

Interpolarea funcțiilor.

– Metode de interpolare pe porțiuni. –

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" București, Facultatea de Inginerie Electrică

Suport didactic pentru disciplina *Metode numerice*, 2017-2018

Cuprins

- 1 Introducere
 - Preliminarii
 - Formularea problemei interpolării

- 2 Metode de interpolare pe porțiuni
 - Interpolarea liniară pe porțiuni
 - Interpolarea Hermite

Preliminarii

Scrierea formală a unei probleme

$$\mathbf{y} = f(\mathbf{x}), \quad (1)$$

\mathbf{x} - datele problemei (parametri independenți);

\mathbf{y} - mărimile de interes ce se doresc a fi estimate.

De exemplu, f poate reprezenta:

- un **proces de măsurare** a mărimilor \mathbf{y} pentru o anumită stare complet caracterizată de \mathbf{x} ;
- un **program software complicat**, capabil să analizeze configurația caracterizată complet de datele \mathbf{x} și să calculeze printr-un algoritm de postprocesare mărimile \mathbf{y} .

Preliminarii

Formularea problemei (neriguros)

Se dă o funcție reprezentată prin date:

$(\mathbf{x}_k, \mathbf{y}_k)$, $k = 0, \dots, n$, unde $\mathbf{y}_k = f(\mathbf{x}_k)$.

Se dorește găsirea unei expresii analitice pentru o funcție g care să aproximeze aceste date adică

$g(\mathbf{x}_k) \approx \mathbf{y}_k$ sau chiar $g(\mathbf{x}_k) = \mathbf{y}_k$.

- **Interpolare** setului de date: g trece prin punctele mulțimii de date: $g(\mathbf{x}_k) = \mathbf{y}_k$
- **Aproximarea (regresia)** setului de date = g trece printre punctele mulțimii de date: $g(\mathbf{x}_k) \approx \mathbf{y}_k$

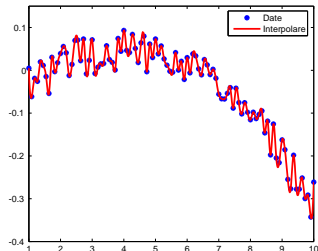
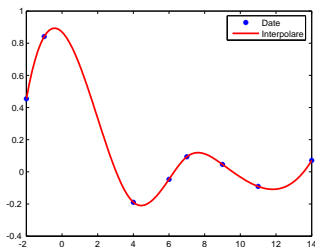
Preliminarii

Observații:

- 1 \mathbf{x}_k se numește și **rețea (grid) de discretizare**.
- 2 Interpolarea/aproximarea este utilă și dacă **funcția este reprezentată prin cod** = există un software capabil să calculeze $f(\mathbf{x})$ pentru orice \mathbf{x} dorit, dacă efortul de evaluare al lui f este mare.

Preliminarii

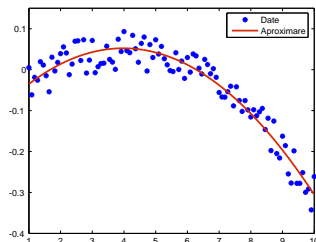
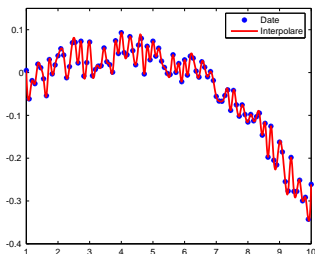
Exemple: interpolare



Interpolarea unui set de date. În cazul în care setul de date are foarte multe valori, interpolarea poate genera oscilații nedorite.

Preliminarii

Exemple: interpolare vs. aproximare



Avantajul aproximării: se diminuează erorile de măsurare din rezultatul final.

Precizări $f : ? \rightarrow ?$

- **Cazul scalar unidimensional (1D):** $f, g : [a, b] \rightarrow \mathbb{R}$.
- **Cazul vectorial unidimensional** $f : [a, b] \rightarrow \mathbb{R}^m$, $m > 1$
se reduce la m interpolări/aproximări 1D.
- **Cazul scalar bidimensional (2D)** $f, g : [a, b] \times [c, d] \rightarrow \mathbb{R}$
- **Cazul scalar n -dimensional (nD)** $f, g : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}$.
- **Cazul cel mai general** $f, g : \mathcal{D} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ se reduce la m situații de tip nD .

În cele ce urmează vom pp. cazul 1D.

Distanța dintre două funcții

Se dorește ca $g : [a, b] \rightarrow \mathbb{R}$ să aproximeze/interpoleze cât mai bine funcția $f : [a, b] \rightarrow \mathbb{R}$.

\Leftrightarrow

distanța dintre cele două funcții

$$d(f, g) = \|f - g\| \quad (2)$$

să fie cât mai mică.

Există mai multe procedee de definire a normei.

Distanța dintre două funcții

Procedee de definire a normei.

- **Aria dintre graficele celor două funcții**

$$d_1(f, g) = \frac{1}{b-a} \int_a^b |f(x) - g(x)| dx. \quad (3)$$

Dezavantaj: local, pot exista diferențe foarte mari între f și g .

- **Abaterea medie pătratică**

$$d_2(f, g) = \sqrt{\frac{1}{b-a} \int_a^b (f(x) - g(x))^2 dx}. \quad (4)$$

Același dezavantaj.

Distanța dintre două funcții

Procedee de definire a normei.

- **Abaterea maximă dintre cele două funcții**

$$d_3(f, g) = \max_{x \in [a, b]} |f(x) - g(x)|. \quad (5)$$

Din pdv al acurateții - este cea mai avantajoasă.

OBS: Niciuna din aceste norme nu se poate evalua.

Distanța dintre două funcții

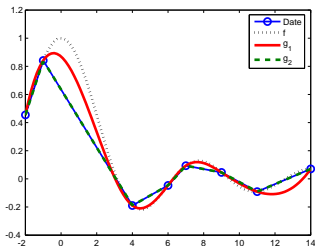
Normele discrete:

$$d_{1d}(f, g) = \sum_{k=0}^n |g(x_k) - f(x_k)|, \quad (6)$$

$$d_{2d}(f, g) = \sqrt{\sum_{k=0}^n (g(x_k) - f(x_k))^2}, \quad (7)$$

$$d_{3d}(f, g) = \max_{k=0, n} |g(x_k) - f(x_k)|. \quad (8)$$

Distanța dintre două funcții



Avantaj: pot fi evaluate cu ușurință.

Dezavantaj: se pierde posibilitatea evaluării acurateții între noduri. Mai mult $d_d(f, g_1) = 0$; $d_d(f, g_2) = 0$; \Rightarrow problemă prost formulată.

Formularea problemei interpolării

Se caută g pentru care $d_d(f, g) = 0$, unde f este cunoscută într-un număr finit de puncte $f(x_j) = y_j$.

Echivalent cu a impune **condițiile de interpolare**

$$g(x_j) = f(x_j), \quad j = 0, \dots, n, \quad (9)$$

\Leftrightarrow

$$g(x_j) = y_j, \quad j = 0, \dots, n. \quad (10)$$

Pentru a face ca problema să fie bine formulată matematic (soluția să existe și să fie unică) funcția g se caută în spațiul polinoamelor generalizate

\Leftrightarrow

g adică se caută de forma unei combinații liniare de m funcții φ_k , $k = 1, \dots, m$ numite **funcții de bază**:

$$g(x) = \sum_{k=0}^m c_k \varphi_k(x). \quad (11)$$

Formularea problemei interpolării

Funcțiile de bază **se aleg** înainte de rezolvarea propriu-zisă a problemei interpolării. Exemple:

- $\varphi_0(x) = 1, \varphi_1(x) = \sin x, \varphi_2(x) = \cos x, \varphi_3(x) = \sin(2x), \text{ etc.}$
- $\varphi_0(x) = 1, \varphi_1(x) = x, \varphi_2(x) = x^2, \varphi_3(x) = x^3, \text{ etc.}$

Cei m coeficienți c_k **se calculează** din impunerea condițiilor de interpolare:

$$\sum_{k=0}^m c_k \varphi_k(x_j) = y_j, \quad j = 0, \dots, n, \quad (12)$$

⇒ Sistem algebric linear cu $n + 1$ ecuații și $m + 1$ necunoscute.

Formularea problemei interpolării

Pentru buna formulare matematică se impune ca $m = n$ și

$$\Delta = \begin{vmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \cdots & \cdots & \cdots & \cdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{vmatrix} \neq 0. \quad (13)$$

$\Delta \neq 0 \Leftrightarrow x_j$ sunt distincte și φ_k sunt liniar independente.

Formularea problemei interpolării

Date:

- un tabel de valori (x_k, y_k) , $k = 0, \dots, n$, unde punctele rețelei de discretizare x_k sunt distincte două câte două;
- $n + 1$ funcții de bază liniar independente $\varphi_k(x)$, $k = 0, \dots, n$.

Se cer:

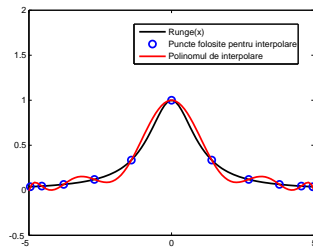
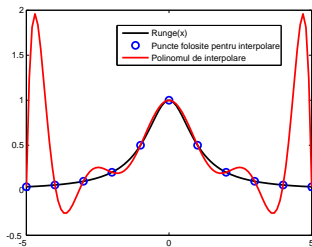
- coeficienții c_k , $k = 0, \dots, n$ pentru care sunt satisfăcute condițiile de interpolare $g(x_j) = y_j$, $j = 0, \dots, n$ unde $g(x) = \sum_{k=0}^n c_k \varphi_k(x)$ este polinomul de interpolare al datelor din tabel.

Interpolarea globală - concluzii

- Funcțiile de bază $\varphi_k(x)$ au o singură expresie pe tot domeniul de definiție \Rightarrow
 \Rightarrow polinomul de interpolare $g(x)$ are o singură expresie pe tot domeniul de definiție.

Dezavantaj: efectul Runge - oscilații la capete.

Remediu: Interpolarea Cebyshev



Interpolarea globală - concluzii

- Efectul Runge poate fi evitat dacă nodurile gridului de discretizare se aleg în conformitate cu rădăcinile polinoamelor Cebyshev.

Dezavantaj: funcția trebuie să fie definită prin cod, dar acest lucru nu este posibil întotdeauna.

Remediul: Interpolarea pe porțiuni.

- Funcțiile de bază $\varphi_k(x)$ sunt "funcții cu acoladă", au expresii diferite pe intervalele care discretizează domeniul de definiție \Rightarrow
 \Rightarrow polinomul de interpolare $g(x)$ este o "funcție cu acoladă".

Interpolarea liniară pe porțiuni

$\varphi_k(x)$ sunt polinoame Lagrange $l_k(x)$ definite pe porțiuni.

$$l_k(x_k) = 1$$

$$l_k(x_j) = 0 \text{ pentru } j \neq k$$

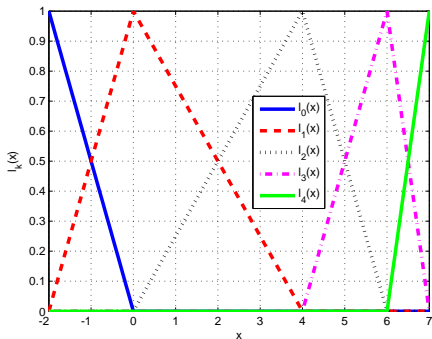
$$l_0(x) = \begin{cases} \frac{x-x_1}{x_0-x_1} & \text{dacă } x \in [x_0, x_1] \\ 0 & \text{dacă } x \in (x_1, x_n] \end{cases}$$

$$l_k(x) = \begin{cases} \frac{x-x_{k-1}}{x_k-x_{k-1}} & \text{dacă } x \in [x_{k-1}, x_k) \\ \frac{x-x_{k+1}}{x_k-x_{k+1}} & \text{dacă } x \in [x_k, x_{k+1}] \\ 0 & \text{dacă } x \in [x_1, x_{k-1}) \cup (x_{k+1}, x_n] \end{cases} \quad k = 2, n-1$$

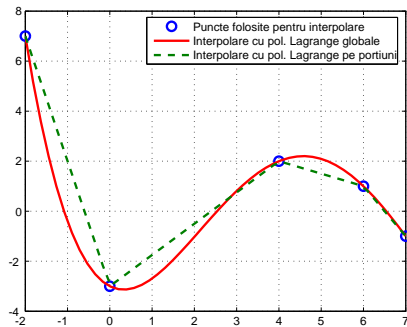
$$l_n(x) = \begin{cases} \frac{x-x_{n-1}}{x_n-x_{n-1}} & \text{dacă } x \in [x_{n-1}, x_n] \\ 0 & \text{dacă } x \in [x_0, x_{n-1}) \end{cases}$$

$$g(x) = \sum_{k=0}^n c_k l_k(x) \quad (14)$$

Interpolarea liniară pe porțiuni



Polinoame Lagrange pe porțiuni.



Polinoame de interpolare.

Interpolarea liniară pe porțiuni

funcție `interpolare_lpp($n, x, y, xcrt$)`

; evaluează polinomul de interpolare liniară pe porțiuni în `xcrt`

; declarații - parametri de intrare

întreg `n` ; dimensiunea problemei - nr. de intervale

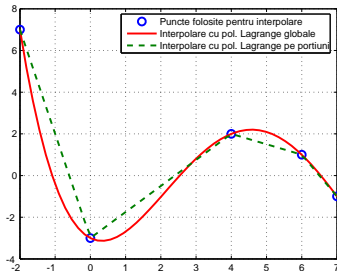
tablou real `x[n], y[n]`; tabelul de valori, indici de la zero

real `xcrt` ; punctul de evaluat

`k = cauta($n, x, xcrt$)`

întoarce $(y_{k+1} - y_k) / (x_{k+1} - x_k) * (xcrt - x_k) + y_k$

Interpolarea liniară pe porțiuni



Dezavantaj:

funcția de interpolare nu este derivabilă în noduri.

Remediu:

creșterea gradului polinoamelor care se folosesc pe porțiuni.

Interpolarea Hermite

- interpolarea unei funcții pe o rețea de noduri x_k în care se cunosc valorile funcției y_k și valorile derivatelor acesteia y'_k .

x	x_0	x_1	\cdots	x_k	\cdots	x_n
y	y_0	y_1	\cdots	y_k	\cdots	y_n
y'	y'_0	y'_1	\cdots	y'_k	\cdots	y'_n

Condiții de interpolare: $\forall k = 0, \dots, n-1$:

$$\begin{cases} g(x_k) = y_k, \\ g(x_{k+1}) = y_{k+1}, \\ g'(x_k) = y'_k, \\ g'(x_{k+1}) = y'_{k+1}. \end{cases} \quad (15)$$

\Rightarrow polinom de interpolare de gradul 3 pe fiecare subinterval.

Interpolarea Hermite

$$g(x) = c_{0k} + c_{1k}(x - x_k) + c_{2k}(x - x_k)^2 + c_{3k}(x - x_k)^3, \quad (16)$$

$$g'(x) = c_{1k} + 2c_{2k}(x - x_k) + 3c_{3k}(x - x_k)^2, \quad x \in [x_k, x_{k+1}].$$

$$\begin{cases} g(x_k) = y_k, \\ g(x_{k+1}) = y_{k+1}, \\ g'(x_k) = y'_k, \\ g'(x_{k+1}) = y'_{k+1}. \end{cases} \quad (17)$$

⇒

$$\begin{cases} c_{0k} = y_k, \\ c_{0k} + c_{1k}(x_{k+1} - x_k) + c_{2k}(x_{k+1} - x_k)^2 + c_{3k}(x_{k+1} - x_k)^3 = y_{k+1}, \\ c_{1k} = y'_k, \\ c_{1k} + 2c_{2k}(x_{k+1} - x_k) + 3c_{3k}(x_{k+1} - x_k)^2 = y'_{k+1}, \end{cases}$$

Interpolarea Hermite

$$\begin{cases} c_{0k} = y_k, \\ c_{1k} = y'_k, \\ c_{2k} = [3(y_{k+1} - y_k) - (x_{k+1} - x_k)(2y'_k + y'_{k+1})] / (x_{k+1} - x_k)^2, \\ c_{3k} = [(y'_{k+1} + y'_k) - 2(2y_{k+1} - y_k) / (x_{k+1} - x_k)] / (x_{k+1} - x_k)^2. \end{cases} \quad (18)$$

Dificultate:

de cele mai multe ori nu se cunosc informații despre y'_k .

Soluție:

Evaluarea derivatelor se poate face numeric (interpolare Bessel):

$$y'_0 = \frac{y_1 - y_0}{x_1 - x_0} \quad y'_n = \frac{y_n - y_{n-1}}{x_n - x_{n-1}},$$

$$y'_k = \frac{\beta^2 y_{k+1} + (\alpha^2 - \beta^2) y_k - \alpha^2 y_{k-1}}{\alpha \beta (\alpha + \beta) h},$$

$x_{k+1} - x_k = \alpha h, x_k - x_{k-1} = \beta h.$ Nenatural (ordinul interpolării la formulele de derivare nr.)

Interpolarea Hermite

Cel mai avantajos: evaluarea derivatelor din impunerea unor condiții de continuitate suplimentare pentru derivata a doua a polinomului de interpolare în nodurile rețelei de interpolare \Rightarrow **Interpolare spline cubică clasică** impune

$$g''(x_k - 0) = g''(x_k + 0), \quad k = 1, \dots, n - 1. \quad (19)$$

Deoarece

$$g''(x) = 2c_{2k} + 6c_{3k}(x - x_k), \quad x \in [x_k, x_{k+1}]. \quad (20)$$

\Rightarrow

$$2c_{2,k-1} + 6c_{3,k-1}(x_k - x_{k-1}) = 2c_{2k}, \quad k = 1, \dots, n - 1. \quad (21)$$

\Rightarrow

Interpolarea Hermite

$$\begin{aligned} & \frac{1}{x_k - x_{k-1}} y'_{k-1} + 2 \left(\frac{1}{x_k - x_{k-1}} + \frac{1}{x_{k+1} - x_k} \right) y'_k + \frac{1}{x_{k+1} - x_k} y'_{k+1} = \\ & = 3 \frac{y_k - y_{k-1}}{(x_k - x_{k-1})^2} + 3 \frac{y_{k+1} - y_k}{(x_{k+1} - x_k)^2}. \end{aligned} \quad (22)$$

$n - 1$ relații și $n + 1$ necunoscute.
Pentru unicitate se impun încă două condiții la capete.

Interpolarea Hermite

Pentru unicitate se impun încă două condiții la capete.

Variante:

- **Condiții forțate la capete:** se impun y'_0 și y'_n ca la interpolarea Bessel
- **Condiții naturale la capete:**

$$g''(x_0) = 0, \quad g''(x_n) = 0.$$

⇒

$$2y'_0 + y'_1 = 3 \frac{y_1 - y_0}{x_1 - x_0}, \quad y'_{n-1} + 2y'_n = 3 \frac{y_n - y_{n-1}}{x_n - x_{n-1}}. \quad (23)$$

Interpolarea Hermite

Sistemul de rezolvat pentru evaluarea derivatelor - **tridiagonal**:

$$2y'_0 + y'_1 = 3 \frac{y_1 - y_0}{x_1 - x_0},$$

$$\begin{aligned} & \frac{1}{x_k - x_{k-1}} y'_{k-1} + 2 \left(\frac{1}{x_k - x_{k-1}} + \frac{1}{x_{k+1} - x_k} \right) y'_k + \frac{1}{x_{k+1} - x_k} y'_{k+1} = \\ & = 3 \frac{y_k - y_{k-1}}{(x_k - x_{k-1})^2} + 3 \frac{y_{k+1} - y_k}{(x_{k+1} - x_k)^2}, \end{aligned}$$

$$y'_{n-1} + 2y'_n = 3 \frac{y_n - y_{n-1}}{x_n - x_{n-1}}.$$

Interpolarea Hermite

Avantajul interpolării spline cubice clasice:

minimizează curbura pătratică medie a polinomului de interpolare

$$\int_a^b (g''(x))^2 dx$$

⇒ nu apar oscilații nedorite între punctele rețelei de interpolare

Interpolarea Hermite

Algoritmul interpolării spline cubice:

funcție pregătește_spline($n, x, y, yder$)

; calculează derivatele funcției în nodurile rețelei de discretizare

; declarații - parametri de intrare

întreg n

; dimensiunea problemei - numărul de intervale

tablou real $x[n], y[n]$

; tabelul de valori, indici de la zero

tablou real $yder[n]$

; parametri de ieșire

tablou real $p[n], q[n], r[n]$

; matricea tridiagonală asamblată

; assemblează matricea tridiagonală

$q_0 = 2$

$r_0 = 1$

$b_0 = 3(y_1 - y_0)/(x_1 - x_0)$

pentru $k = 1, n - 1$

$$p_k = 1/(x_k - x_{k-1})$$

$$q_k = 2/(x_k - x_{k-1}) + 2/(x_{k+1} - x_k)$$

$$r_k = 1/(x_{k+1} - x_k)$$

$$b_k = 3(y_k - y_{k-1})/(x_k - x_{k-1})^2 + 3(y_{k+1} - y_k)/(x_{k+1} - x_k)^2$$

•

$p_n = 1$

$q_n = 2$

$b_n = 3(y_n - y_{n-1})/(x_n - x_{n-1})$

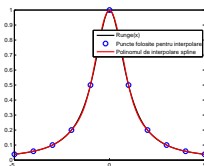
Gauss_tridiag($n + 1, p, q, r, b, yder$)

Interpolarea Hermite

```

funcție aproximare_spline(n, x, y, yder, xcrt)
; evaluează polinomul de aproximare spline în punctul xcrt
; declarații - parametri de intrare
întreg n ; dimensiunea problemei - numărul de intervale
tablou real x[n], y[n], yder[n] tabelul de valori, indici de la zero
real xcrt ; punctul de evaluat
k = cauta(n, x, xcrt)
c0k = y_k
c1k = yder_k
h = x_{k+1} - x_k
c2k = (3(y_{k+1} - y_k) - h(2 * yder_k + yder_{k+1})) / h^2
c3k = (yder_{k+1} + yder_k - 2(y_{k+1} - y_k) / h) / h^2
h crt = xcrt - x_k
întoarce c0k + c1k * h crt + c2k * h crt^2 + c3k * h crt^3

```



Interpolarea *spline* a funcției Runge pe o rețea uniformă de puncte.

Lectură obligatorie

- Pag 162-168 din

[3] Gabriela Ciuprina - Algoritmi numerici pentru calcule științifice în ingineria electrică, Editura MatrixROM, 2013, disponibil la

http://lmn.pub.ro/~gabriela/books/AlgNr_MatrixRom2013.pdf