
Politechnika Świętokrzyska
Wydział Elektrotechniki, Automatyki i Informatyki
Katedra Elektrotechniki Przemysłowej i Automatyki
Zakład Urządzeń i Systemów Automatyki

Komputerowa symulacja układów dynamicznych

Rozwiązywanie równań różniczkowych

Instrukcja laboratoryjna
(wersja robocza)

Układ dynamiczny może być opisany układem równań różniczkowych w postaci:

$$\frac{dy(t)}{dt} = y' = f(y, t) \quad (1)$$

gdzie odpowiednio:

$f()$ - funkcja nieliniowa,

y - zmienna stanu,

t - czas.

Dla warunków początkowych $y_0 = y(t_0)$, poszukiwane rozwiązanie równania różniczkowego w chwili $t = t_1$, gdzie: $h = (t_1 - t_0)$, można wyznaczyć korzystając z rozwinięcia funkcji $f()$ w szereg Taylora w pobliżu punktu t_0 :

$$y_1 = y(t_0 + h) = y(t_0) + hy'(t_0) + \frac{h^2}{2!}y''(t_0) + \dots \quad (2)$$

Ograniczając rozwinięcie do:

$$y_1 = y(t_0) + hy'(t_0) \quad (3)$$

oraz podstawiając w miejsce pochodnej prawą stronę równania (1), uzyskuje się zależność:

$$y_1 = y(t_0) + hf(y(t_0), t_0) \quad (4)$$

Powtarzając powyższą procedurę dla kolejnych kroków otrzymuje się zależność określającą metodę Eulera (tabela 1). Zwiększenie dokładności rozwiązania równania (1), według przedstawionej procedury, wymaga dalszego rozwinięcia szeregu Taylora. Nie jest to wygodne, ponieważ dalsze składniki zawierają wyższe pochodne funkcji f . Wada ta została wyeliminowana przez Rungego i Kuttę poprzez wprowadzenie funkcji aproksymującej składniki wyższego rzędu w rozwinięciu Taylora (tabela 1).

Tabela 1. Algorytmy rozwiązywania równań różniczkowych

Algorytm Eulera	$x_{n+1} = x_n + hf(x_n, t_n)$
Algorytm Rungego - Kuty (R-K) 2 rzędu	$k_1 = hf(x_n, t_n)$ $k_2 = hf(x_n + \frac{k_1}{2 \cdot a}, t_n + \frac{h}{2 \cdot a})$ $x_{n+1} = x_n + (1-a) \cdot k_1 + a \cdot k_2$ a. Heuna: $a = \frac{1}{2}$ b. Eulera - Cauch'ego: $a = 1$
Algorytm Rungego - Kuty (R-K) 4 rzędu	$k_1 = hf(x_n, t_n)$ $k_2 = hf(x_n + \frac{k_1}{2}, t_n + \frac{h}{2})$ $k_3 = hf(x_n + \frac{k_2}{2}, t_n + \frac{h}{2})$ $k_4 = hf(x_n + k_3, t_n + h)$ $x_{n+1} = x_n + \frac{k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4}{6}$

Przykład

Wykorzystując metodę Eulera rozwiązać równanie różniczkowe:

$$y'' + y = \sin\left(\frac{t}{2}\right) + \cos\left(\frac{2t}{2}\right)$$

dla warunków początkowych:

$$y(0) = -1 \quad y'(0) = -2$$

Pomoc:

Rozwiązanie dokładne wynosi:

$$y = \frac{4\sin\left(\frac{t}{2}\right) + 4\cos\left(\frac{t}{2}\right) - 8\sin(t) - 7\cos(t)}{3}$$

Rozwiązanie

1. Podane równanie różniczkowe przekształcono do układu równań różniczkowych pierwszego rzędu:

$$y_1 = y, \quad y_2 = y'$$

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = \begin{bmatrix} y_2 \\ \sin(\frac{t}{2}) + \cos(\frac{t}{2})y \end{bmatrix}$$

Uzyskany układ równań zaimplementowano w postaci funkcji systemu Matlab – listing 1

Listing 1. Układ równań zaimplementowany w postaci funkcji

```
function [dy] = funkcja(t, y)
dy = [y(2); sin(0.5*t) + cos(0.5*t) - y(1)]
end
```

Do rozwiązania problemu wykorzystano metodę Eulera.

Listing 2. Rozwiązanie równania różniczkowego z wykorzystaniem metody Eulera

```
clear all
h = 0.01; % krok
time = 8*pi; % czas rozwiązania
y = [-1; -2]; % warunki początkowe
t = 0:h:time;
for i = 1:length(t)-1
    y(:,i+1) = y(:,i) + h * funkcja(t(i), y(:,i));
end
yd = ( 4.*sin(t./2) + 4.*cos(t./2) -8.*sin(t) -7.*cos(t) ) ./ 3;
figure(1);
plot(t,yd,'b', t,y(1,:), 'r', t,y(2,:), 'g--');
grid('on');
legend('ydok','ynum','dynum');
figure(2); plot(t, yd - y(1,:));
grid('on');
```

Uwaga:

Przy pomocy poleceń `tic` oraz `toc` sprawdzić ile czasu zajmuje uzyskanie rozwiązania.

Obliczyć błąd rozwiązania wykorzystując zależność:

$$error = \left(\sum (y_d - y_n)^2 \right)^{\frac{1}{2}}$$

gdzie:

y_d -rozwiązanie dokładne,

y_n - rozwiązanie numeryczne.

Powyższy wzór implementuje drugą normę wektorową (normę euklidesową), Polecenie w systemie Matlab: `norm(Y, 2)`. Gdzie Y jest wektorem, dla rozważanego problemu $Y=y_d-y_n$.

Zadania do wykonania

Zadanie 1

Rozwiązać równanie różniczkowe z podanego przykładu stosując algorytmy Heuna i Eulera-Couch'ego. Wygenerować wykresy jak w podanym przykładzie. Obliczyć czas potrzebny do uzyskania rozwiązania oraz błąd według podanego wzoru. Uzyskane wyniki porównać z metodą Eulera.

Uwaga: W celu porównania metod ustalić krok h oraz czas rozwiązania `time` (we wszystkich skryptach, implementujących analizowane metody wartości te powinny być takie same).

Zadanie 2

Rozwiązać równanie różniczkowe z podanego przykładu stosując algorytmy Rungego- Kuty 4 rzędu. Wygenerować wykresy jak w podanym przykładzie. Obliczyć czas potrzebny do uzyskania rozwiązania oraz błąd według podanego wzoru. Uzyskane wyniki porównać z metodami: Eulera, Heuna, Eulera Couch'ego.

Uwaga: W celu porównania metod ustalić krok h oraz czas rozwiązania `time` (we wszystkich skryptach, implementujących analizowane metody wartości te powinny być takie same).

Zadanie 3

Skorzystać z systemu pomocy programu Matlab i znaleźć informacje dotyczące wbudowanych metod: `ode23` oraz `ode45`. Wykorzystać uzyskane informacje, rozwiązując równanie różniczkowe z podanego przykładu przy pomocy wbudowanych metod.

Zadanie 4

Wykorzystując wbudowane w system Matlab metody rozwiązywania

równań różniczkowych (`ode23` oraz `ode45`), wyznaczyć odpowiedzi na wymuszenie skokowe i impulsowe obiektu inercyjnego I rzędu:

$$G(s) = \frac{k}{1 + sT}$$

Utworzyć funkcje implementujące równania różniczkowe opisujące badany model, oraz skrypt który:

- zainicjuje parametry modelu,
- uruchomi algorytm wybranej metody,
- wygeneruje wykres uzyskanych odpowiedzi.

Pomoc:

$$G(s) = \frac{k}{1 + sT} = \frac{Y}{U}$$

$$k \cdot U = Y(1 + sT)$$

$$k \cdot U = Y + sY \cdot T \quad / \mathcal{L}^{-1}$$

$$k \cdot u(t) = y(t) + T \frac{dy(t)}{dt}$$

$$\frac{dy(t)}{dt} = \frac{k \cdot u(t) - y(t)}{T}$$