

Uruchomienie

Matlab	- /matlab/bin/matlab.exe
pwd	- nazwa bieżącej kartoteki
mkdir MatlabProjekty	- utworzenie własnej kartoteki projektów
cd /MatlabProjekty	- przejście do własnej kartoteki projektów
Simulink	-

Rodzina krzywych

Zadanie narysuj rodzinę funkcji $y = ax^2$, dla różnych wartości a

figure, hold on a = [1 5 6]; t = [0:2:20]; for i=1:3, plot(t, t.*t * a(i)); end	figure, hold on a = [1 5 6]; for i=1:3, plot((0:20), (0:20).*(0:20) * a(i)); end
------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------

Wykres 3-wymiarowy

Krzywa w przestrzeni – np. sprężyna

(1) t = [pi : pi/18 : 20*pi];	– [pi, pi+1*pi/18, pi+2*pi/18, ..., 20*pi]
(2) plot3(sin(t), cos(t), t);	– równa spirala
(3) plot3(sin(t) ./ t, cos(t) ./ t, t);	– zwężająca się spirala

Obraz powierzchni – np. $z = x \cdot e^{(-x^2 - y^2)}$

(1) [x, y] = meshgrid(-2 : .2 : 2, -2 : .2 : 3);	- dziedzina funkcji
(2) z = x .* exp(-x.^2 - y.^2);	- wektor rzędnej
(3) mesh(z);	- wykres siatkowy 3-D
(4) view(30, 10);	- obrót prawo-lewo, góra-dół

Rozwiązywanie równania różniczkowego $\ddot{x} + b\dot{x} + cx = u$

Model w pliku tekstowym „wzor3.mdl” + skrypt

$$\ddot{x} = u - b\dot{x} - cx$$

Podstaw: $y_1 = x; y_2 = \dot{x}$

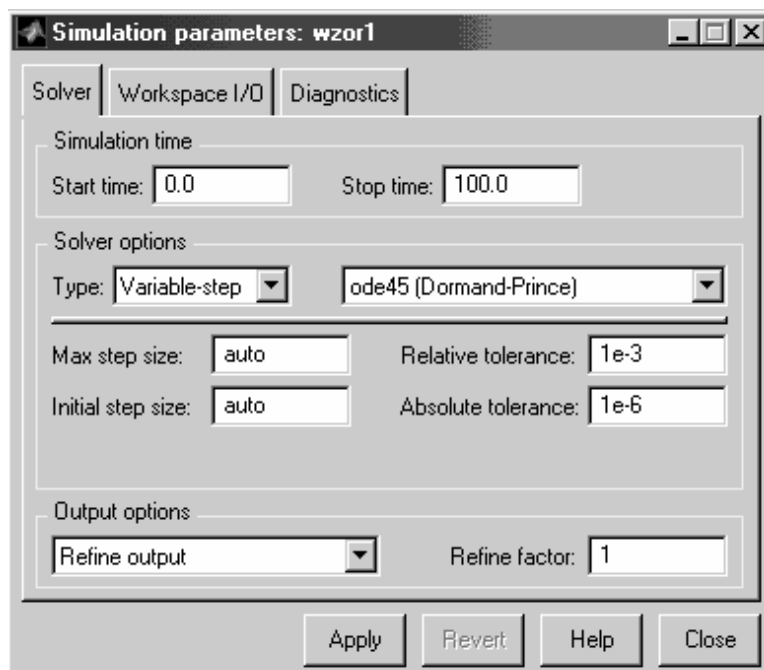
Układ równań:
$$\begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = -by_2 - cy_1 \end{cases}$$

M-plik funkcyjny „wzor3.m”

```
function xdot = wzor3(t, x)      % x wektor
global steps
steps = steps + 1;
y = zeros(2, 1);               %wektor 2-elem.
y(1) = x(2);
y(2) = - b * x(2) - c * x(1);
```

Skrypt uruchomieniowy:

```
steps = 0;
t0 = 0; t2 = 3;
x0 = [1 -1];
[t, x] = ode23('wzor3', t0, t2, x0, 1e-1, 0);
steps, plot(t,x)
```



Solver options

Krok:

Variable-step

Fixed step

Algorytm:

discrete (no continuous states)

ode45 (Dormand-Prince)

ode23 (Bogacki-Shampine)

ode113 (Adams)

ode15s (stiff/NDF)

ode23s (stiff/Mod. Rosenbrock)

ode23t (mod. stiff/Trapezoidal)

ode23tb (stiff/TR-BDF2)

Przykład nonstiff system (ruch ciała sztywnego bez wymuszenia)

$$\begin{cases} \dot{x}_1 = x_2 x_3 & x_1(0) = 0 \\ \dot{x}_2 = -x_1 x_3 & x_2(0) = 1 \\ \dot{x}_3 = -0.51 x_1 x_2 & x_3(0) = 1 \end{cases}$$

M-plik „rigid”

```
function dy = rigid(t,y)
dy = zeros(3,1); % a column vector
dy(1) = y(2) * y(3);
dy(2) = -y(1) * y(3);
dy(3) = -0.51 * y(1) * y(2);
```

Uruchomienie

%zmienna tolerancja, symulacja w czasie [0 12], warunki początkowe [0 1 1]

```
options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]);
[t,y] = ode45('rigid',[0 12],[0 1 1],options);
plot(T,Y(:,1),'-',T,Y(:,2),'-.',T,Y(:,3),'-')
```

Przykład stiff system (równanie van der Pola)

$$\begin{cases} \dot{x}_1 = x_2 & x_1(0) = 0 \\ \dot{x}_2 = 1000(1 - x_1^2)x_2 - x_1 & x_2(0) = 1 \end{cases}$$

M-plik „vdp1000”

```
function dy = vdp1000(t,y)
dy = zeros(2,1); % a column vector
dy(1) = y(2);
dy(2) = 1000*(1 - y(1)^2)*y(2) - y(1);
```

Uruchomienie

%domyślna tolerancja, symulacja w czasie [0 3000], warunki początkowe [2 0]

```
[T,Y] = ode15s('vdp1000',[0 3000],[2 0]);
plot(T,Y(:,1),'-o');
```