FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

Proiect nr. 4

B. REZOLVAREA ECUAȚIILOR DIFERENȚIALE CU METODE DE TIP RUNGE-KUTTA (METODA RUNGE-KUTTA DE ORDINUL 2, METODA RUNGE-KUTTA DE ORDINUL 3 ȘI METODA RUNGE-KUTTA DE ORDINUL 4)

Proiect la disciplina *Matematici asistate de calculator*

Student

Armina-Mihaela Cioabă,

anul I IS, subgrupa 2.1

An universitar 2018-2019

Semestrul II

Cuprins

TEMA B: Rezolvarea ecuațiilor diferențiale cu metode de tip Runge-Kutta (metode Runge-Kutta de ordin 2, metode Runge-Kutta de ordinul 3 și metode Runge-Kutta de ordinul 4)

1. Considerații teoretice

Metodele de tip Runge-Kutta sunt metode monopas cu algoritm explicit.

Pentru reducerea erorii, la determinarea lui y_i, i = 1, 2, …, n se calculează valorile lui f(x,y) într-un număr de puncte intermediare ale intervalului [x_{i-1}, x_i], acest număr de puncte fiind legat direct de ordinul p al metodei.

Forma generală a metodelor de tip Runge-Kutta este următoarea:

$$k_1 = h \cdot f(x_{i-1}, y_{i-1}), \quad (1.1)$$

$$k_j = h \cdot f(x_{i-1} + h \cdot b_j,\ y_{i-1} + \sum_{m=1}^{j-1} c_{jm} \cdot k_m), \text{ j=2...p}, \quad (1.2)$$

$$y_i = y_{i-1} + \sum_{m=1}^{p} a_m \cdot k_m \quad (1.3)$$

Determinarea coeficienților a_m, m=1,..., p, b_j,j=2,...,p și c_{jm}, m=1,..., j-1, j=2,..., p - prin dezvoltarea în serie Taylor a ambilor termeni ai relației anterioare și identificarea coeficientilor expresiilor obținute. [1]

Metoda Runge-Kutta de ordin 2 (p=2):

$$k_1 = h \cdot f(x_{i-1}, y_{i-1}) \quad (1.4)$$

$$k_2 = h \cdot f\left(x_{i-1} + \frac{h}{2}, y_{i-1} + \frac{k_1}{2}\right) \quad (1.5)$$

$$y_i = y_{i-1} + \frac{k_1}{2} + \frac{k_2}{2} \quad (1.6)$$

Metoda Runge-Kutta de ordin 3 (p=3):

$$k_1 = h \cdot f(x_{i-1}, y_{i-1}) \quad (1.7)$$

$$k_2 = h \cdot f\left(x_{i-1} + \frac{h}{2}, y_{i-1} + \frac{k_1}{2}\right) \quad (1.8)$$

$$k_3 = h \cdot f(x_{i-1} + h, y_{i-1} - k_1 + 2k_2) \quad (1.9)$$

$$y_i = y_{i-1} + \frac{1}{6}(k_1 + 4k_2 + k_3) \tag{1.10}$$

Metoda Runge-Kutta de ordin 4 (p=4):

$$k_1 = h \cdot f(x_{i-1}, y_{i-1}) \tag{1.11}$$

$$k_2 = h \cdot f\left(x_{i-1} + \frac{h}{2}, y_{i-1} + \frac{k_1}{2}\right) \tag{1.12}$$

$$k_3 = h \cdot f\left(x_{i-1} + \frac{h}{2}, y_{i-1} + \frac{k_2}{2}\right) \tag{1.13}$$

$$k_4 = h \cdot f(x_{i-1} + h, y_{i-1} + k_3) \tag{1.14}$$

$$y_i = y_{i-1} + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{1.15}$$

Pentru a rezolva ecuația diferențială trebuie respectați următorii pași (pseudocod):

Se află pasul h=(b-a)/n;

Se ia x din intervalul dat;

Se pune condiția inițială ca y(0)=y_0;

Se calculează valorile lui f(x,y) în punctele intermediare ale intervalului dat;

Se aplică formulele în funcție de ordinal p al metodei folosite. [2]

2. Descrierea metodelor numerice

Metodele de tip Euler sunt explicite şi nu necesită valori de start, însă au ca dezavantaje atât volumul consistent de calcule, cât și erorile făcute la fiecare pas, ele putând să se propage imprevizibil. Faptul că au un ordin scăzut al erorii de consistenţă conduce la o aplicabilitate limitată. În scopul obţinerii unor metode de ordin ridicat trebuie renunţat fie la proprietatea de a fi unipas şi pastrată liniaritatea, fie în mod invers. Metodele de tip Runge-Kutta sunt metode monopas cu algoritm explicit. Acestea sunt neliniare şi conservă caracteristicile metodelor unipas, având un ordin ridicat.

Metodele de tip Runge-Kutta au ca avantaje faptul că sunt metode directe, adică pentru determinarea aproximării soluţiei la pasul i+1 avem nevoie de informaţiile de la punctul anterior x_i, y_i. De asemenea ele sunt identice cu seriile Taylor până la termenii h^n , unde h este pasul curent iar n este diferit pentru metode diferite din această familie şi defineşte ordinul metodei. În plus, în procesul de calcul nu necesită decât evaluarea funcţiei din memebrul drept pentru diverse valori x şi y, calculul derivatelor acesteia fiind exclus. [1]

Exemplul 1:

Să se rezolve problema Cauchy următoare:

$$\begin{cases} y'(x) = x + \dfrac{y}{x} \\ y(1) = 2 \end{cases}$$

Pe intervalul [1,2] folosind metoda Runge Kutta de ordin 2. [3]

Folosind metoda Runge-Kutta de ordinul 2, se particularizează relațiile (1.4), (1.5), (1.6) și obținem:

x_0 = 1.0000 y_0 = 2.0000

x_1 = 1.1000 y_1 = 2.3095

x_2= 1.2000 y_2= 2.6391

x_3 = 1.3000 y_3 = 2.9886

x_4 = 1.4000 y_4= 3.3582

x_5 = 1.5000 y_5 = 3.7477

x_6 = 1.6000 y_6 = 4.1572

x_7= 1.7000 y_7= 4.5868

x_8 = 1.8000 y_8= 5.0363

x_9 = 1.9000 y_9= 5.5058

x_{10} = 2.0000 y_{10} = 5.9954

Exemplul 2:

Să se rezolve problema Cauchy următoare:

$$\begin{cases} y'(x) = x^2 + y^2 + 1 \\ y(0) = 0 \end{cases}$$

Pe intervalul [0,1] folosind metoda Runge Kutta de ordin 3. [3]

Folosind metoda Runge-Kutta de ordinul 3, se particularizează relațiile (1.7), (1.8), (1.9), (1.10) și obținem:

$x_0 = 0$ $y_0 = 0$

$x_1 = 0.1000$ $y_1 = 0.1007$

$x_2 = 0.2000$ $y_2 = 0.2054$

$x_3 = 0.3000$ $y_3 = 0.3187$

$x_4 = 0.4000$ $y_4 = 0.4457$

$x_5 = 0.5000$ $y_5 = 0.5930$

$x_6 = 0.6000$ $y_6 = 0.7696$

$x_7 = 0.7000$ $y_7 = 0.9889$

$x_8 = 0.8000$ $y_8 = 1.2723$

$x_9 = 0.9000$ $y_9 = 1.6573$

$x_{10} = 1.0000$ $y_{10} = 2.2179$

Exemplul 3:

Să se rezolve problema Cauchy următoare:

$$\begin{cases} y'(x) = \dfrac{x - y}{x} \\ y(1) = 1 \end{cases}$$

Pe intervalul [1,2] folosind metoda Runge Kutta de ordin 4. [3]

Folosind metoda Runge-Kutta de ordinal 4, se particularizează relațiile (1.11), (1.12), (1.13), (1.14), (1.15) şi obținem:

x_0 = 1.0000 y_0 = 1.0000

x_1 = 1.1000 y_1 = 1.0039

x_2= 1.2000 y_2= 1.0155

x_3 = 1.3000 y_3 = 1.0331

x_4 = 1.4000 y_4= 1.0554

x_5 = 1.5000 y_5 = 1.0815

x_6 = 1.6000 y_6 = 1.1106

x_7= 1.7000 y_7= 1.1421

x_8 = 1.8000 y_8= 1.1757

x_9 = 1.9000 y_9= 1.2111

x_{10} = 2.0000 y_{10} = 1.2479

3. Prezentarea implementării metodelor numerice utilizate

Metoda Runge-Kutta de ordinul 2

```
function [X,Y]=RungeKutta2(f,a,b,y0,n) %Apelul si declararea functiei
%a si b reprezinta capetele intervalului, y0 reprezinta conditia initiala,
%iar n reprezinta numarul de iteratii
tic;
h=(b-a)/n; %Aflarea pasului
t=a:h:b; %Initializarea domeniului de valori
Y=zeros(n,1); %Alocarea de spatiu lui Y de n linii si o coloana
X=[a:h:b]' %Transpunerea lui X (in X se afla abscisele)
Y(1,1)=y0; %Initializarea lui y0

% Aplicarea algoritmului Runge-Kutta de ordin 2
for i=2:(length(t))
```

```
    k1=h*f(X(i-1,1),Y(i-1,1));
   k2=h*f(X(i-1,1)+h,Y(i-1,1)+k1);
 % Aflarea valorilor functiei in x
  Y(i,1)=Y(i-1,1)+k1/2+k2/2;
end
Y
% Aflarea timpului de executie al algoritmului Runge-Kutta de ordin 2
timprungekutta2=toc
% Aflarea timpul de executie al metodei ode23
tic;
[xval,yval] = ode23(f,t,y0)
timpode23=toc
% Diferenta
diferenta=Y-yval;
% Eroarea
eroare=norm(diferenta)
%Reprezentarea grafica a rezultatelor obtinute folosind functia ode23
%cu negru
%Reprezentarea grafica a rezultateloer obtinute folosind algoritmul
%Runge-Kutta de ordin 2 cu rosu
plot(xval,yval,'b',X,Y,'r--')
pause;
```

Metoda Runge-Kutta de ordinul 3

```
function [X,Y]=RungeKutta3(f,a,b,y0,n) %Apelul si declararea functiei
%a si b reprezinta capetele intervalului, y0 reprezinta conditia initiala,
%iar n reprezinta numarul de iteratii
tic;
h=(b-a)/n; %Aflarea pasului
t=a:h:b; %Initializarea domeniului de valori
Y=zeros(n,1); %Alocarea de spatiu lui Y de n linii si o coloana
X=[a:h:b]' %Transpunerea lui X (in X se afla abscisele)
Y(1,1)=y0; %Initializarea lui y0

% Aplicarea algoritmului Runge-Kutta de ordin 2
for i=2:(length(t))
   k1=h*f(X(i-1,1),Y(i-1,1));
   k2=h*f(X(i-1,1)+h/2, Y(i-1,1)+k1/2);
   k3=h*f(X(i-1,1)+h,Y(i-1,1)-k1+2*k2);
```

```
 % Aflarea valorilor functiei in x
  Y(i,1)=Y(i-1,1)+(k1+4*k2+k3)/6;
end
Y
% Aflarea timpului de executie al algoritmului Runge-Kutta de ordin 2
timprungekutta3=toc
% Aflarea timpul de executie al metodei ode23
tic;
[xval,yval] = ode23(f,t,y0)
timpode23=toc
% Diferenta
diferenta=Y-yval;
% Eroarea
eroare=norm(diferenta)
%Reprezentarea grafica a rezultatelor obtinute folosind functia ode23
%cu negru
%Reprezentarea grafica a rezultateloer obtinute folosind algoritmul
%Runge-Kutta de ordin 3 cu rosu
plot(xval,yval,'b',X,Y,'r--')
pause;
```

Metoda Runge-Kutta de ordinul 4

```
function [X,Y]=RungeKutta4(f,a,b,y0,n) %Apelul si declararea functiei
%a si b reprezinta capetele intervalului, y0 reprezinta conditia initiala,
%iar n reprezinta numarul de iteratii
tic;
h=(b-a)/n; %Aflarea pasului
t=a:h:b; %Initializarea domeniului de valori
Y=zeros(n,1); %Alocarea de spatiu lui Y de n linii si o coloana
X=[a:h:b]' %Transpunerea lui X (in X se afla abscisele)
Y(1,1)=y0; %Initializarea lui y0

% Aplicarea algoritmului Runge-Kutta de ordin 2
for i=2:(length(t))
   k1=h*f(X(i-1,1),Y(i-1,1));
   k2=h*f(X(i-1,1)+h/2,Y(i-1,1)+k1/2);
   k3=h*f(X(i-1,1)+h/2,Y(i-1,1)+k2/2);
   k4=h*f(X(i-1,1)+h/2,Y(i-1,1)+k3/2);
 % Aflarea valorilor functiei in x
```

```
    Y(i,1)=Y(i-1,1)+(k1+2*(k2+k3)+k4)/6;
end
Y
% Aflarea timpului de executie al algoritmului Runge-Kutta de ordin 2
timprungekutta4=toc
% Aflarea timpul de executie al metodei ode23
tic;
[xval,yval] = ode45(f,t,y0)
timpode45=toc
% Diferenta
diferenta=Y-yval;
% Eroarea
eroare=norm(diferenta)
%Reprezentarea grafica a rezultatelor obtinute folosind functia ode23
%cu negru
%Reprezentarea grafica a rezultateloer obtinute folosind algoritmul
%Runge-Kutta de ordin 4 cu rosu
plot(xval,yval,'b',X,Y,'r--')
pause;
```
[4]

4. Prezentarea funcțiilor utilizate în Matlab

Funcția ode23

Rezolvă ecuații diferențiale, metodă de ordin scăzut.

Funcția ode45

Rezolvă ecuații diferențiale, metodă de ordin mediu.

Funcția tic

Această funcție a mediului Matlab pornește un cronometru pentru a măsura performanța. Funcția tic lucrează împreună cu funcția toc.

Funcția toc

Această funcție a mediului Matlab citește timpul măsurat de cronometrul pornit de funcția tic.

Funcția zeros

Creează matricea tuturor zerourilor.

Funcția length

Se folosește pentru aflarea lungimii unui vector și pentru aflarea dimensiunii maxime a unei matrice.

Funcția norm

Calculează norma euclidiană.

Funcția plot

Face reprezentarea grafică a rezultatelor.

5. Studiul performanței

Problema 1:

Să se rezolve problema Cauchy următoare:

$$\begin{cases} y'(x) = x + \dfrac{y}{x} \\ y(1) = 2 \end{cases}$$

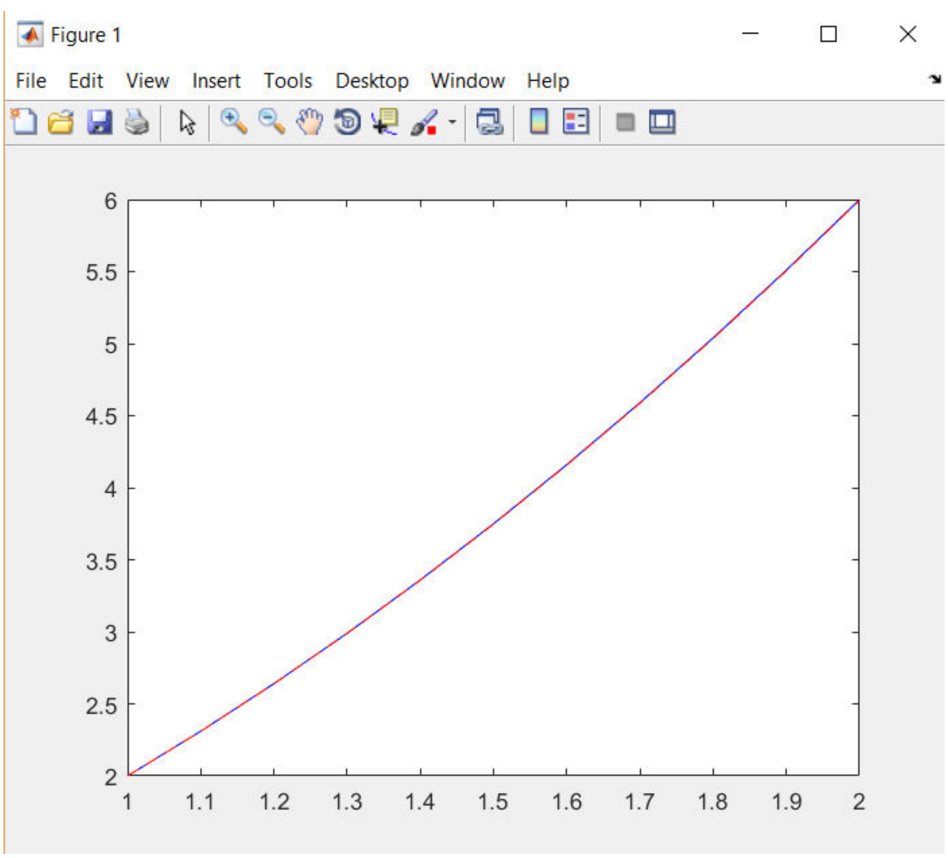
Pe intervalul [1,2] folosind metoda Runge Kutta de ordin 2.

timprungekutta2 =0.1537

timpode23 =0.5942

eroare =0.0089

X =	Y =
1.0000	2.0000
1.1000	2.3095
1.2000	2.6391
1.3000	2.9886
1.4000	3.3582
1.5000	3.7477
1.6000	4.1572
1.7000	4.5868
1.8000	5.0363
1.9000	5.5058
2.0000	5.9954



Problema 2:

Să se rezolve problema Cauchy următoare:

$$\begin{cases} y'(x) = x^2 + y^2 + 1 \\ y(0) = 0 \end{cases}$$

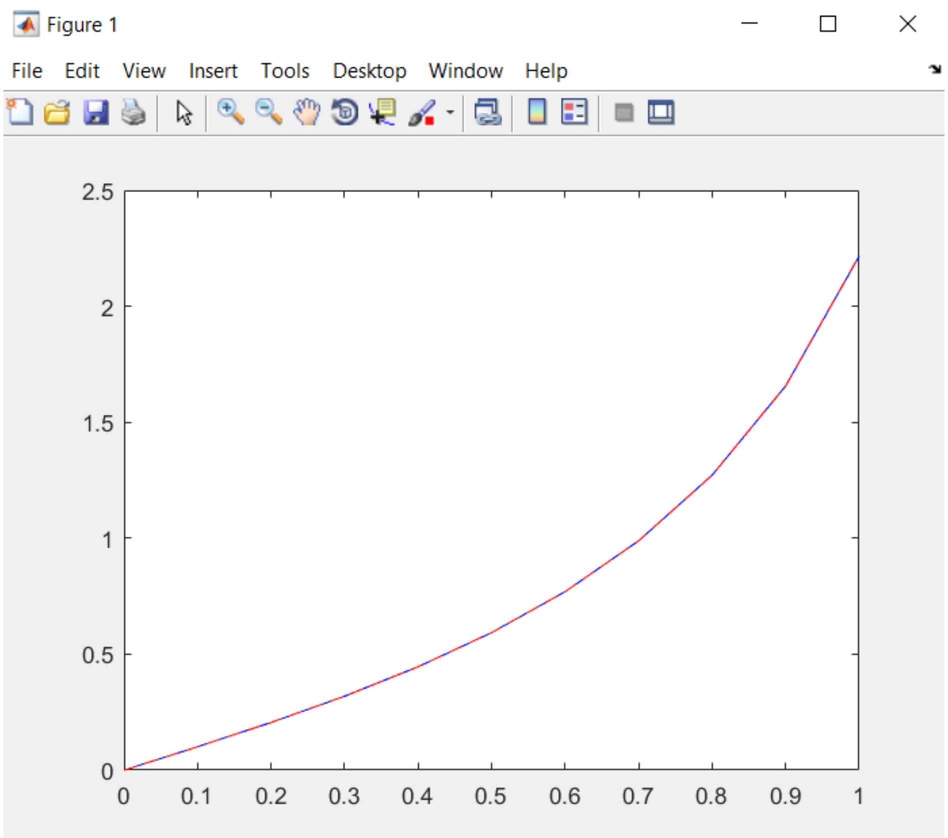
Pe intervalul [0,1] folosind metoda Runge Kutta de ordin 3.

timprungekutta3 = 0.0030

timpode23 = 0.0115

eroare = 0.0021

X =	Y =
0	0
0.1000	0.1007
0.2000	0.2054
0.3000	0.3187
0.4000	0.4457
0.5000	0.5930
0.6000	0.7696
0.7000	0.9889
0.8000	1.2723
0.9000	1.6573
1.0000	2.2179



Problema 3:

Să se rezolve problema Cauchy următoare:

$$\begin{cases} y'(x) = \dfrac{x-y}{x} \\ y(1) = 1 \end{cases}$$

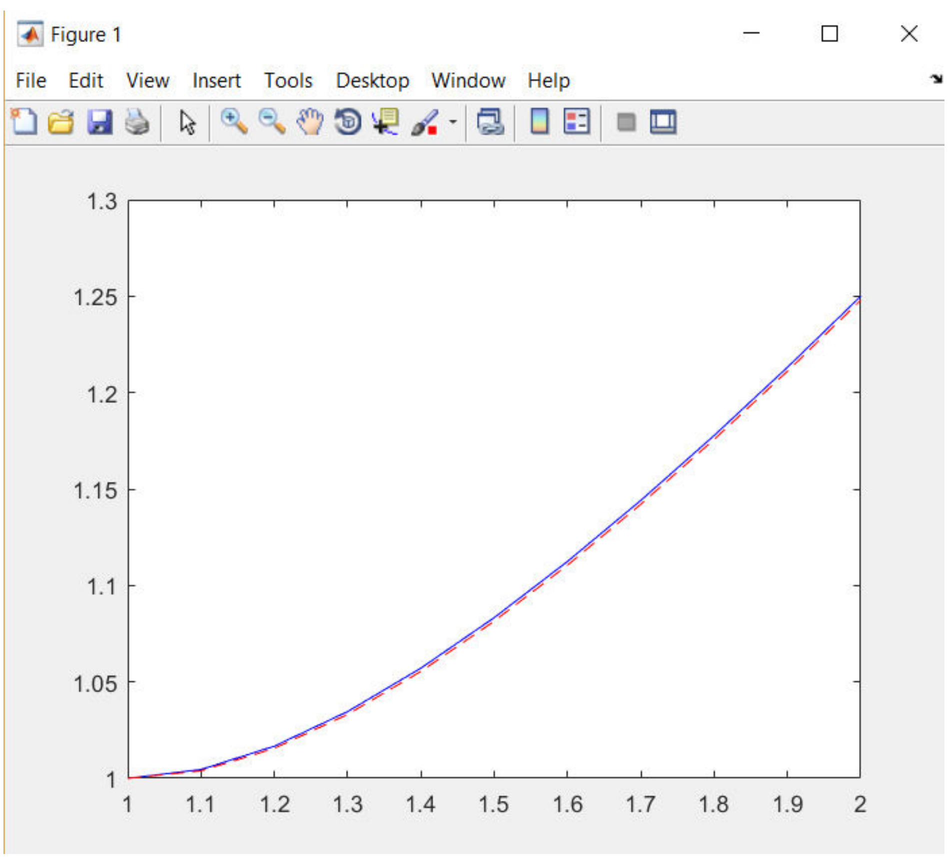
Pe intervalul [1,2] folosind metoda Runge Kutta de ordin 4.

timprungekutta4 = 0.0035

timpode45 = 0.3860

eroare = 0.0056

X =	Y =
1.0000	1.0000
1.1000	1.0039
1.2000	1.0155
1.3000	1.0331
1.4000	1.0554
1.5000	1.0815
1.6000	1.1106
1.7000	1.1421
1.8000	1.1757
1.9000	1.2111
2.0000	1.2479



Problema	Metoda numerică 1 Metoda Runge-Kutta	Funcția/comanda Matlab 2 ode23, ode45
Problema 1	timprungekutta2 =0.1537	timpode23 =0.5942
Problema 2	timprungekutta3 = 0.0030	timpode23 = 0.0115
Problema 3	timprungekutta4 = 0.0035	timpode45 = 0.3860
Timpul mediu	0.0534	0.3305666666666

Problema	Metoda numerică 1 Metoda Runge Kutta	Funcția/comanda Matlab 2 ode23, ode45
Problema 1	0.0089	0.0089
Problema 2	0.0021	0.0021
Problema 3	0.0056	0.0056
Eroarea absolută medie	0.0055333333333	0.0055333333333

6. Concluzii

Metodele Runge-Kutta sunt metode care presupun aproximarea soluției problemei. Acestea nu necesită calculul derivatelor, ci doar evaluarea funcției în puncte intermediare ale intervalului.

S-a observat că eroarea este foarte mică, atât metoda Runge-Kutta implementată în mediul Matlab de utilizator, cât și funcțiile mediului Matlab ode23 și ode45 fiind destul de precise.

7. Bibliografie

[1] - R.-E. Precup, UPT, 2019, Curs Matematici asistate de calculator

[2] - http://iota.ee.tuiasi.ro/~mgavril/Metode/Luc12/MetLuc12.html

[3] - R.-E. Precup, UPT, 2019, Matematici asistate de calculator – probleme de rezolvat

[4] - R.-E. Precup, Matematici asistate de calculator. Algoritmuri, Editura Orizonturi Universitare, Timişoara, 2007