Curs 7:
Interpolare cu funcții spline cubice.
Utilizarea funcțiilor Matlab pentru interpolare

Octavia-Maria BOLOJAN

octavia.nica@math.ubbcluj.ro

15 Noiembrie 2017

Funcții spline

- una dintre problemele care pot apărea frecvent în momentul în care încercăm să aproximăm o funcție cu un polinom de grad înalt este aceea că polinomul are minimul și maximul foarte apropiate, adesea dându-i astfel un caracter ondulatoriu
- acest lucru nu este dorit dacă polinomul va fi folosit pentru interpolare
- Vom defini funcția spline de gradul k, apoi vom studia cazul special al *funcției spline cubice*

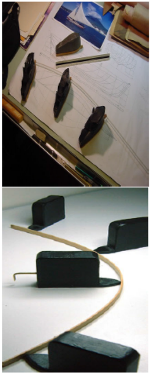
O funcție **spline** $s(x)$ de gradul k cu n noduri, $x_1 < x_2 < ... < x_n$, are următoarele proprietăți:

(i) funcția spline $s(x)$ este dată în intervalul $[x_i, x_{i+1}]$, $i = \overline{0, n}$ (unde $x_0 = -\infty, x_{n+1} = \infty$) de către un polinom care are gradul cel mult k (de regulă un polinom diferit în fiecare interval);

(ii) funcția spline $s(x)$ și toate derivatele ei de ordinul $1, 2, ..., k - 1$ sunt continue pe $(-\infty, \infty)$.

- În cazul $k = 3$, polinoamele din fiecare interval sunt cubice și prima și a doua derivată a lor sunt continue în punctele de capăt ale intervalelor
- Un astfel de set de polinoame formează o **funcție spline cubică**

$\rightarrow$ Funcțiile spline sunt utilizate în aplicații pe scară largă datorită proprietăților remarcabile
$\rightarrow$ Termenul "spline" provine din limba engleză si desemnează un instrument pentru trasarea curbelor netede prin puncte date



spline

florar

- Să presupunem că ne sunt date un set de puncte $x_1 < x_2 < ... < x_n$, nu neaparat echidistante, și un set de valori $f(x_1), f(x_2), ...f(x_n)$ în aceste puncte
- Dacă punctele sunt unite printr-un polinom de grad 3, rezultă **interpolarea spline cubică**
- Cele mai frecvent folosite funcții spline sunt cele de ordinul 3
- La aceste funcții, derivata a doua $f''(x_i)$ trebuie să fie continuă atât pe intervalul de interpolare $[x_i, x_{i+1}]$, cât și la capetele acestuia

Interpolarea spline cubică

Fie

- $I = [a, b] \subset \mathbb{R}$
- $\Delta_n = (a = x_1 < x_2 < \ldots < x_n = b)$ o diviziune a intervalului $[a, b]$
- f_i, $i = \overline{1, n}$, pentru $n \in \mathbb{N}^*$

  Se numește **funcție spline cubică**, corespunzătoare diviziunii Δ_n, o funcție $s : [a, b] \to \mathbb{R}$ care satisface condițiile:

$$\begin{cases} i)\ s\big|_{[x_i, x_{i+1}]} \text{ este polinom de gradul 3 pentru } i = \overline{1, n-1}; \\ ii)\ s \in C^2[a, b]. \end{cases} \tag{2.1}$$

  Se notează $S_3(\Delta_n)$ - **mulțimea funcțiilor spline cubice, corespunzătoare diviziunii** Δ_n.

- Funcția spline cubică asociată sistemului de puncte $(x_i, f_i)_{i=\overline{1,n}}$ este o funcție cubică corespunzătoare diviziunii Δ_n, $s : [a, b] \to \mathbb{R}$, cu proprietatea:

$$s(x_i) = f_i, \quad i = \overline{1, n}. \tag{2.2}$$

În condițiile formulate mai sus, $(\forall)\, i=\overline{1,n-1}$, funcția s se reprezintă în mod unic sub forma:

$$s(x)=c_{1i}+c_{2i}(x-x_i)+\frac{c_{3i}}{2}(x-x_i)^2+\frac{c_{4i}}{6}(x-x_i)^3,\ \ (\forall)\, x\in[x_i,x_{i+1}], \tag{2.3}$$

unde

$$\begin{cases} c_{1i}=f_i\\ c_{2i}=s_i\\ c_{3i}=\frac{1}{x_{i+1}-x_i}\left[6\frac{f_{i+1}-f_i}{x_{i+1}-x_i}-(4s_i+2s_{i+1})\right]\\ c_{4i}=\frac{6}{(x_{i+1}-x_i)^2}\left[(s_i+s_{i+1})-2\frac{f_{i+1}-f_i}{x_{i+1}-x_i}\right], \end{cases} \tag{2.4}$$

iar $s_1, s_2, ..., s_n$ sunt soluțiile sistemului

$$\begin{aligned} &(x_{i+1}-x_i)\,s_{i-1}+2\,(x_{i+1}-x_{i-1})\,s_i+(x_i-x_{i-1})\,s_{i+1} \\ = \ \ &3\left[\frac{f_i-f_{i-1}}{x_i-x_{i-1}}\,(x_{i+1}-x_i)+\frac{f_{i+1}-f_i}{x_{i+1}-x_i}\,(x_i-x_{i-1})\right], \end{aligned} \tag{2.5}$$

$(\forall)\, i=\overline{2,n-1}$.

- Pentru determinarea unică a spline-ului cubic, sistemul (2.4) trebuie completat cu condiții la capete. De exemplu, putem presupune că sunt cunoscute valorile pentru $s''(x_1)$ și $s''(x_n)$:

$$s''(x_1) = \alpha, \quad s''(x_n) = \beta \tag{2.6}$$

sau că sunt cunoscute valorile $s'(x_1), s'(x_n)$:

$$s_1 = s'(x_1) = \alpha, \quad s_n = s'(x_n) = \beta. \tag{2.7}$$

- Astfel, relațiile de mai sus sunt echivalente cu:

$$\begin{cases} 2s_1 + s_2 = 3\frac{f_2 - f_2}{x_2 - x_1} - \frac{\alpha}{2}(x_2 - x_1) \\ s_{n-1} + 2s_n = 3\frac{f_n - f_{n-1}}{x_n - x_{n-1}} + \frac{\beta}{2}(x_n - x_{n-1}). \end{cases} \tag{2.8}$$

Observație:

$\rightarrow$ Din (2.5) și (2.8) sau din (2.5) și (2.7) se pot determina $s_1, s_2, ..., s_n$
$\rightarrow$ Apoi, din (2.4) se determină $c_{1i}, c_{2i}, c_{3i}, c_{4i}$, pentru $i = \overline{1, n-1}$.
$\rightarrow$ Astfel, conform cu (2.3), funcția s este cunoscută pe $[a, b]$

$\rightarrow$ Pentru $f : [a, b] \rightarrow \mathbb{R}$, de două ori derivabilă, determinăm funcția spline cubică de interpolare asociată diviziunii Δ_n și funcției f, adică o funcție spline cubică $s \in S_3(\Delta_n)$, cu proprietățile:

$$s(x_i) = f(x_i), \quad (\forall)\, i = \overline{1, n-1}$$

și

$$s''(a) = f''(a), \quad s''(b) = f''(b)$$

sau

$$s'(a) = f'(a), \quad s'(b) = f'(b).$$

Exemplu

Fie

$$f : [0, 2\pi] \to \mathbb{R}, \; f(x) = \sin x + \cos x$$

și

$$x_i = (i-1)h, i = \overline{1, n},$$

unde $h = \frac{2\pi}{k}$, $k \in \mathbb{N}^*$. Notăm $f_i = f(x_i), i = \overline{1, n}$. Se cere determinarea funcției spline cubice s asociată sistemului de puncte $\{x_i, f_i\}_{i=\overline{1,n}}$ și reprezentarea în același sistem de coordonate a funcțiilor s și f pentru $n = 4$. Folosiți condițiile la capete (2.6).

Soluție.

- Pentru $n = 4$, se obține diviziunea

$$\Delta_4 = 0 = x_1 < x_2 < x_3 < x_4 = 2\pi$$

cu

$$x_1 = 0,\ x_2 = 2.0944,\ x_3 = 4.1888,\ x_4 = 6.2832.$$

- Valorile corespunzătoare ale funcției sunt:

$$f(x_1) = 1.0000,\ f(x_2) = 0.3660,\ f(x_3) = -1.3660,\ f(x_4) = 1.0000.$$

- Pentru a determina s_1, s_2, s_3, s_4, avem de rezolvat sistemul

$$\begin{bmatrix} 2.0000 & 1.0000 & 0 & 0 \\ 2.0944 & 8.3776 & 2.0944 & 0 \\ 0 & 2.0944 & 8.3776 & 2.0944 \\ 0 & 0 & 1.0000 & 2.0000 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} -0.6694 \\ -7.0981 \\ 1.9019 \\ 3.1503 \end{bmatrix}$$

(din relațiile (2.5)).

- Se obține soluția

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} 0.1108 \\ -0.8910 \\ 0.0640 \\ 1.5432 \end{bmatrix}.$$

Pentru spline-ul cubic, avem:

- pentru $x \in [x_1, x_2]$:

$$s(x) = 1 + 0.1108\,(x - x_1) - \frac{0.2280}{2}\,(x - x_1)^2 - \frac{0.2390}{6}\,(x - x_1)^3$$

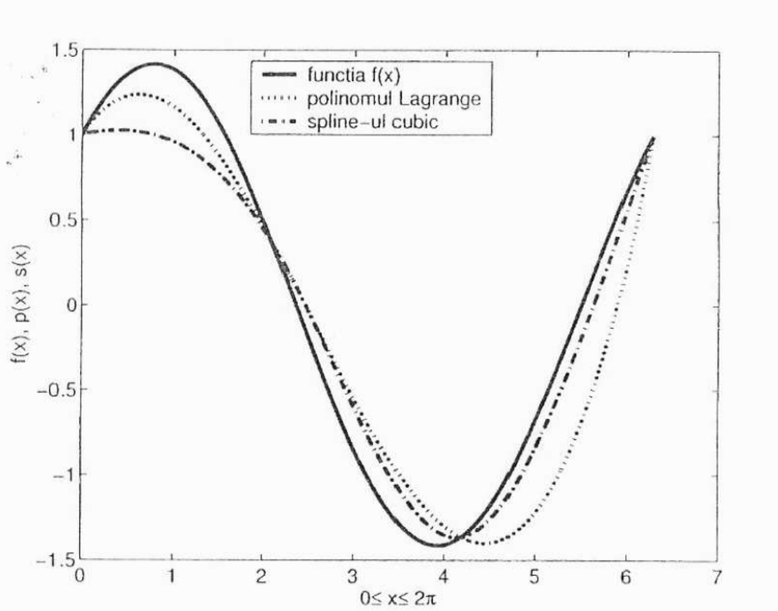
- pentru $x \in [x_2, x_3]$:

$$s(x) = 0.3660 - 0.8910\,(x - x_2) - \frac{0.7286}{2}\,(x - x_2)^2 + \frac{1.1312}{6}\,(x - x_2)^3$$

- pentru $x \in [x_3, x_4]$:

$$s(x) = -1.3660 + 0.0640\,(x - x_3) + \frac{1.6405}{2}\,(x - x_3)^2 - \frac{0.8921}{6}\,(x - x_3)^3 .$$

În următorul grafic, sunt reprezentate pentru $n = 4$, funcția, polinomul de interpolare și spline-ul cubic corespunzător.

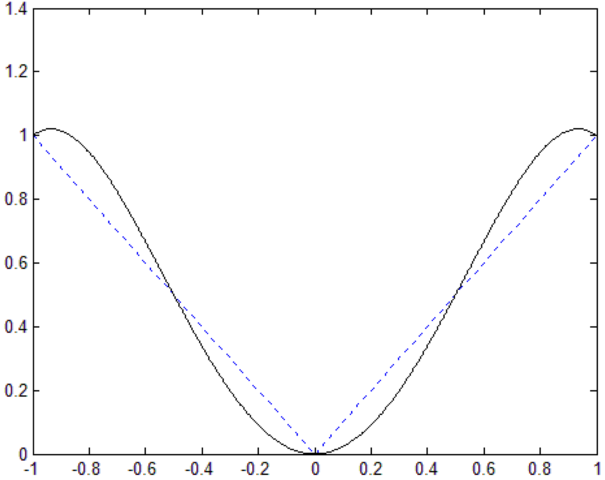


1. Interpolare Lagrange

→ Test convergență interpolare Lagrange

```
function lagmoduleq(n)
%pentru noduri echidistante - |x|
k=1:n;
xn=-1:2/n:1;
yn=abs(xn);
xg=-1:0.04:1;
yg=abs(xg);
ta=-1:2/(n*150):1;
ya=lagr(xn,yn,ta);
plot(xg,yg,':',ta,ya,'k-');
```

- pt $n = 4$ se obține reprezentarea:



→ Determinarea polinoamelor fundamentale Lagrange (funcția `PFL` calculeaza polinoamele fundamentale Lagrange, `x` - nodurile de interpolare, `t` - punctele în care se face evaluarea)

```
function Z=pfl(x,t)
m=length(x); n=length(t);
TT=zeros(m,n); Z=TT;
XX=zeros(m,m);
for i=1:m
 TT(i,:)=t-x(i);
end
for i=1:m
 XX(i,:)=x-x(i);
end
for j=1:m
 TX(j)=prod(XX([1:j-1,j+1:m],j));
end
for i=1:m
 Z(i,:)=prod(TT([1:i-1,i+1:m],:))/TX(i);
end
```

- funcția va fi aplelată într-un alt script având următoarele secvențe de cod:

```
function fi=lagr2(x,y,xi)
%LAGR2 - computes Lagrange interpolation polynomial
% using basic polynomials
% x,y - node coordinates
% xi - evaluation points
if nargin ~=3
error('wrong number of arguments')
end
z=pfl(x,xi);
fi=y*z;
```

- sau putem folosi secvența:

```
function fi=lagr(x,y,xi)
%LAGR - computes Lagrange interpolation polynomial
% x,y - coordinates of nodes
% xi - evaluation points
if nargin ~=3
 error('ilegal no.  of arguments')
end
[mu,nu]=size(xi);
fi=zeros(mu,nu);
np1=length(y);
for i=1:np1
 z=ones(mu,nu);
 for j=[1:i-1,i+1:np1]
 z=z.*(xi-x(j))/(x(i)-x(j));
 end;
 fi=fi+z*y(i);
end
```

2. Interpolarea spline cubică

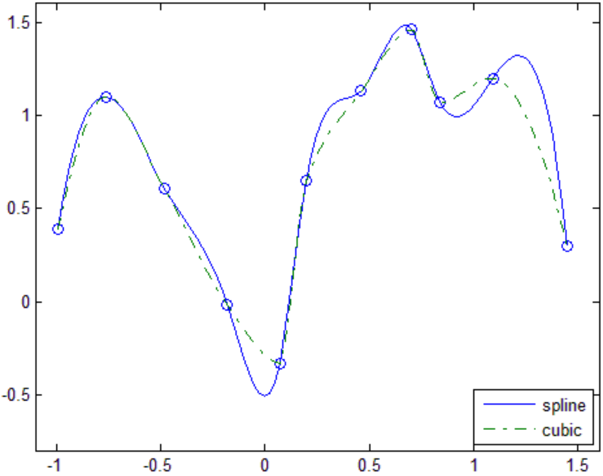
- Funcția Matlab **interp1()** interpolează între punctele date, găsind valorile unei funcții $f(x)$ pe baza acestor puncte
- sunt disponibile mai multe variante: `interp1(x,Y,xi);` `interp1(Y,xi);` `interp1(x,Y,xi,metoda)`
- `interp1(x,Y,xi,metoda)` permite realizarea interpolării folosind metode alternative:

  - 'nearest' : interpolarea prin puncte apropiate vecine
  - 'linear' : interpolarea lineară (implicită)
  - 'spline' : interpolare spline cubică
  - 'pchip' : interpolare cubică Hermite
  - 'cubic' : similară interpolării 'pchip'

etc

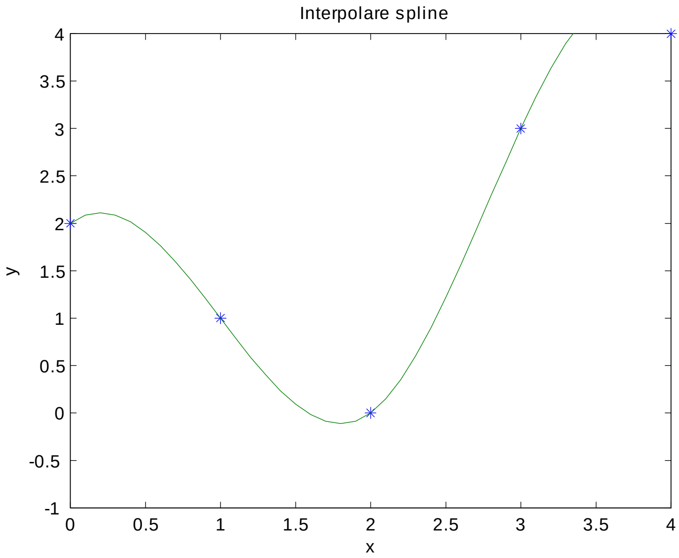
→ Exemplu:

```
x =[-0.99, -0.76, -0.48, -0.18, 0.07, 0.2, ...
 0.46, 0.7, 0.84, 1.09, 1.45];
y = [0.39, 1.1, 0.61, -0.02, -0.33, 0.65, ...
 1.13, 1.46, 1.07, 1.2, 0.3];
plot(x,y,'o'); hold on
xi=linspace(min(x),max(x),100);
ys=interp1(x,y,xi,'spline');
yc=interp1(x,y,xi,'cubic');
h=plot(xi,ys,'-',xi,yc,'-.');
legend(h,'spline','cubic',4)
axis([-1.1,1.6,-0.8,1.6])
```


1.5
1
0.5
0
-0.5
-1
-0.5
0
0.5
1
1.5
spline
cubic

→ Să se determine valorile de interpolare spline pentru sistemele de valori: x=0:4, y=[2 1 0 3 4], xx=0:0.2:4; și să se reprezinte graficul corespunzator interpolării.

```
x=0:4;
y=[2 1 0 3 4];
xx=0:0.1:4;
yy=spline(x,y,xx);
plot(x,y,'*',xx,yy)
axis([0 4 -1 4])
xlabel('x');ylabel('y');
title('Interpolare spline')
```


Interpolare spline
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
y
0
0.5
1
1.5
2
2.5
3
3.5
4
x