

Metode numerice pentru probleme Cauchy

1. Ecuații diferențiale. Probleme Cauchy

1.2. Metode unipas

1.2.4. Metode de tip Runge – Kutta (continuare)

Considerăm problema Cauchy:

$$\begin{cases} y'(t) = f(t, y), & a \leq t \leq b \\ y(a) = y_a \end{cases} \quad (1.18)$$

și rețeaua de puncte: $t_i = a + ih, \quad i = 1, \dots, N.$

În general o metodă de tip **Runge – Kutta în r stadii** este o metodă unipas de forma:

$$\begin{aligned}
y_0 &= y_a \\
y_{i+1} &= y_i + h\phi(t_i, y_i, f, h), \quad i = 1, \dots, N-1
\end{aligned}
\tag{1.34a}$$

unde

$$\phi(t_i, y_i, f, h) = \sum_{s=1}^r c_s K_s
\tag{1.34b}$$

$$\begin{aligned}
K_1(t_i, y_i) &= f(t_i, y_i) \\
K_s(t_i, y_i) &= f\left(t_i + \mu_s h, y_i + h \sum_{j=1}^{s-1} a_{sj} K_j\right), \quad s = 2, 3, \dots, r
\end{aligned}
\tag{1.34c}$$

Se observă că pentru calculul lui K_s facem apel la K_{s-1}, \dots, K_2, K_1 , deci metoda descrisă mai sus este o metodă explicită. Pentru a păstra ordinul de aproximare a derivatei la $O(h^2)$ se impun condițiile:

$$\mu_s = \sum_{j=1}^{s-1} a_{sj}, \quad s = 2, 3, \dots, r$$

$$\sum_{s=1}^r c_s = 1$$
(1.35)

Pentru metodele cu $r \leq 4$ se poate arăta că au ordinul de exactitate $O(h^r)$.

În general unei metode Runge-Kutta cu r stadii i se asociază o un tabel, numit **tabelă Butcher**, care se poate scrie matricial:

$\boldsymbol{\mu}$	\mathbf{A}	μ_1	a_{11}	a_{12}	\cdots	a_{1r}
	\mathbf{c}^T	μ_2	a_{21}	a_{22}	\cdots	a_{2r}
		\vdots	\vdots	\vdots	\cdots	\vdots
		μ_r	a_{r1}	a_{r2}	\cdots	a_{rr}
			c_1	c_2	\cdots	c_r

Printre cele mai populare metode de tip Runge-Kutta amintim metoda lui Kutta având ordinul de exactitate 3 care are tabela Butcher:

0	0				$y_{i+1} = y_i + \frac{h}{6}(K_1 + 4K_2 + K_3)$
1/2	1/2	0			
1	-1	2	0	$K_1 = f(t_i, y_i)$	
		1/6	2/3	1/6	$K_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}hK_1\right)$
					$K_3 = f(t_i + h, y_i - hK_1 + 2hK_2)$

iar dintre cele cu 4 stadii amintim:

i) Metoda Runge-Kutta standard :

0	0				$y_{i+1} = y_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$	
$\frac{1}{2}$	$\frac{1}{2}$	0				$K_1 = f(t_i, y_i)$
$\frac{1}{2}$	0	$\frac{1}{2}$	0			
1	0	0	1	0	$K_3 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}hK_2\right)$	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$		$K_4 = f(t_i + h, y_i + hK_3)$

ii)Metoda Runge-Kutta-Gill :

0	0				$y_{i+1} = y_i + \frac{h}{6}(K_1 + (2 - \sqrt{2})K_2 + (2 + \sqrt{2})K_3 + K_4)$
$\frac{1}{2}$	$\frac{1}{2}$	0			$K_1 = f(t_i, y_i)$
$\frac{1}{2}$	$\frac{1}{2}(-1 + \sqrt{2})$	$1 + \frac{\sqrt{2}}{2}$	0		
1	0	$-\frac{\sqrt{2}}{2}$	$1 + \frac{\sqrt{2}}{2}$	0	$K_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}hK_1\right)$
	$\frac{1}{6}$	$\frac{1}{6}(2 - \sqrt{2})$	$\frac{1}{6}(2 + \sqrt{2})$	$\frac{1}{6}$	$K_3 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}(-1 + \sqrt{2})hK_1 + \left(1 + \frac{\sqrt{2}}{2}\right)hK_2\right)$
	$\frac{1}{6}$	$\frac{1}{6}(2 - \sqrt{2})$	$\frac{1}{6}(2 + \sqrt{2})$	$\frac{1}{6}$	$K_4 = f\left(t_i + h, y_i - \frac{\sqrt{2}}{2}hK_2 + \left(1 + \frac{\sqrt{2}}{2}\right)hK_3\right)$

iii)Metoda Kutta de trei optimi :

0	0				$y_{i+1} = y_i + \frac{h}{8}(K_1 + 3K_2 + 3K_3 + K_4)$
$\frac{1}{3}$	$\frac{1}{3}$	0			$K_1 = f(t_i, y_i)$
$\frac{2}{3}$	$\frac{1}{3}$	0	0		
1	1	-1	1	0	$K_2 = f\left(t_i + \frac{1}{3}h, y_i + \frac{1}{3}hK_1\right)$
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$	$K_3 = f\left(t_i + \frac{2}{3}h, y_i + \frac{1}{3}hK_2\right)$
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$	$K_4 = f(t_i + h, y_i + hK_1 - hK_2 + hK_3)$

Exemplul 1: Rezolvați problema inițială

$$y' = \frac{1}{t^2} - \frac{y}{t} - y^2, \quad 1 \leq t \leq 2, \quad y(1) = -1$$

care are soluția exactă $y(t) = -\frac{1}{t}$.

Aplicăm în continuare metoda Runge-Kutta standard. Programul matlab este:

```
%exemplu Runge Kutta standard
a=1;b=2;           %capetele intervalului
N=6;              %pasul rețelei
h=(b-a)/(N-1);   %numarul de noduri

y=zeros(N,1);%initializam vectorul solutie pentru Euler modificat
ye=zeros(N,1); %solu'ia exacta
y_e=zeros(N,1);%initializam vectorul solutie pentru Euler
y(1)=-1; ye(1)=-1; y_e=-1;           %conditia initiala
```



```

t=a:h:b; %pasii de timp

for i=2:N
    y_e(i)=ye(i-1)+h*fEx(t(i-1),y(i-1));

    K1=fEx(t(i-1),y(i-1));
    K2=fEx(t(i-1)+0.5*h,y(i-1)+0.5*h*K1);
    K3=fEx(t(i-1)+0.5*h,y(i-1)+0.5*h*K2);
    K4=fEx(t(i-1)+h,y(i-1)+h*K3);
    y(i)=y(i-1)+h/6*(K1+2*K2+2*K3+K4);

    ye(i)=-1/t(i);
end

plot(a:h:b,y,'.k') %reprezentam grafic solutia numerica
hold on
plot(a:h:b,ye,'b')
plot(a:h:b,y_e,'r')

```

unde am folosit funcția fEx.m dată mai jos

```

function rez=fEx(t,y)
rez=1/(t^2)-y/t-y^2;

```

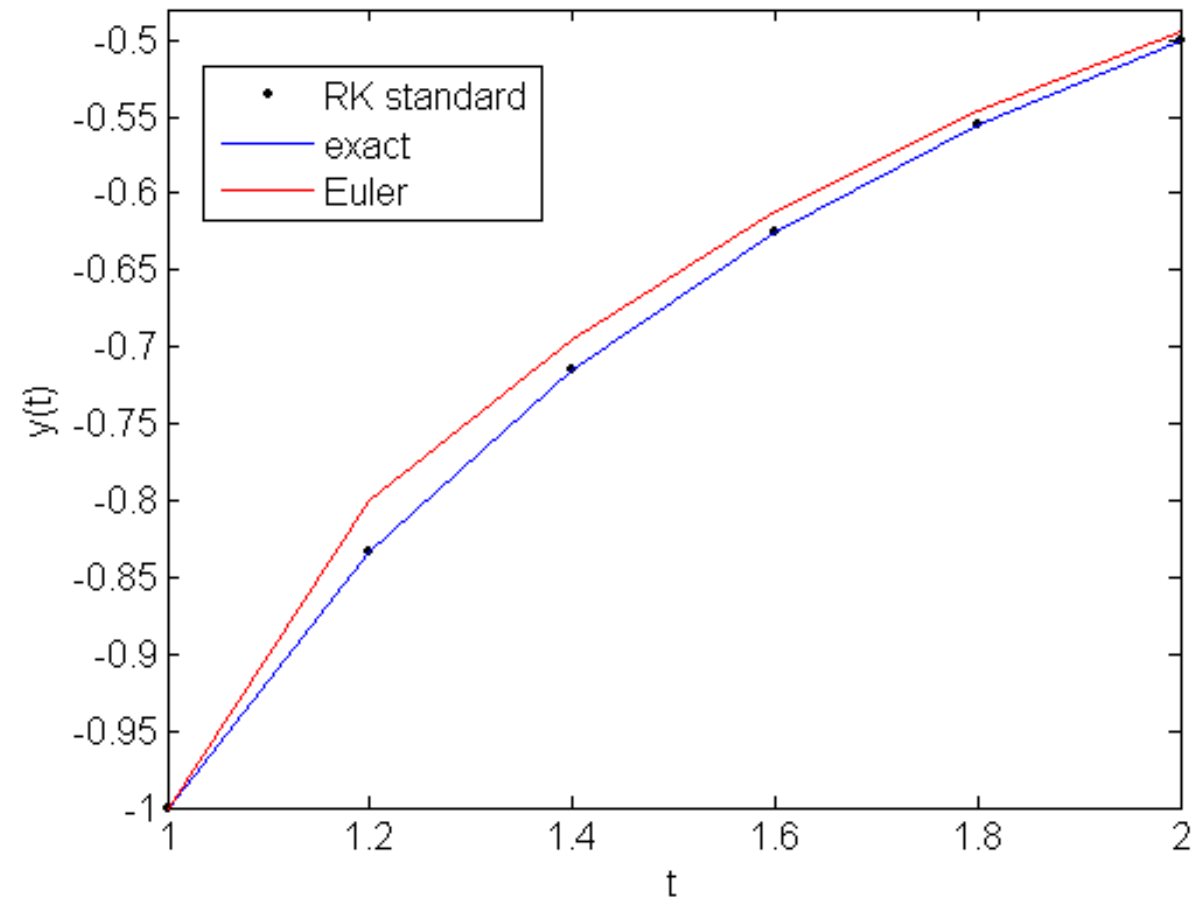


Figura 10. Soluția exactă și soluțiile numerice obținute folosind metoda lui Euler și metoda Runge-Kutta standard

Nodul	$y(\text{exact})$	y_{RK}	$ y - y_{RK} $
1.00	-1.000000	-1.000000	0.000000
1.20	-0.833311	-0.833333	0.000021
1.40	-0.714252	-0.714285	0.000033
1.60	-0.624958	-0.625000	0.000041
1.80	-0.555507	-0.555555	0.000048
2.00	-0.499945	-0.500000	0.000054

Tabel 4. Comparație între soluția analitică (y) și valorile obținute prin metoda Runge-Kutta standard (y_{RK})

Probleme cu valori pe frontieră

2.1. Preliminarii

O clasă a parte de probleme practice sunt modelate de **ecuații diferențiale cu valori pe frontieră** numite și **probleme cu valori pe frontieră** sau **probleme bilocale**. Forma generală a problemelor cu valori pe frontieră este dată de sistemul de ecuații diferențiale:

$$\frac{dy}{dx} = f(x, y), \quad f : [a, b] \times \mathbf{R}^d \rightarrow \mathbf{R}^d, \quad d \geq 2 \quad (2.1)$$

completat în capetele intervalului $[a,b]$ cu condiții de forma:

$$g(y(a), y(b)) = 0 \quad (2.2)$$

unde $g : \mathbf{R}^d \times \mathbf{R}^d \rightarrow \mathbf{R}^d$ este o aplicație ce poate fi neliniară. Ca și cazuri particulare putem avea de exemplu pentru $d = 2$:

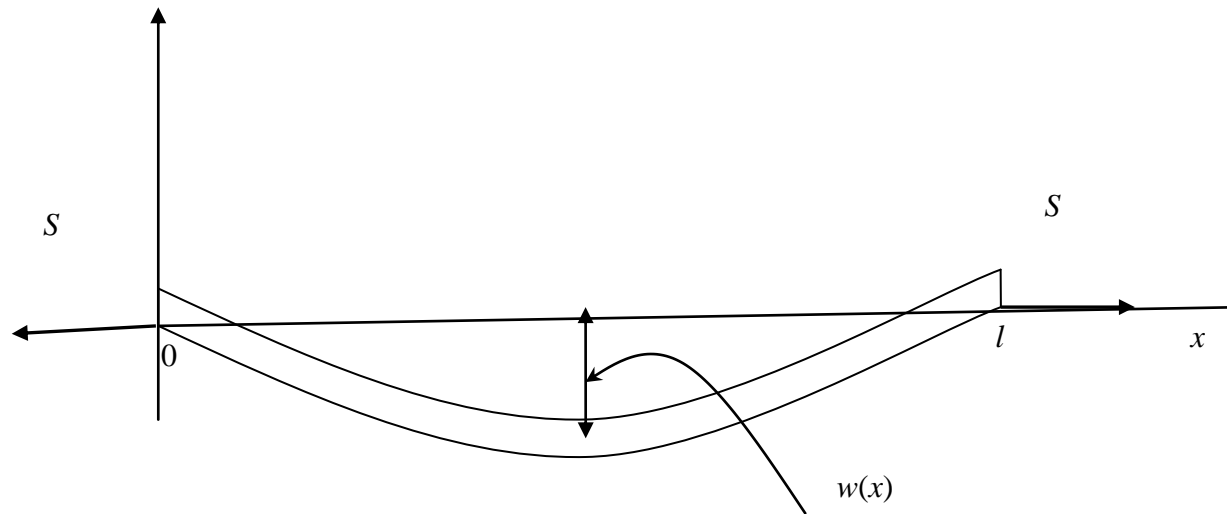
$$y(a) = A \quad (\text{Dirichlet})$$

$$\frac{dy}{dx}(a) = A \quad (\text{Neumann})$$

$$\frac{dy}{dx}(a) + \alpha y(a) = A, \quad \alpha < 0 \quad (\text{Robin})$$

unde A este o constantă.

Exemplul 1: Încovoierea unei bare sub propria sa greutate.



este descrisă de ecuația:

$$\frac{d^2 w}{dx^2}(x) = \frac{S}{EI} w(x) + \frac{qx}{2EI} (x - l)$$

unde $w(x)$ reprezintă deplasarea, l este lungimea barei, q este intensitatea încărcării barei, E este modulul de elasticitate, S reprezintă tensiunea la cele două capete, iar I este momentul de inerție al barei. Se observă că deplasările în capete sunt nule, așadar putem pune condițiile la frontieră:

$$w(0) = 0; \quad w(l) = 0.$$

Exemplul 2: Convecția mixtă într-un canal în prezența unei reacții chimice (Groșan și ceilalți, 2009)

$$U'' + \lambda \theta - \alpha = 0$$

$$\theta'' + K e^\theta = 0$$

$$U(0) = 0, \quad \theta(0) = r_T, \quad U(1) = 0, \quad \theta(1) = -r_T$$

unde U reprezintă viteza adimensională, θ reprezintă temperatura adimensională, iar λ , α , K și r_T sunt parametrii ce depind de datele problemei.

2.2. Rezultate de existență și unicitate

Considerăm pentru început câteva exemple (vezi Agratini și ceilalți, 2002). Fie problema bilocală:

$$y'' - y = 0, \quad y(0) = 0, \quad y(b) = \beta \tag{2.3}$$

cu soluția

$$y(x) = \beta \frac{\sinh x}{\sinh b}, \quad x \in [0, b] \tag{2.4}$$

Dacă modificăm problema (2.3) astfel

$$y''+y=0, \quad y(0)=0, \quad y(b)=\beta \quad (2.5)$$

atunci dacă:

$$b \neq k\pi, \quad k \in \mathbf{N}^* \text{ atunci } y(x) = \beta \frac{\sin x}{\sin b} \quad (2.6a)$$

$$b = k\pi, \quad k \in \mathbf{N}^* \text{ și } \beta \neq 0 \text{ atunci nu există soluție} \quad (2.6b)$$

$$b = k\pi, \quad k \in \mathbf{N}^* \text{ și } \beta = 0 \text{ atunci } y(x) = c \sin x, \quad c \in \mathbf{R} \quad (2.6c)$$

Se observă că pentru cazul (2.6c) avem o infinitate de soluții, așadar

$b = k\pi, \quad k \in \mathbf{N}^*$ este un punct critic.

Modificăm din nou ecuația astfel:

$$y'' + |y| = 0, \quad y(0) = 0, \quad y(b) = \beta \quad (2.7)$$

Dacă $b = \pi$ atunci

i) $\beta = 0$ există o infinitate de soluții de forma $y(x) = c \sin x$, $c \geq 0$

ii) $\beta > 0$ nu există soluții

iii) $\beta < 0$ există o soluție unică $y(x) = \beta \frac{\sinh x}{\sinh b}$

Dacă $b < \pi$ atunci există soluție unică:

$$y(x) = \begin{cases} 0, & \beta = 0 \\ \beta \frac{\sin x}{\sin b}, & \beta > 0 \\ \beta \frac{\sinh x}{\sinh b}, & \beta < 0 \end{cases} \quad (2.8)$$

Dacă $b > \pi$ atunci

i) $\beta = 0$ există o soluție unică $y(x) = 0$

ii) $\beta > 0$ nu există soluții

iii) $\beta < 0$ există două soluții

$$y_1(x) = \beta \frac{\sinh x}{\sinh b}, \quad x \in [a, b] \quad (2.9)$$

$$y_2(x) = \begin{cases} -\beta \frac{\sin x}{\sinh(b-x)}, & x \in [0, \pi] \\ \beta \frac{\sinh(x-\pi)}{\sinh(b-\pi)}, & x \in [\pi, b] \end{cases} \quad (2.10)$$

Se observă că o schimbare minoră a ecuației duce la o modificare substanțială a numărului soluțiilor și a formei acestora.

Considerăm în continuare o problemă bilocală neliniară de ordinul al doilea:

$$y''(x) = f(x, y, y'), \quad x \in [a, b] \quad (2.11)$$

cu condițiile pe frontieră

$$\begin{aligned} a_0 y(a) - a_1 y'(a) &= \alpha \\ b_0 y(b) - b_1 y'(b) &= \beta \end{aligned} \quad (2.12)$$

unde a_0 și a_1 nu sunt simultan nule, la fel și b_0 și b_1 . Presupunem că f este continuă pe $[a, b] \times \mathbf{R} \times \mathbf{R}$ și satisface condițiile Lipschitz uniforme:

$$\begin{aligned} \left| f(x, u_1^*, u_2) - f(x, u_1, u_2) \right| &\leq L_1 |u_1^* - u_1| \\ \left| f(x, u_1, u_2^*) - f(x, u_1, u_2) \right| &\leq L_2 |u_2^* - u_2| \end{aligned} \quad \forall x \in [a, b], \quad u_1, u_2 \in \mathbf{R} \quad (2.13)$$

O problema Cauchy pentru (2.11) are soluție unică pe intervalul $[a, b]$.

Considerăm în continuare problema Cauchy

$$u''(x) = f(x, u, u'), \quad x \in [a, b] \quad (2.14)$$

cu condițiile inițiale

$$\begin{aligned} a_0 u(a) - a_1 u'(a) &= \alpha \\ c_0 u(a) - c_1 u'(a) &= s \end{aligned} \quad (2.15)$$

Se observă că pentru a putea obține cele două condiții inițiale din (2.15) este necesar ca sistemul să fie nesingular, adică $c_0 a_1 - c_1 a_0 \neq 0$. Deoarece constantele c_0 și c_1 sunt arbitrare le fixăm astfel încât

$$c_0 a_1 - c_1 a_0 = 1 \quad (2.16)$$

Atunci condițiile inițiale (2.15) devin

$$\begin{aligned}u(a) &= a_1 s - c_1 \alpha \\ u'(a) &= a_0 s - c_0 \alpha\end{aligned}\tag{2.17}$$

unde parametrul s trebuie determinat. Deci pentru fiecare parametru s se obține o soluție $u(x, s)$ a problemei Cauchy (2.14) și (2.17). Pentru a rezolva problema cu valori pe frontieră (2.11) și (2.12) trebuie să avem:

$$\phi(s) = 0; \quad \phi(s) := b_0 u(b, s) + b_1 u'(b, s) - \beta\tag{2.18}$$

Teorema 11: Problema cu valori pe frontieră (2.11) și (2.12) are atâtea soluții distincte câte zerouri distincte are $\phi(s)$.

În general, pentru un sistem neliniar de d ecuații diferențiale problema bilocală cu condiții liniare pe frontieră are forma:

$$\begin{cases} \frac{dy}{dx} = f(x, y), & x \in [a, b], & y \in \mathbf{R}^d \\ Ay(a) + By(b) = \gamma \end{cases} \quad (2.19)$$

unde A și B sunt matrici pătratice de ordinul d cu elemente constante, iar γ este un vector de dimensiune d . Presupunem că $\text{rang}[A, B] = d$.

Problema cu valori inițiale asociată problemei cu valori pe frontieră (2.19) este:

$$\begin{cases} \frac{du}{dx} = f(x, u), & x \in [a, b], & u \in \mathbf{R}^d \\ u(a) = s \end{cases} \quad (2.20)$$

unde $s \in \mathbf{R}^d$ este un vector inițial de încercare. Se observă că pentru fiecare s se obține o soluție distinctă a problemei Cauchy (2.20). Atunci rezolvarea problemei cu valori pe frontieră se reduce la rezolvarea sistemului de ecuații neliniare:

$$\phi(s) = 0; \quad \phi(s) := As + Bu(b, s) - \gamma \quad (2.21)$$

Ca în cazul particular considerat anterior numărul soluțiilor problemei bilocale (2.19) este același cu numărul zerourilor lui $\phi(s)$.

2.3. Metoda shooting

Am văzut în secțiunea anterioară că o problemă cu valori pe frontieră se poate rezolva transformând-o într-o problemă Cauchy căreia i se mai adaugă un sistem de ecuații algebrice din care aflăm valorile inițiale ce lipsesc.

În cazul în care ecuația sau sistemul care trebuie rezolvat nu are toate condițiile inițiale precizate iar o parte dintre aceste condiții din condiții sunt date pe frontieră, atunci această problemă se poate rezolva folosind *metoda shooting*. Metoda poartă această denumire, în traducere -

problema tirului, deoarece este similară cu problema unui ofițer de artilerie care după ce a încadrat ținta cu două lovituri (una lungă și una scurtă) trebuie să dirijeze tirul în așa fel încât în cele din urmă proiectilul să atingă ținta. Bineînțeles că o înălțime intermediară primelor două va avea drept efect o lovitură mai apropiată de obiectiv. (vezi Gerald și Wheatley, 1999). Practic, trebuie alese valorile inițiale lipsă astfel încât să fi satisfăcute condițiile pe frontieră, iar pentru aceasta este nevoie de un mecanism de reajustare bazat pe distanța față de *țintă*. Există în general două moduri de aplicare a metodei shooting. Prin analogie cu problema aflării rădăcinii unei ecuații algebrice primul mod este similar

cu metoda înjumătățirii intervalului, iar cel de-al doilea mod cu metoda lui Newton sau metoda tangentei.

Înjumătățirea intervalului

Presupunem că avem o ecuație diferențială ordinară care are precizate condițiile inițiale într-un punct x_0 și condiția pe frontieră în punctul x_L . Vom rezolva ecuația considerând-o ca o problemă inițială în care valoarea inițială este aleasă arbitrar în punctul x_0 . Corectitudinea acestei alegeri este verificată rezolvând numeric ecuația, rezolvare în care putem folosi orice metodă pentru probleme cu condiții inițiale, și văzând cât de aproape este soluția de valoarea ei reală din x_L . De obicei, pentru o

alegere arbitrară, condiția pe frontieră în x_L nu este niciodată satisfăcută. De acest motiv se folosesc diferite procedee care să ne conducă la găsirea valorii inițiale corecte.

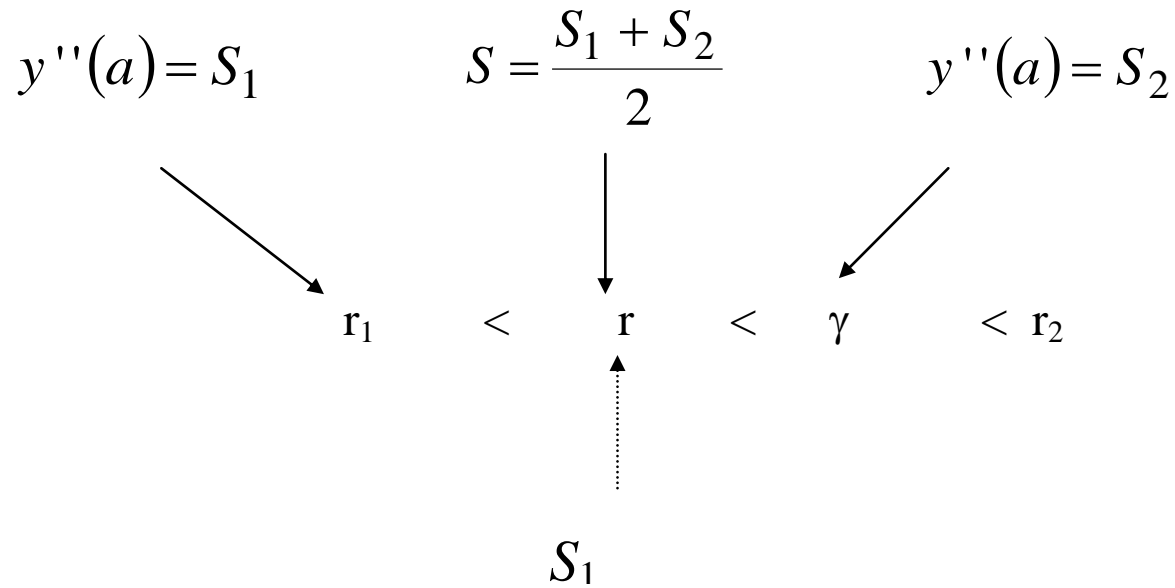
Presupunem că avem o ecuație diferențială ordinară de ordinul trei cu următoarele condiții la limită:

$$y(a) = \alpha, \quad y'(a) = \beta, \quad y'(b) = \gamma \quad (2.22)$$

Dorim să aflăm valoarea $y''(a) = \delta$ care să satisfacă condiția $y'(b) = \gamma$. În acest sens vom alege două valori $y''(a) = S_1$ și $y''(a) = S_2$ astfel încât prin rezolvarea ecuației cu aceste condiții inițiale obținem două valori $y'(b) = r_1$

și $y'(b) = r_2$ cu proprietatea că $r_1 \leq \gamma \leq r_2$. Vom rezolva apoi ecuația pentru $y''(a) = S$, unde

$$S = \frac{S_1 + S_2}{2} \tag{2.23}$$



Modul de alegere a noilor valori S_1 și S_2

și pentru care obținem $y''(a) = r$. Dacă $r < \gamma$ atunci S_1 va primi valoarea S , iar dacă $r \geq \gamma$ atunci S_2 va primi valoarea. Prin repetarea acestui procedeu intervalul din care alegem valoarea inițială se înjumătățește până când $y'(b)$ este suficient de apropiat de γ pentru S din acest interval.

Exemplu 3: Considerăm ecuația lui Blasius (vezi Oroveanu, 1967), ecuație diferențială care modelează scurgerea cu strat limită pe o placă plană:

$$f''' + f f'' = 0$$

$$f(0) = 0, f'(0) = 0, f'(\eta_\infty) = 1 \quad (2.24)$$

unde $f = f(\eta)$ iar prin η_∞ se înțelege o valoare mare de exemplu $\eta_\infty = 7$.

Ecuatia (2.24) este echivalentă cu sistemul de ecuații de ordinul întâi:

$$\begin{aligned}y_1' &= y_2; & y_2' &= y_3; & y_3' &= -y_1 y_3 \\ y_1(0) &= 0; & y_2(0) &= 0; & y_2(\eta_\infty) &= 1\end{aligned}$$

Pentru rezolvarea ecuației (2.24) considerăm două valori particulare pentru condiția inițială lipsă, $f''(0)$, și anume $S_1 = 0.1$ pentru care prin rezolvarea problemei inițiale prin metoda Runge – Kutta standard obținem $f'(\infty) = 0.356604$ și $S_2 = 0.7$ pentru care vom obține $f'(\infty) = 1.304993$. Așadar, am încadrat condiția inițială lipsă între două valori și putem aplica în continuare algoritmul metodei shooting.

Dăm în continuare programul scris în Matlab care rezolvă această problemă. Variabilele care apar în program au următoarele semnificații:

- a, b – capetele intervalului;
- S_1, S_2 – aproximațiile pentru $f''(0)$;
- yp – valoare lui f' în ∞ ;

- k – numărul de iterații;
A – matricea rezultatelor, $A = [\eta, f, f', f'']$;

```
function dy=fBlasius(t,y)
    dy=[y(2),y(3),-y(3)*y(1)];

function [x,y]=Kutta(f,a,b,N,y0)
h=(b-a)/(N-1) % pasul
y=zeros(N,length(y0)); %initializarea solutiei
y(1,:)=y0;
t=a:h:b;
x=t;
for i=2:N
    K1=feval(f,t(i-1),y(i-1,:));
    K2=feval(f,t(i-1)+0.5*h,y(i-1,:)+0.5*h*K1);
    K3=feval(f,t(i-1)+h,y(i-1,)-h*K1+2*h*K2);
    y(i,:)=y(i-1,:)+h/6*(K1+4*K2+K3);
end
```

```

%programul principal
format long;
a=0; b=7;N=701;
S1=0.1; S2=0.7;
yp=2;
k=0; %numarul de iteratii
while abs(yp-1)>0.001
    S=(S1+S2)/2;
    y0=[0,0,S];
    [x,y]=Kutta('fBlasius',a,b,N,y0);
    [m,n]=size(y);
    if y(m,2)<1
        S1=S;
    else
        S2=S;
    end;
    yp=y(m,2)
    k=k+1;
end;

A=[x',y];
disp(' eta      fsec      fprim      f      ')
disp(A);
plot(x,y(:,2));

```

În urma rulării programului, conform Tabelului 6, se observă apropierea de soluția reală pe măsură ce numărul de iterații crește

Numărul de iterații	$f'(\infty)$	$f''(0)$
1	0.89861923822595	0.400000
2	1.11116579667870	0.550000
3	1.00769999886139	0.475000
4	0.95393918747406	0.437500
5	0.98100355083518	0.456250
6	0.99439654224340	0.465625
7	1.00105931308546	0.470312
8	0.99773070678931	0.467968
9	0.99939570238780	0.469140

Tabelul 6. Valorile pentru $f'(\infty)$ și $f''(0)$ la diferite iterații

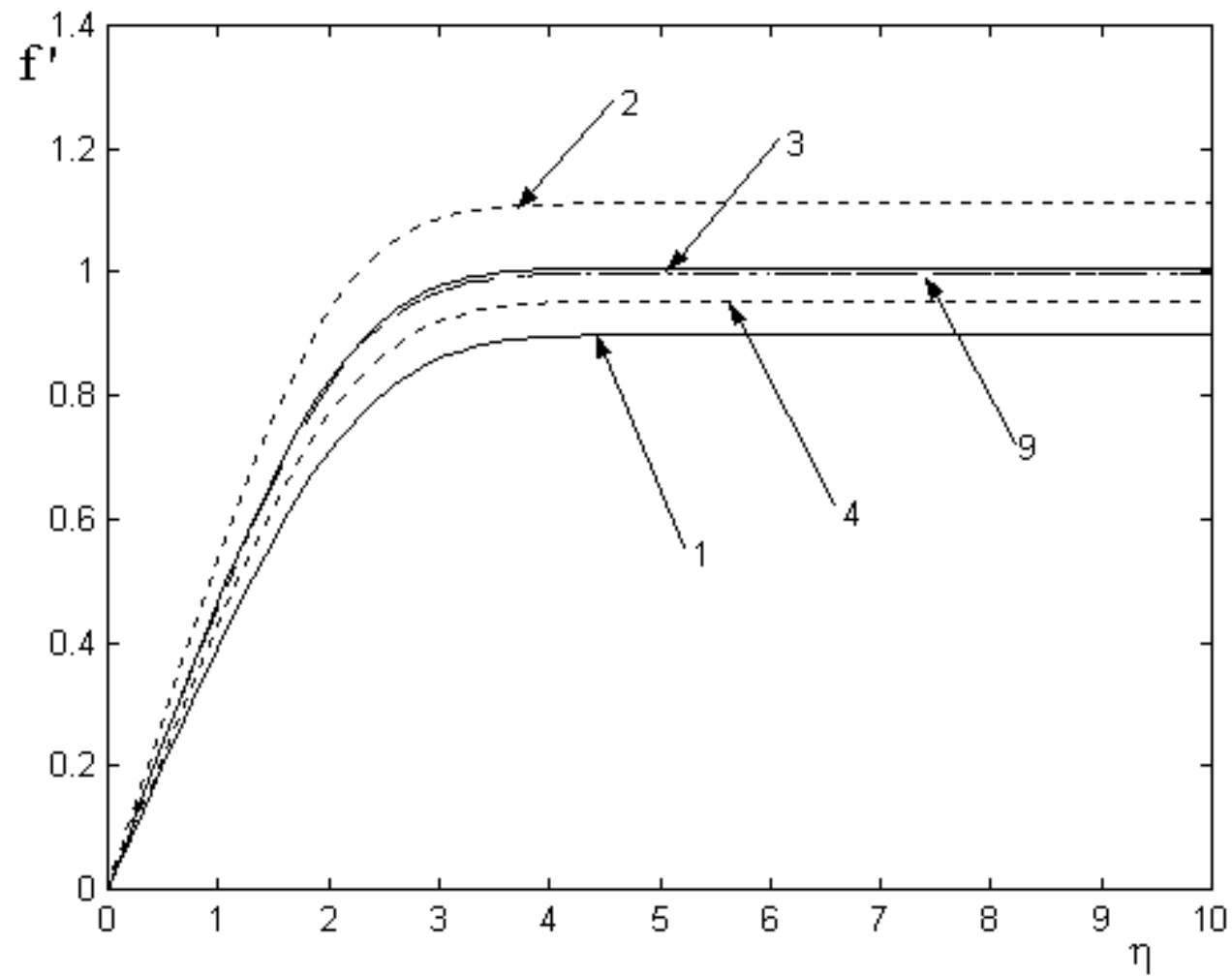


Figura 12. Aproximații succesive ale soluției.

Metoda lui Newton

Această metodă menține ecuația diferențială în formă neliniară și condiția la limită lipsă este căutată sistematic folosind metoda lui Newton. Vom arăta cum se obține algoritmul metodei shooting pentru o ecuație diferențială neliniară de ordinul trei.

Considerăm ecuația diferențială:

$$\frac{d^3 f}{dx^3} = g\left(x, f, \frac{df}{dx}, \frac{d^2 f}{dx^2}\right) \quad (2.25)$$

cu condițiile la limită

$$f(0) = 0, \quad \frac{df}{dx}(0) = 0, \quad \frac{df}{dx}(L) = f'_L \quad (2.26)$$

unde accentul notează derivata în raport cu variabila independentă x , iar L este limita dreaptă a intervalului în care este definită problema noastră.

Transformăm ecuația (2.25) într-un sistem de ecuații diferențiale ordinare de ordinul I:

$$\left\{ \begin{array}{l} \frac{df}{dx} = a \\ \frac{da}{dx} = b \\ \frac{db}{dx} = g(x, f, a, b) \end{array} \right. \quad (2.27)$$

iar condițiile pe frontieră se vor scrie în noua formă astfel:

$$f(0) = 0, \quad \frac{df}{dx}(0) = a(0) = 0, \quad \frac{d^2 f}{dx^2}(0) = b(0) = s \quad (2.28)$$

unde am notat cu s condiția inițială necunoscută. Problema care apare acum este găsirea lui s astfel încât soluția sistemului (2.27) cu condițiile inițiale (2.28) să satisfacă condițiile la limită (2.26), adică să fie respectată condiția dată în punctul L.

În acest sens, soluția problemei inițiale (2.27) și (2.28) trebuie să aibă în punctul L o valoare apropiată de f'_L , ceea ce, matematic, se scrie introducând funcția ϕ , care respectă condiția:

$$a(L, s) - f'_L = \phi(s) = 0 \quad (2.29)$$

Considerând că am aflat o valoare s^n pentru s la iterația n , pentru a afla criteriul de obținere al valorii lui s la iterația $n + 1$, s^{n+1} , dezvoltăm în serie Taylor funcția ϕ :

$$0 = \phi(s^{n+1}) = \phi(s^n) + (s^{n+1} - s^n) \frac{d\phi}{ds}(s^n) + \dots \quad (2.30)$$

de unde obținem

$$s^{n+1} = s^n - \frac{\phi(s^n)}{\frac{d\phi}{ds}(s^n)} \quad (2.31)$$

ceea ce este echivalent cu

$$s^{n+1} = s^n - \frac{a(L, s^n) - f'_L}{\frac{\partial a}{\partial s}(L, s^n)} \quad (2.32)$$

Necunoscuta care apare în relația (2.32) este în acest moment derivata lui a în raport cu s . Pentru a afla această valoare derivăm relațiile (2.27) și (2.28) în raport cu s și obținem:

$$\left\{ \begin{array}{l} \frac{d}{dx} \left(\frac{\partial f}{\partial s} \right) = \frac{\partial a}{\partial s} \\ \frac{d}{dx} \left(\frac{\partial a}{\partial s} \right) = \frac{\partial b}{\partial s} \\ \frac{d}{dx} \left(\frac{\partial b}{\partial s} \right) = \frac{\partial g}{\partial f} \frac{\partial f}{\partial s} + \frac{\partial g}{\partial a} \frac{\partial a}{\partial s} + \frac{\partial g}{\partial b} \frac{\partial b}{\partial s} \end{array} \right. \quad (2.33)$$

$$\frac{\partial f}{\partial s}(0) = 0, \quad \frac{\partial a}{\partial s}(0) = 0, \quad \frac{\partial b}{\partial s}(0) = 1 \quad (2.34)$$

Introducem notațiile:

$$F = \frac{\partial f}{\partial s}, \quad A = \frac{\partial a}{\partial s}, \quad B = \frac{\partial b}{\partial s} \quad (2.35)$$

și înlocuindu-le în ecuațiile (2.33) și (2.34) avem:

$$\left\{ \begin{array}{l} \frac{dF}{dx} = A \\ \frac{dA}{dx} = B \\ \frac{dB}{dx} = \frac{\partial g}{\partial f} F + \frac{\partial g}{\partial a} A + \frac{\partial g}{\partial b} B \end{array} \right. \quad (2.36)$$

$$F(0) = 0, \quad A(0) = 0, \quad B(0) = 1 \quad (2.37)$$

În acest moment putem da algoritmul metodei shooting pentru rezolvarea ecuațiilor (2.25) și (2.26):

Pasul 1: $s = s^1$

Pasul 2: rezolvăm ecuațiile (2.25) și (2.26) cu condiția $b(0) = s^1$

Pasul 3: rezolvăm ecuațiile (2.36) și (2.37) și reținem valoarea $A(L, s^1) = A(L)$

Pasul 4: aflăm $s^2 = s^1 - \frac{a(L, s^1) - f'_L}{A(L, s^1)}$

Pasul 5: repetăm pașii 1. – 4. cu $s^1 = s^2$ până se obține acuratețea dorită

Trebuie menționat faptul că pentru o alegere inițială a valorii s cât mai apropiată de valoarea reală, numărul pașilor se reduce semnificativ. De asemenea, nu pentru orice valoare inițială se obține o soluție, această valoare trebuind să se afle în vecinătatea valorii reale. Aflarea unei valori satisfăcătoare pentru începerea metodei se poate face prin experimentări numerice, prin verificarea mai multor valori de start.

Exemplu 4: Considerăm din nou ecuația lui Blasius scrisă sub forma

$$y_1' = y_2$$

$$y_2' = y_3$$

$$y_3' = -y_1 y_3$$

$$y_1(0) = 0; y_2(0) = 0; y_2(\eta_\infty) = 1$$

Conform algoritmului de mai sus avem de rezolvat sistemul:

$$y_1' = y_2$$

$$y_4' = y_5$$

$$y_2' = y_3$$

$$y_5' = y_6$$

$$y_3' = -y_1 y_3$$

$$y_6' = -y_3 y_4 - y_1 y_5$$

$$y_1(0) = 0; y_2(0) = 0; y_3(0) = s$$

$$y_4(0) = 0; y_5(0) = 0; y_6(0) = 1$$

iar pentru aflarea lui s folosim iterația

$$s^2 = s^1 - \frac{y_2(L, s^1) - 1}{y_5(L, s^1)}$$

Dăm în continuare programul matlab, unde am folosit funcția `Kutta.m` descrisă la exemplul anterior:

```
function dy=fBlasNewt(t,y)
dy=[y(2),y(3),-y(3)*y(1), y(5), y(6), -y(3)*y(4)-y(1)*y(6)];

format long;
a=0; b=5;N=251;
S=0.5;
yp=2;
k=0; %numarul de iteratii
while abs(yp-1)>0.0001
    y0=[0,0,S,0,0,1];
    [x,y]=Kutta('fBlasNewt',a,b,N,y0);
    S=S-(y(N,2)-1)/y(N,5);
    yp=y(N,2);
    k=k+1;
end;

disp([x',y]);
plot(x,y(:,2));
```

Numărul de iterații	$f'(\infty)$	$S = f''(0)$
1	1.655190	0.406241
2	0.907751	0.468075
3	0.997769	0.469644
4	0.999998	0.469645

Tabelul 7. Valorile pentru $f'(\infty)$ și $f''(0)$ la diferite iterații

Putem face observația că metoda lui Newton necesită mai puține iterații dar trebuie să rezolve un sistem de ecuații diferențiale dublu față de sistemul inițial. Dacă problema are mai multe soluții, atunci se aplică algoritmul de mai multe ori pentru diverse valori de pornire ale lui s .

2. 5. Rezolvatori Matlab

Probleme Cauchy. Rezolvatorii ODE

$$\begin{cases} y'(t) = f(t, y), & a \leq t \leq b \\ y(a) = y_a \end{cases}$$

Clasa de funcții ODE rezolvă probleme cu valori inițiale. Au sintaxa

```
[t, Y] = solver(odefun, tspan, y0, options, p1, p2, ...)
```

unde solver poate fi ode45, ode23, ode113, ode15s, ode23s, ode23t sau ode23tb.

Parametrii de ieșire sunt:

t – vectorul coloană al absciselor;

y – tabloul soluțiilor

Parametrii de intrare

`odefun` – specifică funcția din membrul drept al ecuației

`tspan` –intervalul de integrare.

`y0`– specifică valoarea inițială

`options` – opțiunile permit setarea unor parametri ai rezolvitorului și se creează cu `odeset`.

După `options` pot să apară parametri variabili, `p1`, `p2`, . . . care sunt transmiși funcției `odefun` la fiecare apel.

Funcția	Tip	Comentarii
<code>ode45</code>	Nonstiff	Metodă imbricată Runge-Kutta explicită cu ordinele 4 și 5. Are la bază perechea Dormand-Prince(DORPRI5).
<code>ode23</code>	Nonstiff	Metodă imbricată Runge-Kutta explicită cu ordinele 2 și 3. Are la bază perechea Bogacki-Shampine , BS(2,3).
<code>ode113</code>	Nonstiff	Metodă multipas explicită, cu ordin variabil de la 1 la 13. Are la bază metoda Adams-Bashford-Multon .
<code>ode15s</code>	Stiff	Metodă implicită cu mai mulți pași cu ordin variabil de la 1 la 15. Folosește formule de diferențiere numerică (NDFs) sau optional diferențe finite regresive (BDFs).
<code>ode23s</code>	Stiff	Metodă construită pe o pereche Rosenbrock modificată (cu un pas) de ordine 2 și 3.
<code>ode23t</code>	Stiff	Metoda implicită a trapezului, cu ordinele 2 și 3
<code>ode23tb</code>	Stiff	Algoritm Runge-Kutta implicit, ordinele 2 și 3 ce implică în primul stadiu metoda trapezului, iar în al doilea o metodă cu diferențe finite regresive de ordin 2.

Exemplu 2: Rezolvați ecuația pendulului folosind metoda ode45:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \sin \theta = 0, \quad \theta(0) = 0, \quad \frac{d\theta}{dt}(0) = 0.1$$

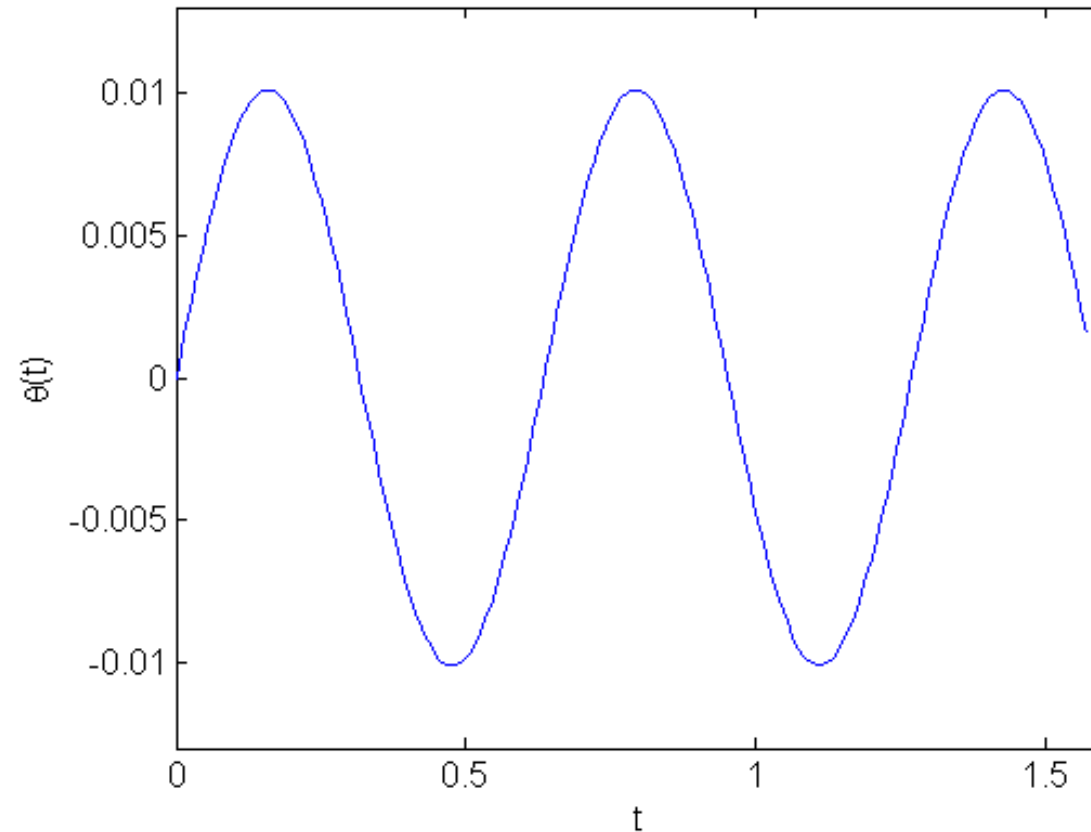
Notăm $y_1 = \theta$ și rescriem sistemul în forma:

$$\frac{dy_1}{dt} = y_2; \quad \frac{dy_2}{dt} = -\frac{g}{L} \sin y_1, \quad \text{cu condițiile} \quad y_1(0) = 0; \quad \frac{dy_1}{dt}(0) = 0.1$$

Programul Matlab este următorul:

```
%ecuatia pendulului
a=0;b=pi/2;%capetele intervalului de integrare
g=9.8;%acelaeratia gravitacionala
L=0.1;%lungimea pendulului
y0=[0,0.1];%conditiile initiale
options=odeset('RelTol',1e-8);%modificarea optiunilor
[t,y]=ode45('fPendul',[a,b],y0,options,g,L)%apelarea rezolvitorului
plot(t,y(:,1));
function dy=fPendul(t,y,flag,g,L)
dy=[y(2);-g/L*sin(y(1))];
```


Pentru integrare au fost necesare 169 de noduri, pasul minim a avut valoarea 0.008387, iar pasul maxim 0.013269.



Soluția numerică a ecuației pendulului.