Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a umelej inteligencie

Simulačné systémy

(Zadanie č. 3)

2. ročník, Inteligentné systémy
Zimný semester 2009

Michal Puheim

Text zadania

Riešenie Nelineárnej Diferenciálnej Rovnice (NDR) numericky so zvolenou numerickou technikou a algoritmicky v programovacom jazyku Matlab.

Obsah zadania:
1. Zadať NDR s konštantnými koeficientami.
2. Numericky vyriešiť v tabuľke zvolenou metódou.
3. Algoritmicky vyriešiť NDR v programovacom jazyku Matlab.

Dodefinovanie zadania

Pre vypracovanie zadania som si vybral diferenciálnu rovnicu

$$y'(t) = -ty + \frac{4t}{y},$$

s počiatočnou podmienkou $y(0)=1$, riešenú na intervale $t \in \langle 0;4 \rangle$ s krokom 0,1.

Analýza úlohy

Pre numerické riešenie použijeme metódu Runge-Kutta 4. rádu, ktorú možno zapísať:

$$y_{n+1} = y_n + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

kde konštanty K_1, K_2, K_3 a K_4 sú určené:

$$f(t;y) = -ty + \frac{4t}{y}$$
$$K_1 = h \cdot f(t_n; y_n)$$
$$K_2 = h \cdot f(t_n + \frac{h}{2}; y_n + \frac{K_1}{2})$$
$$K_3 = h \cdot f(t_n + \frac{h}{2}; y_n + \frac{K_2}{2})$$
$$K_4 = h \cdot f(t_n + h; y_n + K_3)$$

Postupne s krokom 0,1 touto numerickou metódou vypočítame hodnoty y pre všetky $t \in \langle 0;4 \rangle$.

krok	t	y
1.	0,0	1,00000
2.	0,1	1,01482
3.	0,2	1,05718
4.	0,3	1,12170
5.	0,4	1,20149
6.	0,5	1,28981
7.	0,6	1,38093
8.	0,7	1,47042
9.	0,8	1,55503
10.	0,9	1,63261

krok	t	y
11.	1,0	1,70187
12.	1,1	1,76221
13.	1,2	1,81362
14.	1,3	1,85645
15.	1,4	1,89140
16.	1,5	1,91932
17.	1,6	1,94115
18.	1,7	1,95787
19.	1,8	1,97040
20.	1,9	1,97960

krok	t	y
21.	2,0	1,98621
22.	2,1	1,99086
23.	2,2	1,99406
24.	2,3	1,99621
25.	2,4	1,99763
26.	2,5	1,99855
27.	2,6	1,99913
28.	2,7	1,99949
29.	2,8	1,99970
30.	2,9	1,99983

krok	t	y
31.	3,0	1,99991
32.	3,1	1,99995
33.	3,2	1,99997
34.	3,3	1,99999
35.	3,4	1,99999
36.	3,5	2,00000
37.	3,6	2,00000
38.	3,7	2,00000
39.	3,8	2,00000
40.	3,9	2,00000

Návrh riešenia

Prepis rovnice do substitučného kanonického tvaru pre numerické riešenie metódou ode45:

$$y(t) = x_1$$
$$y'(t) = x'_1 = -t_1 + \frac{4t}{x_1}$$

Numerické riešenie pomocou metódy Runge-Kutta použijeme pre porovnanie.

Implementácia riešenia

Program naprogramovaný v prostredí simulačného jazyka MATLAB je zostavený z troch súborov – hlavného programu ndr.m, funkcie dy.m a funkcie rungekutta.m.

Použité premenné:
T – časový interval,
PP – interval obsahujúci počiatočné podmienky PP(1) a PP(2),
t1,t2 – vektory času pre obe numerické metódy,
y – vektor funkcie *y* určenej numericky pomocou funkcie ode45,
d – vektor funkcie *y* určenej numericky metódou Runge-Kutta.

Popis programu:
V hlavnej časti programu je pevne daný časový interval T a počiatočné podmienky. Program odovzdá údaje funkcii ode45, ktorá zároveň používa funkciu dy.m, kde je rovnica zapísaná pomocou substitučného kanonického tvaru. Funkcia ode45 numerickým výpočtom určí vektor t1 (z intervalu T) a k nemu ekvivalentné hodnoty funkcie y. Program následne pomocou vlastnej funkcie rungekutta.m určí vektor t2 a vypočíta vektor d (riešenia funkcie y), ktorý je vypočítaný cez algoritmus metódy Runge-Kutta 4. rádu. Program nakoniec vykreslí priebehy funkcií pre obe metódy riešenia.

Zdrojový kód ndr.m

```
%  Program pre riesenie nelinearnej diferencialnej rovnice
%  (dy/dt)=-ty+4t/y
%  na casovom intervale t=[0..4]
%  s pociatocnou podmienkou y(0)=1

%  Urcenie casoveho intervalu
T=0:0.1:4;
%  Urcenie pociatocnych podmienok
PP=1;
%  Kontrolny vypis
fprintf('Pocitam diferencialnu rovnicu (dy/dt)=-ty+4t/y,\n')
fprintf('na casovom intervale t=[%d..%d], pre pociatocne podmienky t=0,
y(t)=1.\n',T(1),T(length(T)))

%  Riesenie pomocou funkcie ode45
[t1,y]=ode45('dy',T,PP);
%  Riesene metodou Runge-Kutta
[t2,d]=rungekutta(T);

%  Vykreslenie vyriesenych priebehov:
subplot(1,3,1)
plot(t1,y(:,1))
title('ode45 y(t)'), xlabel('t'),ylabel('yn(t)')
subplot(1,3,2)
plot(t2,d,'g--')
title('Runge-Kutta riesenie riesenie y(t)'), xlabel('t'),ylabel('ya(t)')
subplot(1,3,3)
plot(t1,y(:,1),t2,d,'g--')
title('Obe riesenia y(t)'), xlabel('t'),ylabel('yn(t),ya(t)')
return
```

Zdrojový kód dy.m

```
function xder=dy(t,x)
%  Zapis danej DR pomocu substitucneho kanonickeho tvaru
xder=-t.*x(1)+4*t./x(1);
return
```

Zdrojový kód rungekutta.m

```
function [vt,vy]=rungekutta(T)
%  Vypocet funkcie metodou Runge-Kutta 4. radu
vt=zeros(length(T)-1,1);
vy=zeros(length(T)-1,1);
poc=1; % pocitadlo
h=h=T(2)-T(1); % krok
t=0;
y=1;
while t<=max(T)
    vt(poc)=t;
    vy(poc)=y;
    K1=h*(-t*y+4*t/y);
    K2=h*(-(t+h/2)*(y+K1/2)+4*(t+h/2)/(y+K1/2));
    K3=h*(-(t+h/2)*(y+K2/2)+4*(t+h/2)/(y+K2/2));
    K4=h*(-(t+h)*(y+K3)+4*(t+h)/(y+K3));
    y=y+(K1+2*K2+2*K3+K4)/6;
    t=t+h;
    poc=poc+1;
end
return
```

Príloha: Model v Simulinku