Rozwiązywanie układów równań różniczkowych zwyczajnych

Autor: Wiktor Murawski

Przedmiot: Modelowanie matematyczne
Prowadzący: dr inż. Jakub Wagner

Politechnika Warszawska
Wydział Matematyki i Nauk Informacyjnych

Oświadczam, że niniejsza praca, stanowiąca podstawę do uznania osiągnięcia efektów uczenia się z przedmiotu Modelowanie matematyczne, została wykonana przeze mnie samodzielnie.

Warszawa
1 grudnia 2024

Spis treści

1 Lista Symboli i Akronimów

URRZ	układ równań różniczkowych zwyczajnych
t	zmienna skalarna, czas
h	wartość kroku całkowania
$N(h)$	zależna od kroku całkowania liczba punktów rozwiązania
h_{min}	najmniejszy badany krok całkowania
h_{max}	największy badany krok całkowania
$y_1(t), y_2(t), x(t)$	funkcje w dziedzinie czasu t
$\dot{y}_1(t), \dot{y}_2(t)$	dokładne rozwiązanie URRZ
$\hat{y}_1(t,h), \hat{y}_2(t,h)$	przybliżone rozwiązanie URRZ dla kroku całkowania h
$c_i, a_{i,j}, w_i$	współczynniki w tabeli Butchera; $i, j \in \{1, 2, 3\}$
$\mathbf{y}(t)$	pionowy wektor zawierający wartości $y_1(t)$ i $y_2(t)$, $\mathbf{y} = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$
t_n	wartość czasu, $t_n = t_0 + (n-1)h$, $n \in \mathbb{Z}^+$
$\mathbf{y}_n$	$\mathbf{y}(t_n)$
$\mathbf{f}(t_n, \mathbf{y}_n)$	funkcja określona przez URRZ: $\left.\frac{d\mathbf{y}(t)}{dt}\right\rvert_{t=t_n} = \mathbf{f}(t_n, \mathbf{y}_n)$
$\delta_1(h), \delta_2(h)$	zagregowane błędy względne dla kroku całkowania h

2 Wprowadzenie

Dany jest następujący układ równań różniczkowych zwyczajnych (URRZ):

$$\left.\begin{aligned} \frac{dy_1(t)}{dt} &= -\frac{14}{3}y_1(t) - \frac{2}{3}y_2(t) + x(t) \\ \frac{dy_2(t)}{dt} &= \frac{2}{3}y_1(t) - \frac{19}{3}y_2(t) + x(t) \end{aligned}\right\} \text{ dla } t \in [0, 8], \text{ w którym } x(t) = \exp(-t)\sin(t) \tag{1}$$

W celu rozwiązania URRZ przedstawionego w (1) dla zerowych warunków początkowych, tj. $y_1(0) = y_2(0) = 0$:

1. wyznaczono dokładne rozwiązanie URRZ za pomocą procedury `dsolve` (*MATLAB Symbolic Toolbox*)

2. zastosowano procedurę `ode45` (*MATLAB*), będącą zaawansowaną implementacją metody Rungego-Kutty czwartego rzędu z adaptacyjnym krokiem czasowym

3. zaimplementowano oraz zastosowano trzy inne metody dyskretne:

   - metodę zdefiniowaną wzorem $\mathbf{y}_n = \mathbf{y}_{n-1} + h\mathbf{f}\left(t_{n-1} + \frac{h}{2}, \mathbf{y}_{n-1} + \frac{h}{2}\mathbf{f}\big(t_{n-1}, \mathbf{y}_{n-1}\big)\right)$
   - metodę zdefiniowaną wzorem $\mathbf{y}_n = \mathbf{y}_{n-2} + h\Big[\mathbf{f}\left(t_n, \mathbf{y}_n\right) + \mathbf{f}\left(t_{n-2}, \mathbf{y}_{n-2}\right)\Big]$
   - metodę zdefiniowaną wzorem $\mathbf{y}_n = \mathbf{y}_{n-1} + h\sum_{k=1}^{3} w_k \mathbf{f}_k$

     gdzie $\mathbf{f}_k = \mathbf{f}\left(t_{n-1} + c_k h, \mathbf{y}_{n-1} + h\sum_{\kappa=1}^{3} a_{k,\kappa}\mathbf{f}_\kappa\right)$

     a współczynniki przyjmują wartości przedstawione w poniższej tabeli Butchera:

$$\begin{array}{c|ccc} c_1 & a_{1,1} & a_{1,2} & a_{1,3} \\ c_2 & a_{2,1} & a_{2,2} & a_{2,3} \\ c_3 & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline & w_1 & w_2 & w_3 \end{array} \quad = \quad \begin{array}{c|ccc} 0 & \frac{1}{6} & -\frac{1}{6} & 0 \\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 \\ 1 & \frac{1}{6} & \frac{5}{6} & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

3 Metodyka i Wyniki Doświadczeń

3.1 dsolve

Za pomocą poniższego kodu korzystającego z procedury `dsolve` wyznaczono rozwiązanie (1):

plik `solve_using_dsolve.m`

```
function [y1_exact, y2_exact] = solve_using_dsolve(x)

syms t y1(t) y2(t)

eq1 = diff(y1,t,1) == -14/3*y1(t) - 2/3*y2(t) + x(t);
eq2 = diff(y2,t,1) == 2/3*y1(t) - 19/3*y2(t) + x(t);
eqns = [eq1; eq2];

cond1 = y1(0) == 0;
cond2 = y2(0) == 0;
conds = [cond1; cond2];

sol = dsolve(eqns,conds);
y1_exact = sol.y1;
y2_exact = sol.y2;

end % function
```

Uzyskano:

$$\dot{y}_1(t) = -\frac{1}{2}\exp(-6t)\left(\frac{\exp(5t)\left(\cos(t) - 5\sin(t)\right)}{39} - \frac{1}{39}\right) - 2\exp(-5t)\left(\frac{\exp(4t)\left(\cos(t) - 4\sin(t)\right)}{51} - \frac{1}{51}\right)$$

$$\dot{y}_2(t) = -\exp(-6t)\left(\frac{\exp(5t)\left(\cos(t) - 5\sin(t)\right)}{39} - \frac{1}{39}\right) - \exp(-5t)\left(\frac{\exp(4t)\left(\cos(t) - 4\sin(t)\right)}{51} - \frac{1}{51}\right)$$

4 Dyskusja Wyników Eksperymentów Numerycznych

Interpretacja i analiza uzyskanych wyników. Dyskusja na temat poprawności i znaczenia uzyskanych rezultatów.

Bibliografia

1. Autor, Tytuł, Wydawnictwo, Rok wydania.
2. Strona internetowa: `http://przyklad.com`.
3. Dokumentacja MATLAB: `https://www.mathworks.com/help`.

Listing Programów

plik `Projekt1.m`

```
% Przedział t
tmin = 0;
tmax = 8;

% Współczynniki URRZ
A = [-14/3,-2/3; 2/3,-19/3];
b = [1;1];

% Funkcja x
x = @(t) exp(-t)*sin(t);


%% Zadanie 1

[y1_exact, y2_exact] = solve_using_dsolve(x);
fprintf("y_1(t) = \n");
pretty(y1_exact);
fprintf("y_2(t) = \n");
pretty(y2_exact);


%% Zadanie 2

% Wartość kroku
h = 0.2;

% Wyznaczenie N(h) oraz wektora t
N = floor((tmax - tmin)/h) + 1;
t = tmin:h:tmax;

% Procedura ode45

% Funkcja f
f = @(t,y) A*y + b*x(t);
% Przedział czasowy
tspan = [0, 8];
% Warunki początkowe
y0 = [0; 0];

% Wywołanie procedury
[T_ode45,y_ode45] = ode45(f,tspan,y0);

y_ode45 = y_ode45';
y1_ode45 = y_ode45(1,:);
y2_ode45 = y_ode45(2,:);

% Metoda 1
y_metoda1 = metoda1(A,b,x,h,N,t);

% Metoda 2
y_metoda2 = metoda2(A,b,x,h,N,t);

% Metoda 3
y_metoda3 = metoda3(A,b,x,h,N,t);
```

```matlab

cztery_wykresy(tspan,y1_exact,y2_exact,...
  T_ode45,y1_ode45,y2_ode45,t,y_metoda1,y_metoda2,y_metoda3);


%% Zadanie 3

y1_dot = matlabFunction(y1_exact);
y2_dot = matlabFunction(y2_exact);

n = 1e2;
hmin = 0.01;
hmax = 1.00;
h_range = linspace(hmin,hmax,n);

delta11 = zeros(n,1);
delta12 = zeros(n,1);
delta21 = zeros(n,1);
delta22 = zeros(n,1);
delta31 = zeros(n,1);
delta32 = zeros(n,1);

for i = 1:n
  h = h_range(i);
  N = floor((tmax - tmin)/h) + 1;
  t = tmin:h:tmax;

  y1 = metoda1(A,b,x,h,N,t);
  y2 = metoda2(A,b,x,h,N,t);
  y3 = metoda3(A,b,x,h,N,t);

  [delta11(i), delta12(i)] = wyznaczBledy(y1,y1_dot,y2_dot,t,N);
  [delta21(i), delta22(i)] = wyznaczBledy(y2,y1_dot,y2_dot,t,N);
  [delta31(i), delta32(i)] = wyznaczBledy(y3,y1_dot,y2_dot,t,N);
end % for h

% Wykresy błędów zagregowanych w zależności od h
figure(2);clf;
subplot(2,1,1);
hold on;
yscale('log');
plot(h_range,delta11,'DisplayName','$\delta_{1,1}(t)$','LineWidth',1);
plot(h_range,delta21,'DisplayName','$\delta_{1,2}(t)$','LineWidth',1);
plot(h_range,delta31,'DisplayName','$\delta_{1,3}(t)$','LineWidth',1);
lgd = legend('show','Interpreter', 'latex', 'Location', 'northwest');
set(lgd, 'FontSize', 13);

subplot(2,1,2);
hold on;
yscale('log');
plot(h_range,delta12,'DisplayName','$\delta_{1,1}(t)$','LineWidth',1);
plot(h_range,delta22,'DisplayName','$\delta_{1,2}(t)$','LineWidth',1);
plot(h_range,delta32,'DisplayName','$\delta_{1,3}(t)$','LineWidth',1);
lgd = legend('show','Interpreter', 'latex', 'Location', 'northwest');
set(lgd, 'FontSize', 13);
```

plik `metoda1.m`

```
function y = metoda1(A,b,x,h,N,t)

y = zeros(2,N);

for n = 2:N
  y(:,n) = y(:,n-1) + ...
    h*(A*(y(:,n-1) + h/2*(A*y(:,n-1) + b*x(t(n-1)))) + b*x(t(n-1) + h/2));
end % for n

end % function
```

plik `metoda2.m`

```
function y = metoda2(A,b,x,h,N,t)

I = eye(2);
y = zeros(2,N);

% Niejawna metoda Eulera
y(:,2) = (I - h*A) \ (y(:,1) + h*b*x(t(2)));

for n = 3:N
  y(:,n) = (I - h*A) \ ...
       (y(:,n-2) + h*A*y(:,n-2) +h*b*x(t(n)) + h*b*x(t(n-2)));
end % for n

end % function
```

plik `metoda3.m`

```
function y = metoda3(A,b,x,h,N,t)

% Współczynniki metody 3
c = [0,   1/2,   1];
w = [1/6, 2/3, 1/6];
a = [
  1/6, -1/6, 0;
  1/6,  1/3, 0;
  1/6,  5/6, 0;
  ];

I = eye(2);
y = zeros(2,N);

L = [
  I-h*a(1,1)*A,  -h*a(1,2)*A,  -h*a(1,3)*A;
   -h*a(2,1)*A, I-h*a(2,2)*A,  -h*a(2,3)*A;
   -h*a(3,1)*A,  -h*a(3,2)*A, I-h*a(3,3)*A;
  ];

for n = 2:N
  p = [
    A*y(:,n-1) + b*x(t(n-1) + c(1)*h);
```

```
    A*y(:,n-1) + b*x(t(n-1) + c(2)*h);
    A*y(:,n-1) + b*x(t(n-1) + c(3)*h);
  ];

  g = L \ p;

  f1 = g(1:2);
  f2 = g(3:4);
  f3 = g(5:6);

  y(:,n) = y(:,n-1) + h*(w(1)*f1 + w(2)*f2 + w(3)*f3);

end % for n


end % function
```