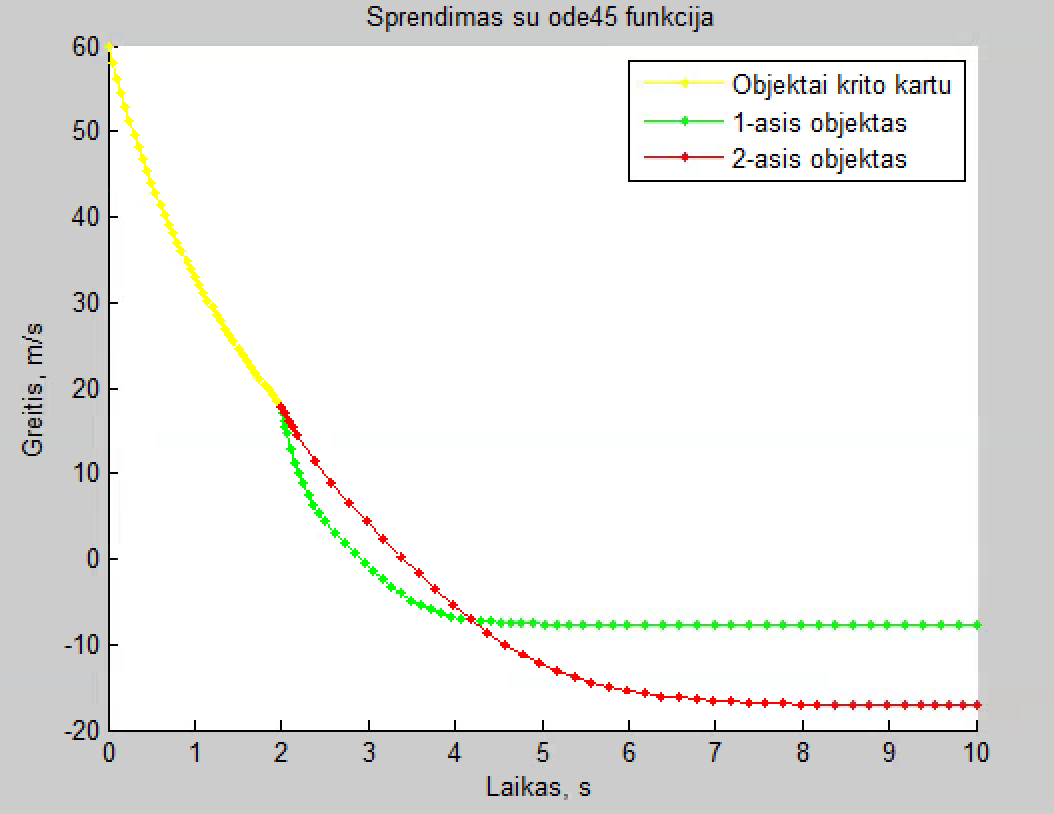
Paveikslas 1 Sprendimas Oilerio metodu, kai žingsnai yra 0.1, 0.15, ir 0.2



Paveikslas 1 Sprendimas Oilerio metodu, kai žingsnai yra 0.3, 0.7, ir 1

Grafikuose matomas dviejų krentančių kūnų, kurių masė ir forma skiriasi, kad skiriasi forma darau prielaidą pagal skirtingus oro pasipriešinimo koeficientus. Naudojant “Oilerio” metodą rezultatams įtakos daro žingsnių pasirinkimas. Jei pasirenkamas didesnis žingsnis metodas tampa nestabilus, tai parodo antras paveikslėlis.

# 2.2. Funkcija ode45



Paveikslas 3 Krentančų kūnų greičio kitimas laike. “Ode45” metodas.

# Išvados

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

Analizuojami du kūnai, paleisti į viršų tuo pat metu. Kūnai yra skirtingos formos ir masės.

Grafikuose matomas krentančių kūnų greičio kitimas laike. Kaip matoma naudojant “Oilerio” metodą, Naudojant “Oilerio” metodą rezultatams įtakos daro žingsnių pasirinkimas. Jei pasirenkamas didesnis žingsnis metodas tampa nestabilus Laboratorinis darbas atliktas naudojant “Oilerio” metodu bei standartine matlab funkcija “Ode45”.

# Programų tekstai

|  |
| --- |
| **Oiler.m** |
| %Oilerio metodas  function oileris    clc, clear all,  close all  spalva=['y'; 'm'; 'c'; 'r'; 'g'; 'b']; %spalvos    m1 = 0.3; % Pirmojo objekto masÎ (kg)  m2 = 0.3; % Antrojo objekto masÎ (kg)  v0 = 60; % Pradinis greitis (m/s)  ks = 0.005; % Sujungt¯ k˚n¯ oro pasiprieinimo koef. (kg/m)  ts = 2; % JudÎjimo kartu laikas (s)  k1 = 0.05; % Oro pasiprieinimo koef. 1-am objektui (kg/m)  k2 = 0.01; % Oro pasiprieinimo koef. 2-am objektui (kg/m)  tmax = 10; % Objekt¯ buvimo ore laikas  % kg \* m/s^2    x=0;% pradinÎs reikmÎs x - laikas, y - greitis    step\_variations = 3; % Kiek yra skirting¯ ˛ingsni¯  dx=[0.3 0.7 1]; % ˛ingsnis  %dx=[0.1 0.15 0.2]; % ˛ingsnis  %disp(dx(1))  figure(1), hold on, grid on, axis([0,10,-30,80]);  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;  for a=1:step\_variations  nsteps=tmax/dx(a); % ˛ingsni¯ 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 = 0.1, 1 objektas',...  '\Deltat = 0.1, 2 objektas',...  '\Deltat = 0.15, 1 objektas',...  '\Deltat = 0.15, 2 objektas',...  '\Deltat = 0.2, 1 objektas',...  '\Deltat = 0.2, 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 |

|  |
| --- |
| **Ode.m** |
| function ode  clc, clear all,  close all  global m m1 m2 v0 v1 v2 ks k1 k2;  m1 = 0.3; % Pirmojo objekto masÎ (kg)  m2 = 0.3; % Antrojo objekto masÎ (kg)  m = m1 + m2;  v0 = 60; % Pradinis greitis (m/s)  ks = 0.005; % Sujungt¯ k˚n¯ oro pasiprieinimo koef. (kg/m)  ts = 2; % JudÎjimo kartu laikas (s)  k1 = 0.05; % Oro pasiprieinimo koef. 1-am objektui (kg/m)  k2 = 0.01; % Oro pasiprieinimo koef. 2-am objektui (kg/m)  tmax = 10; % Objekt¯ 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 |