

Algebră Liniară și Optimizare

Curs 12

Conf. dr. Anca Ignat

Conf. dr. Corina Forăscu

May 16, 2026

Interpolare numerică

Presupunem că despre o funcție f cunoaștem doar valorile într-un număr finit de puncte.

Pornind de la aceste date, dorim să aproximăm funcția f într-un alt punct.

x	x_0	x_1	x_2	\cdots	x_{n-1}	x_n
f	y_0	y_1	y_2	\cdots	y_{n-1}	y_n

În tabelul de mai sus

$$f(x_i) = y_i, \quad i = 0, 1, \dots, n, \text{ cu } x_i \neq x_j \text{ pentru } i \neq j.$$

Dat un punct $x \neq x_i, i = 0, \dots, n$, dorim să aproximăm $f(x)$ cunoscând cele $(n + 1)$ perechi $(x_i, y_i), i = 0, \dots, n$.

Punctele x_i se numesc **noduri de interpolare**.

Forma Newton a polinomului de interpolare Lagrange

Propoziție

Fie $\ell_{k-1}(x, x_0, x_1, \dots, x_{k-1}, f), \ell_{k-1}(x, x_1, \dots, x_k, f) \in \Pi_{k-1}$ polinoamele de interpolare Lagrange pentru funcția f pe sistemul de noduri $\{x_0, x_1, \dots, x_{k-1}\}$, și, respectiv pe sistemul $\{x_1, x_2, \dots, x_k\}$.

Atunci

$$\begin{aligned} & \ell_k(x, x_0, x_1, \dots, x_k, f) \\ &= \frac{(x - x_0)\ell_{k-1}(x, x_1, \dots, x_k, f) - (x - x_k)\ell_{k-1}(x, x_0, \dots, x_{k-1}, f)}{x_k - x_0} \end{aligned}$$

Demonstrație: Se lasă ca exercițiu.

Forma Newton a polinomului de interpolare Lagrange

Considerăm următoarele probleme de interpolare:

$$\{(x_0, y_0), (x_1, y_1), \dots, (x_{k-1}, y_{k-1})\} \rightarrow \ell_{k-1}(x, x_0, x_1, \dots, x_{k-1}, f)$$

$$\{(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)\} \rightarrow \ell_k(x, x_0, x_1, \dots, x_k, f)$$

Ne interesează să găsim o formulă de trecere rapidă de la polinomul de interpolare pe k noduri la cel care are un nod în plus.

Forma Newton a polinomului de interpolare Lagrange

Deoarece polinomul de grad cel mult k :

$$q(x) = \ell_k(x, x_0, x_1, \dots, x_k, f) - \ell_{k-1}(x, x_0, x_1, \dots, x_{k-1}, f) \in \Pi_k.$$

are ca rădăcini punctele x_0, x_1, \dots, x_{k-1} ,

($q(x_i) = y_i - y_i = 0, i = 0, \dots, k-1$) avem relația

$$\ell_k(x, x_0, x_1, \dots, x_k, f) = \ell_{k-1}(x, x_0, x_1, \dots, x_{k-1}, f) + A \prod_{j=0}^{k-1} (x - x_j) \quad (11)$$

în care A este dat de relația

$$A = \frac{\ell_k(x_k, x_0, x_1, \dots, x_k, f) - \ell_{k-1}(x_k, x_0, x_1, \dots, x_{k-1}, f)}{\prod_{j=0}^{k-1} (x_k - x_j)} \quad (12)$$

Forma Newton a polinomului de interpolare Lagrange

Relația (12) se mai poate scrie astfel

$$\begin{aligned} A &= \frac{y_k}{\prod_{j=0}^{k-1} (x_k - x_j)} - \frac{\sum_{i=0}^{k-1} y_i \prod_{\substack{j=0 \\ j \neq i}}^{k-1} \frac{x_k - x_j}{x_i - x_j}}{\prod_{j=0}^{k-1} (x_k - x_j)} \\ &= \frac{y_k}{\prod_{j=0}^{k-1} (x_k - x_j)} - \sum_{i=0}^{k-1} \frac{y_i}{(x_k - x_i) \prod_{\substack{j=0 \\ j \neq i}}^{k-1} (x_i - x_j)} \end{aligned}$$

Forma Newton a polinomului de interpolare Lagrange

Așadar,

$$A_k = \sum_{i=0}^k \frac{y_i}{\prod_{\substack{j=0 \\ j \neq i}}^k (x_i - x_j)}. \quad (13)$$

Dacă definim :

$$[x_k]_f = y_k = f(x_k),$$

atunci

$$[x_0, x_1, \dots, x_k]_f = \frac{[x_1, x_2, \dots, x_k]_f - [x_0, x_1, \dots, x_{k-1}]_f}{x_k - x_0},$$

se numește **diferență divizată de ordin k** a funcției f pe nodurile $\{x_0, x_1, \dots, x_k\}$.

Forma Newton a polinomului de interpolare Lagrange

Propoziție

Pentru orice sistem de noduri $\{x_0, x_1, \dots, x_k\}$ și orice k are loc

$$[x_0, x_1, \dots, x_k]_f = \sum_{i=0}^k \frac{y_i}{\prod_{\substack{j=0 \\ j \neq i}}^k (x_i - x_j)} = \sum_{i=0}^k \frac{y_i}{(w_{n+1}(x_k))'}. \quad (14)$$

Demonstrație: Se face prin inducție. Pentru $k = 1$ avem:

$$[x_0, x_1]_f = \frac{y_0}{x_0 - x_1} + \frac{y_1}{x_1 - x_0} = \frac{[x_1]_f - [x_0]_f}{x_1 - x_0}.$$

Forma Newton a polinomului de interpolare Lagrange

Demonstrație(continuare):

Presupunem că relația (14) este adevărată pentru orice k și pentru orice sistem de noduri $\{x_0, x_1, \dots, x_k\}$.

Pentru $k + 1$ folosim relația de recurență și apoi aplicăm ipoteza inductivă:

$$[x_0, x_1, \dots, x_{k+1}]_f = \frac{[x_1, x_1, \dots, x_{k+1}]_f - [x_0, x_2, \dots, x_k]_f}{x_{k+1} - x_0}$$
$$= \frac{1}{x_{k+1} - x_0} \left(\sum_{i=1}^{k+1} \frac{y_i}{\prod_{\substack{j=1 \\ j \neq i}}^{k+1} (x_i - x_j)} - \sum_{i=1}^k \frac{y_i}{\prod_{\substack{j=1 \\ j \neq i}}^{k+1} (x_i - x_j)} \right)$$

Forma Newton a polinomului de interpolare Lagrange

Demonstrație(continuare):

Deci

$$[x_0, x_1, \dots, x_{k+1}]f = \frac{1}{x_{k+1} - x_0} \left\{ -\frac{y_0}{\prod_{\substack{j=0 \\ j \neq 0}}^k (x_0 - x_j)} + \frac{y_{k+1}}{\prod_{\substack{j=1 \\ j \neq k+1}}^{k+1} (x_{k+1} - x_j)} \right. \\ \left. + \sum_{i=1}^k \left[\frac{y_i}{\prod_{\substack{j=1 \\ j \neq i}}^k (x_i - x_j)} \left(\frac{1}{x_i - x_{k+1}} - \frac{1}{x_i - x_0} \right) \right] \right\}$$

Forma Newton a polinomului de interpolare Lagrange

Demonstrație(continuare): Sau, mai simplu

$$\begin{aligned} [x_0, \dots, x_{k+1}]_f &= \frac{y_0}{\prod_{\substack{j=0 \\ j \neq 0}}^k (x_0 - x_j)} + \frac{y_{k+1}}{\prod_{\substack{j=0 \\ j \neq k+1}}^{k+1} (x_{k+1} - x_j)} + \sum_{i=1}^k \frac{y_i}{\prod_{\substack{j=0 \\ j \neq i}}^{k+1} (x_i - x_j)} \\ &= \sum_{i=0}^{k+1} \frac{y_i}{\prod_{\substack{j=0 \\ j \neq i}}^{k+1} (x_i - x_j)} \end{aligned}$$

Așadar, prin inducție, relația (14) este adevărată pentru orice k .

Forma Newton a polinomului de interpolare Lagrange

Din definiție se observă că diferența divizată

$[x_0, x_1, \dots, x_k]_f$ nu depinde de ordinea nodurilor $\{x_0, x_1, \dots, x_k\}$.

Vom nota în continuare cu $\ell_k(x)$ polinomul de interpolare Lagrange pe nodurile $\{x_0, x_1, \dots, x_k\}$ pentru funcția f .

$$\begin{aligned}\ell_k(x) &= \ell_0(x) + [\ell_1(x) - \ell_0(x)] + \dots + [\ell_k(x) - \ell_{k-1}(x)] + \dots + [\ell_n(x) - \ell_{n-1}(x)] \\ &= y_0 + [x_0, x_1]_f(x - x_0) + \dots + [x_0, x_1, \dots, x_k]_f(x - x_0) \cdot \dots \cdot (x - x_{k-1}) \\ &\quad + \dots + [x_0, x_1, \dots, x_n]_f(x - x_0) \cdot \dots \cdot (x - x_{n-1}).\end{aligned}$$

Forma Newton a polinomului de interpolare Lagrange

Prin urmare, *forma Newton a polinomului de interpolare Lagrange* este dată de

$$\begin{aligned} \ell_n(x) = & y_0 + [x_0, x_1]_f(x - x_0) + [x_0, x_1, x_2]_f(x - x_0)(x - x_1) + \\ & + [x_0, x_1, \dots, x_n]_f(x - x_0) \cdot \dots \cdot (x - x_{n-1}). \end{aligned}$$

Schema Aitken de calcul al diferențelor divizate

În cele ce urmează, ne propunem să calculăm diferențele divizate

$$[x_0, x_1]_f, [x_0, x_1, x_2]_f, \dots, [x_0, x_1, \dots, x_n]_f$$

necesare construirii polinomului de interpolare Lagrange în forma Newton.

Procedeul folosește definiția recursivă a diferențelor divizate și se desfășoară în n pași.

Schema Aitken de calcul al diferențelor divizate

La pasul 1 se calculează numai diferențe divizate de ordinul 1:

$$[x_0, x_1]_f, [x_1, x_2]_f, \dots, [x_{n-1}, x_n]_f$$

În general, la pasul k se calculează diferențe divizate de ordin k :

$$[x_0, x_1, \dots, x_k]_f, [x_1, x_2, \dots, x_{k+1}]_f, \dots, [x_{n-k}, x_{n-k+1}, \dots, x_n]_f$$

La pasul n se calculează o singură diferență divizată de ordin n și anume

$$[x_0, x_1, \dots, x_n]_f.$$

Schema Aitken de calcul al diferențelor divizate

		Pas 1	...	Pas k	...	Pas n
x_0	y_0					
x_1	y_1	$[x_0, x_1]_f$				
x_2	y_2	$[x_1, x_2]_f$				
\vdots						
x_k	y_k	$[x_{k-1}, x_k]_f$		$[x_0, x_1, \dots, x_k]_f$		
\vdots				\vdots	\ddots	
x_{n-1}	y_{n-1}	$[x_{n-2}, x_{n-1}]_f$		$[x_{n-k-1}, \dots, x_{n-1}]_f$		
x_n	y_n	$[x_{n-1}, x_n]_f$		$[x_{n-k}, \dots, x_{n-1}]_f$		$[x_0, x_1, \dots, x_n]_f$

Schema Aitken de calcul al diferențelor divizate

Notăm cu

$$dd[i, k] := [x_i, x_{i+1}, \dots, x_{i+k}]_f$$

diferența divizată de ordin k , pe nodurile consecutive $\{x_i, x_{i+1}, \dots, x_{i+k}\}$, cu $i = 0, \dots, n - k$, $k = 1, \dots, n$, având $dd[i, 0] = y_i$, $i = 0, \dots, n$.

Schema lui Aitken se implementează astfel

$$dd[i, 0] = y_i, \quad i = 0, \dots, n;$$

for $k = 1, \dots, n$

for $i = 0, \dots, n - k$

$$dd[i, k] = \frac{dd[i + 1, k - 1] - dd[i, k - 1]}{x_{i+k} - x_i}.$$

Schema Aitken de calcul al diferențelor divizate

Putem face aceleași calcule folosind un singur vector, de exemplu rescriind vectorul y astfel:

for $k = 1, \dots, n$

for $i = n, \dots, k$

$$y_i = \frac{y_i - y_{i-1}}{x_i - x_{i-k}}.$$

La finalul acestei secvențe de program, vectorul y va conține elementele:

$$y_0, [x_0, x_1]_f, [x_0, x_1, x_2]_f, \dots, [x_0, x_1, \dots, x_n]_f$$

$$(y_k = [x_0, x_1, \dots, x_k]_f, k = 0, \dots, n).$$

Interpolare Newton pe noduri echidistante

Presupunem că nodurile de interpolare sunt echidistante:

$$x_i = x_0 + ih, \quad i = 0, 1, \dots, n.$$

În relația de mai sus, fie se dă h (distanța dintre două noduri succesive $h = x_{i+1} - x_i$), fie se precizează primul și ultimul nod, x_0 și x_n , iar h se calculează astfel $h = \frac{x_n - x_0}{n}$.

Astfel are loc

$$[x_i, x_{i+1}]_f = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{h}.$$

Interpolare Newton pe noduri echidistante

Se introduce noțiunea de **diferență finită de ordinul 1**:

$$\Delta f(x) = f(x + h) - f(x).$$

Pornind de la această definiție se pot introduce și diferențe finite de ordin superior:

$$\begin{aligned}\Delta^2 f(x) &= \Delta(\Delta f(x)) = \Delta(f(x + h) - f(x)) \\ &= f(x + 2h) - 2f(x + h) + f(x).\end{aligned}$$

În general, se pot introduce recursiv **diferențele finite de ordin k** :

$$\Delta^k f(x) = \Delta(\Delta^{k-1} f(x)) = \Delta^{k-1} f(x + h) - \Delta^{k-1} f(x).$$

Interpolare Newton pe noduri echidistante

Prin inducție după k , se poate deduce formula de calcul a diferențelor finite de ordin k folosind doar valorile funcției f .

$$\Delta^k f(x) = \sum_{i=0}^k (-1)^{k-i} C_k^i f(x + ih).$$

Observație: Dacă funcția f este polinom de grad m atunci $\Delta f(x)$ este polinom de grad $m - 1$, $\Delta^2 f(x)$ este polinom de grad $m - 2$, ș.a.m.d.

Prin urmare:

$$\Delta^k f(x) \equiv 0, \text{ pentru } k > m, f - \text{polinom de grad } m.$$

Diferențele divizate și diferențe finite

Legătura între diferențele divizate și cele finite:

$$[x_i, x_{i+1}]_f = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = \frac{\Delta f(x_i)}{h}.$$

$$[x_i, x_{i+1}, x_{i+2}]_f = \frac{[x_{i+1}, x_{i+2}]_f - [x_i, x_{i+1}]_f}{x_{i+2} - x_i} = \frac{\Delta^2 f(x_i)}{2! h^2}.$$

Prin inducție se poate arăta următoarea legătură între diferențele divizate de ordin k și cele finite:

$$[x_i, x_{i+1}, \dots, x_{i+k}]_f = \frac{\Delta^k f(x_i)}{k! h^k}.$$

Polinoame de interpolare pe noduri echidistante

Polinomul de interpolare Lagrange în forma Newton este:

$$\ell_n(x) = y_0 + [x_0, x_1]_f(x - x_0) + [x_0, x_1, x_2]_f(x - x_0)(x - x_1) \\ + \cdots + [x_0, x_1, \dots, x_n]_f(x - x_0) \cdots (x - x_{n-1}).$$

Considerăm că punctul de interpolare este de forma:

$$\bar{x} = x_0 + th.$$

și înlocuim diferențele divizate cu diferențe finite în forma Newton a polinomului de interpolare:

Rezultă:

$$(\bar{x} - x_0)(\bar{x} - x_1) \cdots (\bar{x} - x_{k-1}) = (x_0 + th - x_0) \cdots (x_0 + th - x_0 - (k-1)h) \\ = h^k t(t-1) \cdots (t-k+1).$$

Formula lui Newton progresivă

Polinomul de interpolare devine:

$$\begin{aligned} \ell_n(\bar{x}) = \ell_n(x_0 + th) = & y_0 + \Delta f(x_0)t + \Delta^2 f(x_0) \frac{t(t-1)}{2!} + \dots \\ & + \Delta^k f(x_0) \frac{t(t-1) \cdots (t-k+1)}{k!} + \dots \\ & + \Delta^n f(x_0) \frac{t(t-1) \cdots (t-n+1)}{n!}. \end{aligned}$$

Această relație poartă numele de **formula lui Newton progresivă** pe noduri echidistante.

Formula lui Newton regresivă

Considerăm polinomul de interpolare Lagrange pe nodurile în ordine inversă $\{x_n, x_{n-1}, \dots, x_0\}$:

$$\ell_n(x) = y_n + [x_n, x_{n-1}]_f(x - x_n) + [x_n, x_{n-1}, x_{n-2}]_f(x - x_n)(x - x_{n-1}) \\ + \dots + [x_n, x_{n-1}, \dots, x_0]_f(x - x_n) \cdots (x - x_1).$$

Dacă punctul de interpolare este de forma:

$$\bar{x} = x_n + th,$$

atunci, analog ca mai sus, obținem **formula lui Newton regresivă** pe noduri echidistante.:

$$\ell_n(x) = y_n + t \nabla f(x_n) + \nabla^2 f(x_n) \frac{t(t+1)}{2!} + \dots \\ + \nabla^k f(x_n) \frac{t(t+1) \cdots (t+k-1)}{k!} + \dots \\ + \nabla^n f(x_n) \frac{t(t+1) \cdots (t+n-1)}{n!}.$$

Schema lui Aitken pentru diferențe finite

Pentru formulele lui Newton progresive/regresive, avem nevoie de calculul următoarelor diferențe finite:

Pentru formula progresivă:

$$\Delta f(x_0), \Delta^2 f(x_0), \dots, \Delta^k f(x_0), \dots, \Delta^n f(x_0).$$

Pentru formula regresivă:

$$\Delta f(x_n), \Delta^2 f(x_{n-1}), \dots, \Delta^k f(x_{n-k}), \dots, \Delta^n f(x_0).$$

Metoda de calcul a acestor diferențe finite este similară schemei lui Aitken pentru diferențele divizate.

Schema lui Aitken pentru diferențe finite

	Pas 1	...	Pas k	...	Pas n
y_0					
y_1	$\Delta f(x_0)$				
y_2	$\Delta f(x_1)$				
\vdots					
y_k	$\Delta f(x_{k-1})$		$\Delta^k f(x_0)$		
\vdots			\vdots	\ddots	
y_{n-1}	$\Delta f(x_{n-2})$		$\Delta^k f(x_{n-k-1})$		
y_n	$\Delta f(x_{n-1})$		$\Delta^k f(x_{n-k})$...	$\Delta^n f(x_0)$

Funcții Spline

Fie nodurile

$$x_i \in [a, b], i = 0, \dots, n,$$

cu

$$a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b.$$

Se consideră funcția polinomială, continuă pe porțiuni

$$S(x) = P_i(x), \text{ pentru } x \in [x_i, x_{i+1}], \forall i = 0, \dots, n - 1.$$

Funcții Spline

Mai mult, funcția polinomială se poate scrie

$$S(x) = \begin{cases} P_0(x), & x \in [x_0, x_1), \\ P_1(x), & x \in [x_1, x_2), \\ P_2(x), & x \in [x_2, x_3), \\ \vdots \\ P_{n-2}(x), & x \in [x_{n-2}, x_{n-1}), \\ P_{n-1}(x), & x \in [x_{n-1}, x_n]. \end{cases}$$

unde $P_i(x)$, $i = 0, \dots, n$ sunt polinoame.

Funcția $S(x)$ se numește **funcție spline**.

Funcții spline liniare continue

Definiție

Funcția $S(x)$ definită mai sus se numește **funcție spline liniară continuă** dacă polinoamele

$$P_i(x), \quad i = 0, 1, \dots, n - 1$$

sunt polinoame de gradul 1 și $S(x) \in C[a, b]$, adică:

$$\lim_{\substack{x \rightarrow x_i \\ x < x_i}} S(x) = \lim_{\substack{x \rightarrow x_i \\ x > x_i}} S(x), \quad i = 1, \dots, n - 1.$$

Funcții spline liniare continue

Fie funcția $f : [a, b] \rightarrow \mathbb{R}$ pentru care se cunosc valorile:

$$y_i = f(x_i), \quad i = 0, 1, \dots, n.$$

Funcția spline liniară de interpolare S pentru funcția f îndeplinește condițiile de interpolare:

$$S(x_i) = y_i, \quad i = 0, \dots, n.$$

Ținând seamă că polinoamele $P_i(x)$ sunt polinoame de gradul 1 și $S(x)$ este continuă vom avea condițiile:

$$\begin{cases} P_i(x_i) = y_i, \\ P_i(x_{i+1}) = y_{i+1}, \quad i = 0, \dots, n-1. \\ P_i(x) - \text{polinom de gradul 1.} \end{cases}$$

Funcții spline liniare continue

Din aceste condiții rezultă

$$P_i(x) = \frac{x - x_i}{x_{i+1} - x_i} y_{i+1} + \frac{x_{i+1} - x}{x_{i+1} - x_i} y_i, \quad i = 0, \dots, n - 1.$$

$$S(x_k) = P_{k-1}(x_k) = P_k(x_k) = y_k, \quad k = 1, \dots, n - 1,$$

$$S(x_0) = P_0(x_0) = y_0, \quad S(x_n) = P_{n-1}(x_n) = y_n.$$

Funcții spline cubice de clasă C^2

Se consideră sistemul de noduri distincte aflate în intervalul $[a, b]$:

$$\Delta = \{a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b\}.$$

Funcția $S(x)$ asociată divizării Δ , care îndeplinește condițiile:

$$S(x) \in C^2[a, b],$$

și pentru care polinoamele

$$P_i(x), \quad i = 0, 1, \dots, n-1,$$

au gradul 3, se numește **funcție spline cubică**.

Funcții spline cubice de interpolare

Dată fiind o funcție $f : [a, b] \rightarrow \mathbb{R}$ cu valorile:

$$y_i = f(x_i), \quad i = 0, \dots, n,$$

se consideră funcția spline cubică $S(x)$ de interpolare ce satisface:

$$S(x_i) = y_i, \quad i = 0, \dots, n.$$

Pentru determinarea funcției spline cubice de interpolare observăm că polinoamele:

$$P_i(x) = \alpha_i x^3 + \beta_i x^2 + \gamma_i x + \delta_i, \quad x \in [x_i, x_{i+1}], \quad i = 0, \dots, n-1,$$

Funcții spline cubice de interpolare

Polinoamele P_i implică determinarea a celor $4n$ necunoscute:

$$\{\alpha_i, \beta_i, \gamma_i, \delta_i \mid i = 0, 1, \dots, n-1\},$$

pentru care se impun următoarele condiții

$$\begin{cases} n+1 \text{ condiții din relațiile de interpolare } S(x_i) = y_i, i = \overline{0, n}, \\ 3(n-1) \text{ condiții de continuitate pentru } S(x), S'(x) \text{ și } S''(x) \\ \quad \text{în nodurile } x_i, i = 1, \dots, n-1, \end{cases}$$

în total $4n - 2$ condiții.

Funcții spline cubice de interpolare

Se pot avea în vedere pentru adăugarea a două condiții suplimentare următoarele abordări :

- fixarea pantelor în extremitățile intervalului $[a, b]$. Se presupune că funcția f este derivabilă și se cunosc valorile $f'(a), f'(b)$. Se impun condițiile

$$S'(x_0) = P'_0(x_0) = f'(a), \quad S'(x_n) = P'_{n-1}(x_n) = f'(b);$$

- periodicitatea primelor două derivate:

$$f'(a) = f'(b) \quad (S'(x_0) = P'_0(x_0) = P'_{n-1}(x_n) = S'(x_n)),$$

$$f''(a) = f''(b) \quad (S''(x_0) = P''_0(x_0) = P''_{n-1}(x_n) = S''(x_n));$$

Funcții spline cubice de interpolare

- anularea derivatei secunde în capetele intervalului:

$$f''(a) = f''(b) = 0$$

$$(S''(x_0) = P_0''(x_0), \quad S''(x_n) = P_{n-1}''(x_n) = 0).$$

Funcțiile spline care îndeplinesc aceste condiții se numesc *funcții spline cubice normale*.

- derivata de ordinul al treilea a funcției S este continuă în punctele x_1 și x_{n-1} . Aceasta înseamnă că polinoamele P_0, P_1 respectiv P_{n-2}, P_{n-1} coincid. Acest tip de funcție spline se numește „not - a - knot” și este utilizat în MATLAB.

Funcții spline cubice de interpolare

Ne vom ocupa în continuare de determinarea funcției spline de interpolare în cazul în care cunoaștem prima derivată a funcției f în capetele intervalului de interpolare: $f'(a), f'(b)$. Recapitulând, vom avea următoarele condiții :

$$\left\{ \begin{array}{l} P_i(x_i) = y_i, \quad i = 0, \dots, n-1, P_{n-1}(x_n) = y_n - \text{interpolare,} \\ P_{i-1}(x_i) = P_i(x_i), \quad i = 1, \dots, n-1, - \text{continuitatea funcției } S, \\ P'_{i-1}(x_i) = P'_i(x_i), \quad i = 1, \dots, n-1, - \text{continuitatea primei derivate,} \\ P''_{i-1}(x_i) = P''_i(x_i), \quad i = 1, \dots, n-1, - \text{continuitatea derivatei secunde,} \\ P'_0(x_0) = f'(a), \quad P'_{n-1}(x_n) = f'(b). \end{array} \right.$$

Funcții spline cubice normale

Vom nota:

$$S''(x_i) = a_i, \quad i = 0, 1, \dots, n.$$

Ținând seama de faptul că funcția $S'' \in C[a, b]$ este o funcție liniară pe fiecare dintre intervalele $[x_i, x_{i+1}]$, rezultă că:

$$S''(x) = \frac{x - x_i}{h_i} a_{i+1} + \frac{x_{i+1} - x}{h_i} a_i, \quad x \in [x_i, x_{i+1}],$$

unde

$$h_i = x_{i+1} - x_i, \quad i = 0, 1, \dots, n - 1.$$

Funcții spline cubice

Din relațiile

$$S'(x) = \int S''(x) dx, \quad S(x) = \int S'(x) dx,$$

rezultă:

$$S'(x) = \frac{(x - x_i)^2}{2h_i} a_{i+1} - \frac{(x_{i+1} - x)^2}{2h_i} a_i + b_i,$$

unde

$$x \in [x_i, x_{i+1}], \quad b_i \in \mathbb{R}, \quad i = 0, 1, \dots, n - 1.$$

Funcții spline cubice

Așadar,

$$S(x) = \frac{(x - x_i)^3}{6h_i} a_{i+1} + \frac{(x_{i+1} - x)^3}{6h_i} a_i + b_i x + c_i,$$

unde

$$x \in [x_i, x_{i+1}], \quad b_i, c_i \in \mathbb{R}, \quad i = 0, 1, \dots, n - 1.$$

Deci

$$P(x) = \frac{(x - x_i)^3}{6h_i} a_{i+1} + \frac{(x_{i+1} - x)^3}{6h_i} a_i + b_i x + c_i.$$

Funcții spline cubice

Vom calcula funcția spline pentru cazul:

$$S'(a) = P'_0(x_0) = f'(a),$$

$$S'(b) = P'_{n-1}(x_n) = f'(b).$$

Impunând condițiile de interpolare și de continuitate vom obține:

$$P_i(x_i) = \frac{h_i^2}{6}a_i + b_ix_i + c_i = y_i,$$

$$P_i(x_{i+1}) = \frac{h_i^2}{6}a_{i+1} + b_ix_{i+1} + c_i = y_{i+1}, \quad i = 0, \dots, n-1.$$

Funcții spline cubice

Din relațiile de mai sus, calculăm b_i și c_i în funcție de $a_i, a_{i+1}, y_i, y_{i+1}$:

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(a_{i+1} - a_i),$$

$$c_i = \frac{x_{i+1}y_i - x_i y_{i+1}}{h_i} - \frac{h_i}{6}(x_{i+1}a_i - x_i a_{i+1}), \quad i = 0, \dots, n-1. \quad (*)$$

Avem:

$$P'_0(x) = \frac{(x - x_0)^2}{2h_0}a_1 - \frac{(x_1 - x)^2}{2h_0}a_0 + b_0,$$

$$P'_{n-1}(x) = \frac{(x - x_{n-1})^2}{2h_{n-1}}a_n - \frac{(x_n - x)^2}{2h_{n-1}}a_{n-1} + b_{n-1}.$$

Funcții spline cubice

Din condiția $S'(a) = P'_0(x_0) = f'(a)$ avem:

$$P'_0(x_0) = -\frac{h_0}{2}a_0 + b_0 = -\frac{2h_0}{6}a_0 - \frac{h_0}{6}a_1 + \frac{y_1 - y_0}{h_0} = f'(a).$$

Rezultă:

$$2h_0a_0 + h_0a_1 = 6 \left(\frac{y_1 - y_0}{h_0} - f'(a) \right) \quad (1)$$

Din condiția $S'(b) = P'_{n-1}(x_n) = f'(b)$ avem

$$P'_{n-1}(x_n) = \frac{h_{n-1}}{2}a_n + b_{n-1} = \frac{h_{n-1}}{6}a_{n-1} + \frac{h_{n-1}}{6}a_n + \frac{y_n - y_{n-1}}{h_{n-1}} = f'(b)$$

$$h_{n-1}a_{n-1} + 2h_{n-1}a_n = 6 \left(f'(b) - \frac{y_n - y_{n-1}}{h_{n-1}} \right). \quad (2)$$

Funcții spline cubice

Din condiția de continuitate a primei derivate a funcției spline cubice: $P'_{i-1}(x_i) = P'_i(x_i)$, $i = 1, \dots, n - 1$, ținând seama de:

$$P'_{i-1}(x) = \frac{(x - x_{i-1})^2}{2h_{i-1}}a_i - \frac{(x_i - x)^2}{2h_{i-1}}a_{i-1} + b_{i-1},$$

$$P'_i(x) = \frac{(x - x_i)^2}{2h_i}a_{i+1} - \frac{(x_{i+1} - x)^2}{2h_i}a_i + b_i.$$

Funcții spline cubice

Utilizând formulele (*) pentru b_{i-1} și b_i deduse mai sus, avem

$$P'_{i-1}(x_i) = \frac{h_{i-1}}{2}a_i + \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{h_{i-1}}{6}(a_i - a_{i-1}),$$

$$P'_i(x_i) = -\frac{h_i}{2}a_i + \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(a_{i+1} - a_i)$$

sau

$$h_{i-1}a_{i-1} + 2(h_{i-1} + h_i)a_i + h_ia_{i+1} = 6 \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right), \quad (3)$$

unde $i = 1, \dots, n-1$.

Funcții spline cubice

Sistemul linear format din ecuațiile (1), (3), (2), cu necunoscutele $\{a_0, a_1, \dots, a_n\}$, are forma:

$$Ha = f, \quad \text{cu } H \in \mathbb{R}^{(n+1) \times (n+1)}, \quad f \in \mathbb{R}^{n+1}.$$

Așadar, ecuațiile sistemului sunt:

$$2h_0a_0 + h_0a_1 = 6 \left(\frac{y_1 - y_0}{h_0} - f'(a) \right),$$

$$h_{i-1}a_{i-1} + 2(h_{i-1} + h_i)a_i + h_ia_{i+1} = 6 \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right),$$

$$h_{n-1}a_{n-1} + 2h_{n-1}a_n = 6 \left(f'(b) - \frac{y_n - y_{n-1}}{h_{n-1}} \right).$$

unde $i = 1, \dots, n - 1$,

Funcții spline cubice

Matricea asociată sistemului va fi

$$H = \begin{bmatrix} 2h_0 & h_0 & 0 & \cdots & 0 & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \cdots & 0 & 0 \\ 0 & h_1 & 2(h_1 + h_2) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & \cdots & h_{n-1} & 2h_{n-1} \end{bmatrix}$$

Funcții spline cubice normale

Vectorul termenilor liberi, va fi

$$f = \begin{bmatrix} 6 \left(\frac{y_1 - y_0}{h_0} - f'(a) \right) \\ \vdots \\ 6 \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right), \quad i = 1, \dots, n-1 \\ \vdots \\ 6 \left(f'(b) - \frac{y_n - y_{n-1}}{h_{n-1}} \right) \end{bmatrix}$$

Matricea H are diagonala dominantă atât pe linii cât și pe coloane, este simetrică și pozitiv definită prin urmare putem utiliza metoda Gauss-Seidel sau o metodă de relaxare pentru rezolvarea sistemului $Ha = f$.

Interpolare în sensul celor mai mici pătrate

Fie

x	x_0	x_1	x_2	\cdots	x_{n-1}	x_n
f	y_0	y_1	y_2	\cdots	y_{n-1}	y_n

$$f(x_i) = y_i, \quad i = 0, \dots, n.$$

Dorim să aproximăm funcția f printr-un polinom:

$$f(x) \approx S_f(x; a_0, a_1, \dots, a_m),$$

unde

$$S_f(x; a_0, a_1, \dots, a_m) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0.$$

Coeficienții a_0, a_1, \dots, a_m se găsesc rezolvând o problemă în sensul celor mai mici pătrate.

Interpolare în sensul celor mai mici pătrate

Coeficienții a_0, a_1, \dots, a_m se determină din problema:

$$\min \left\{ \sum_{r=0}^n (S_f(x_r; a_0, a_1, \dots, a_m) - y_r)^2 ; a_0, a_1, \dots, a_m \in \mathbb{R} \right\} .$$

(LSP)

Definim funcția:

$$g : \mathbb{R}^{m+1} \rightarrow \mathbb{R}_+,$$

$$g(a_0, a_1, \dots, a_m) = \sum_{r=0}^n (S_f(x_r; a_0, a_1, \dots, a_m) - y_r)^2 .$$

sau

$$g(a_0, a_1, \dots, a_m) = \sum_{r=0}^n \left(a_m x_r^m + \dots + a_k x_r^k + \dots + a_1 x_r + a_0 - y_r \right)^2 .$$

Interpolare în sensul celor mai mici pătrate

Derivata parțială în raport cu a_k este:

$$\begin{aligned}\frac{\partial g}{\partial a_k}(a_0, a_1, \dots, a_m) &= \\ &= 2 \sum_{r=0}^n \left(a_m x_r^m + \dots + a_k x_r^k + \dots + a_1 x_r + a_0 - y_r \right) x_r^k.\end{aligned}$$

Soluția problemei de minimizare (LSP) este obținută rezolvând sistemul de ecuații liniare, de dimensiune $(m + 1)$:

$$\frac{\partial g}{\partial a_k}(a_0, a_1, \dots, a_m) = 0, \quad k = 0, 1, \dots, m.$$

$$\sum_{r=0}^n \left(a_m x_r^m + \dots + a_k x_r^k + \dots + a_1 x_r + a_0 \right) x_r^k = \sum_{r=0}^n y_r x_r^k,$$

unde $k = 0, \dots, m$.

Interpolare în sensul celor mai mici pătrate

Adică:

$$a_0 \sum_{r=0}^n x_r^k + a_1 \sum_{r=0}^n x_r^{k+1} + \dots + a_m \sum_{r=0}^n x_r^{k+m} = \sum_{r=0}^n y_r x_r^k, \quad k = 0, \dots, m.$$

Constantele $\{a_0, a_1, \dots, a_m\}$ sunt soluția sistemului linear:

$$Ba = z,$$

unde

$$B \in \mathbb{R}^{(m+1) \times (m+1)}, B = (b_{kj})_{k,j=0}^m, \quad z \in \mathbb{R}^{m+1}, z = (z_k)_{k=0}^m$$

Elementele matricei B și ale vectorului z sunt:

$$b_{kj} = \sum_{r=0}^n x_r^{k+j}, \quad z_k = \sum_{r=0}^n y_r x_r^k, \quad k, j = 0, \dots, m.$$