Interpolarea funcţiilor.

– Metode de interpolare globală (recapitulare). –

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" Bucureşti, Facultatea de Inginerie Electrică

Suport didactic pentru disciplina *Algoritmi Numerici*, 2017-2018

Notes

Cuprins

Notes

Preliminarii

Scrierea formală a unei probleme

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

$\mathbf{x}$ - datele problemei (parametri independenţi);
$\mathbf{y}$ - 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 configuraţia caracterizată complet de datele $\mathbf{x}$ şi să calculeze printr-un algoritm de postprocesare mărimile $\mathbf{y}$.

Notes

Preliminarii

Formularea problemei (neriguros)

Se dă o funcţie reprezentată prin date:
$(\mathbf{x}_k, \mathbf{y}_k)$, $k = 0, \ldots, n$, unde $\mathbf{y}_k = f(\mathbf{x}_k)$.
Se doreşte găsirea unei expresii analitice pentru o funcţie 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 <u>prin</u> punctele mulţimii de date: $g(\mathbf{x}_k) = \mathbf{y}_k$
- Aproximarea (regresia) setului de date = g trece <u>printre</u> punctele mulţimii de date: $g(\mathbf{x}_k) \approx \mathbf{y}_k$

Notes

Preliminarii

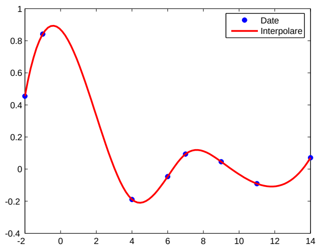
Observaţii:

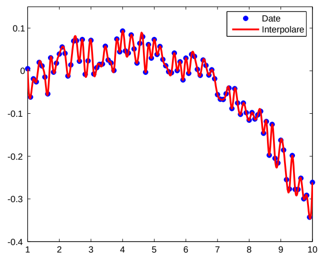
1. $\mathbf{x}_k$ se numeşte şi reţea (grid) de discretizare.
2. Interpolarea/aproximarea este utilă şi dacă funcţia este reprezentată prin cod = există un software capabil să calculeze $f(\mathbf{x})$ pentru orice $\mathbf{x}$ dorit, dacă efortul de evaluare al lui f este mare.

Notes

Preliminarii

Exemple: interpolare



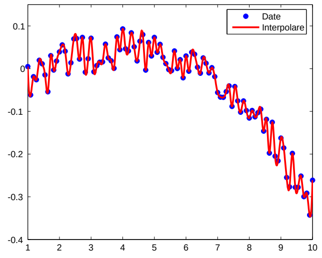


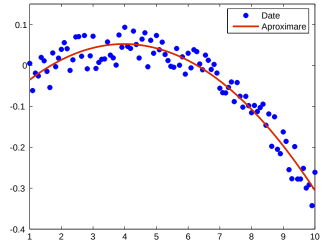
Interpolarea unui set de date. În cazul în care setul de date are foarte multe valori, interpolarea poate genera oscilaţii nedorite.

Notes

Preliminarii

Exemple: interpolare vs. aproximare





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

Notes

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 situaţii de tip nD.

În cele ce urmează vom pp. cazul 1D.

Notes

Distanţa dintre două funcţii

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

$\Leftrightarrow$

distanţa dintre cele două funcţii

$$d(f, g) = \|f - g\| \tag{2}$$

să fie cât mai mică.

Există mai multe procedee de definire a normei.

Notes

Distanţa dintre două funcţii

Procedee de definire a normei.

- Aria dintre graficele celor două funcţii

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

Dezavantaj: local, pot exista diferenţe foarte mari între f sî g.

- Abaterea medie pătratică

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

Acelaşi dezavantaj.

Notes

Distanţa dintre două funcţii

Procedee de definire a normei.

- Abaterea maximă dintre cele două funcţii

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

Din pdv al acurateţii - este cea mai avantajoasă.

OBS: Niciuna din aceste norme nu se poate evalua.

Notes

Distanţa dintre două funcţii

Normele discrete:

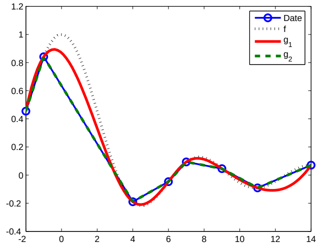
$$d_{1d}(f,g) = \sum_{k=0}^{n} |g(x_k) - f(x_k)|, \quad (6)$$

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

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

Notes

Distanţa dintre două funcţii



Avantaj: pot fi evaluate cu uşurinţă.
Dezavantaj: se pierde posibilitatea evaluării acurateţii între noduri. Mai mult $d_d(f, g_1) = 0$; $d_d(f, g_2) = 0$; $\Rightarrow$ problemă prost formulată.

Notes

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 condiţiile de interpolare

$$g(x_j) = f(x_j), \quad j = 0, \ldots, n, \tag{9}$$

$\Leftrightarrow$

$$g(x_j) = y_j, \quad j = 0, \ldots, n. \tag{10}$$

Pentru a face ca problema să fie bine formulată matematic (soluţia să existe şi să fie unică) funcţia g se caută în spaţiul polinoamelor generalizate
$\Leftrightarrow$
g adică se caută de forma unei combinaţii liniare de m funcţii φ_k, $k = 1, \ldots, m$ numite funcţii de bază:

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

Notes

Formularea problemei interpolării

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

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

Cei m coeficienţi c_k se calculează din impunerea condiţiilor de interpolare:

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

$\Rightarrow$ Sistem algebric liniar cu $n+1$ ecuaţii şi $m+1$ necunoscute.

Notes

Formularea problemei interpolării

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

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

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

Notes

Formularea problemei interpolării

Date:

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

Se cer:

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

Notes

Metode de interpolare globală

- Metodele de interpolare globală = metodele în care funcţiile de bază se definesc compact, printr-o singură expresie pe întreg domeniul de definiţie al funcţiei de interpolat.
- Gradul polinomului de interpolare = numărul de puncte din tabelul de date - 1
- În funcţie de cum se aleg funcţiile de bază se obţin diferite metode de interpolare globală.

Notes

Metoda clasică

Funcţiile de bază: $\varphi_k(x) = x^k$, $k = 0, \dots, n$.
Polinomul de interpolare:

$$g(x) = \sum_{k=0}^{n} c_k x^k = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n, \tag{14}$$

Din impunerea condiţiilor de interpolare ⇒

$$\left\{\begin{array}{l} c_0 + c_1 x_0 + c_2 x_0^2 + \cdots + c_n x_0^n = y_0 \\ c_0 + c_1 x_1 + c_2 x_1^2 + \cdots + c_n x_1^n = y_1 \\ \qquad\qquad \cdots \\ c_0 + c_1 x_n + c_2 x_n^2 + \cdots + c_n x_n^n = y_n \end{array}\right. \tag{15}$$

- etapa de pregătire = calculul coeficienţilor polinomului de interpolare
- etapa de evaluare = evaluarea propriu-zisă a polinomului de interpolare.

Notes

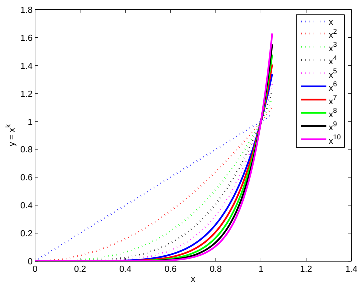
Introducere
Metode de interpolare globală
Interpolarea Chebyshev
Metoda clasică
Metoda Lagrange
Metoda Newton

Metoda clasică

Efort de calcul:

- Etapa de pregătire: $O(2n^3/3)$ - dezavantaj
- Etapa de evaluare: $O(2n)$.

Dezavantaj major: pentru valori mari ale lui n matricea coeficienţilor sistemului este slab condiţionată



Notes

Metoda Lagrange

Funcţiile de bază sunt polinoamele Lagrange

$$\varphi_k(x) = l_k(x) = \frac{\prod_{i=0, i\neq k}^{n}(x - x_i)}{\prod_{i=0, i\neq k}^{n}(x_k - x_i)}, \tag{16}$$

Polinomul de interpolare este

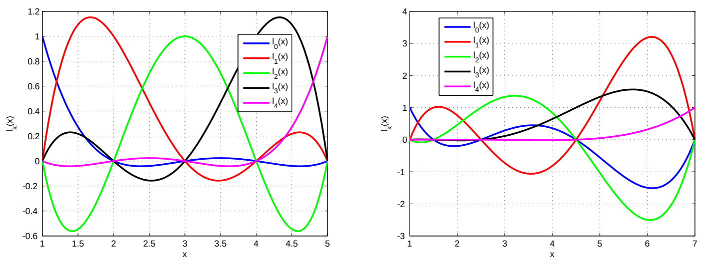
$$g(x) = \sum_{k=0}^{m} c_k l_k(x). \tag{17}$$

$$l_k(x_j) = \begin{cases} 1 & \text{dacă} \quad j = k, \\ 0 & \text{dacă} \quad j \neq k. \end{cases} \tag{18}$$

Notes

Metoda Lagrange

Polinoame Lagrange



Funcţiile Lagrange pentru o reţea de discretizare uniformă (stânga), respectiv neuniformă (dreapta).

Notes

Metoda Lagrange

Condiţiile de interpolare $g(x_j) = y_j$, $j = 0, \dots, n \Rightarrow$

$$\sum_{k=0}^{m} c_k l_k(x_j) = y_j, \tag{19}$$

$\Rightarrow$

$$c_j = y_j, \quad j = 0, \dots, n. \tag{20}$$

Expresia polinomului Lagrange:

$$g(x) = \sum_{k=0}^{n} y_k l_k(x) = \sum_{k=0}^{n} y_k \frac{\prod_{i=0, i \neq k}^{n} (x - x_i)}{\prod_{i=0, i \neq k}^{n} (x_k - x_i)}. \tag{21}$$

Notes

Metoda Lagrange

Exemplu: dreapta ce trece prin 2 puncte

$$g(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}, \tag{22}$$

Exemplu: parabola ce trece prin 3 puncte

$$g(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \tag{23}$$

Implementarea formulelor de acest tip (fără pregătire)
$T_e = \mathrm{O}(4n^2)$ pentru fiecare evaluare.
Efort de calcul total pentru evaluarea în m puncte de
$T_{L-fp} = \mathrm{O}(4mn^2)$.

Notes

Metoda Lagrange - implementare

Se scoate factor comun forţat

$$p = \prod_{k=0}^{n} (x - x_k), \tag{24}$$

polinomul de interpolare fiind

$$g(x) = p \sum_{k=0}^{n} \frac{\alpha_k}{x - x_k}, \tag{25}$$

unde coeficienţii notaţi

$$\alpha_k = \frac{y_k}{\prod_{j=0, j \neq k}^{n} (x_k - x_j)} \tag{26}$$

vor fi calculaţi în etapa de pregătire.

Notes

Metoda Lagrange - implementare

```
procedură Lagrange_pregătire(n, x, y, α)
; pregăteşte coeficienţii din metoda Lagrange
întreg n                    ; dimensiunea problemei - numărul de intervale
tablou real x[n], y[n]      ; tabelul de valori, indici de la zero
; declaraţii - parametri de ieşire
tablou real α[n]            ; coeficienţii
pentru k = 0, n
        α_k = y_k
        pentru j = 0, n
                dacă j ≠ k atunci α_k = α_k/(x_k − x_j)
        •
•
retur
```


Notes

Metoda Lagrange - implementare

```
funcţie Lagrange_evaluare(n, x, y, α, xcrt)
; evaluează polinomul de interpolare Lagrange în punctul xcrt
întreg n                ; dimensiunea problemei - numărul de intervale
tablou real x[n], y[n]  ; tabelul de valori, indici de la zero
tablou real α[n]        ; coeficienţii
real xcrt               ; punctul de evaluat
; alte declaraţii
real p, s
p = 1
pentru k = 0, n
        dacă |xcrt − x_k| < zeroul_maşinii() atunci întoarce y_k
        p = p * (xcrt − x_k)
•
s = 0
pentru k = 0, n
        s = s + α_k/(xcrt − x_k)
•
întoarce s * p
```


Notes

Metoda Lagrange - efort de calcul

Varianta cu pregătire

- Etapa de pregătire: $T_p = \mathrm{O}(2n^2)$
- Etapa de evaluare $T_e = O(5n)$
- Efort de calcul total $T_{L-cp} = \mathrm{O}(2n^2 + 5mn)$.

Varianta fără pregătire

- Etapa de evaluare $T_e = \mathrm{O}(4n^2)$
- Efort de calcul total $T_{L-fp} = \mathrm{O}(4mn^2)$.

Obs: algoritmul trateaza şi cazul în care punctul de evaluare coincide cu unul din punctele din tabel.

Notes

Eroarea de trunchiere

Dacă f este continuă şi derivabilă de un număr finit de ori, se poate obţine o margine a erorii de interpolare

$$e(x) = f(x) - g(x), \tag{27}$$

care poate fi privită ca o eroare de trunchiere deoarece interpolarea caută o aproximare într-un spaţiu finit dimensional de funcţii.

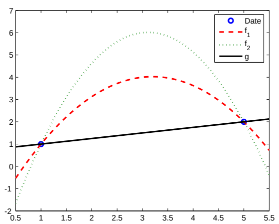
$$|e(x)| \leq \frac{M_{n+1}}{(n+1)!} \prod_{k=0}^{n} |x - x_k|. \tag{28}$$

Eroarea de interpolare depinde de marginea derivatei de un ordin egal cu numărul de puncte de tabel.

Notes

Eroarea de trunchiere

$$|e(x)| \leq \frac{M_{n+1}}{(n+1)!} \prod_{k=0}^{n} |x - x_k|. \tag{29}$$



În cazul unui tabel de valori cu două puncte, presupunând că funcţia adevărată este o parabolă, atunci eroarea de interpolare este mai mică pentru parabola f_1, care are derivata de ordinul doi (curbura) mai mică.

Notes

Metoda Newton

Funcţii de bază:

$$\begin{aligned} &\varphi_0(x) = 1, \\ &\varphi_1(x) = (x - x_0), \\ &\varphi_2(x) = (x - x_0)(x - x_1), \\ &\dots \\ &\varphi_n(x) = (x - x_0)(x - x_1)\cdots(x - x_{n-1}). \end{aligned} \tag{30}$$

Polinomul de interpolare:

$$g(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \cdots + c_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}), \tag{31}$$

Condiţiile de interpolare:

$$\begin{cases} c_0 = y_0, \\ c_0 + c_1(x_1 - x_0) = y_1, \\ c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) = y_2, \\ \dots \\ c_0 + c_1(x_n - x_0) + c_2(x_n - x_0)(x_n - x_1) + \cdots c_n(x_n - x_0)(x_n - x_1)\cdots(x_n - x_{n-1}) = y_n. \end{cases} \tag{32}$$

Metoda Newton

Coeficienţii reprezintă diferenţe divizate ale funcţiei faţă de submulţimi ale nodurilor de discretizare:

$$\begin{cases} c_0 = y_0 = f[x_0] \\ c_1 = (y_1 - y_0)/(x_1 - x_0) = f[x_0, x_1] \\ c_2 = [y_2 - y_0 - (y_1 - y_0)/(x_1 - x_0)(x_2 - x_0)]/(x_2 - x_0)/(x_2 - x_1) = f[x_0, x_1, x_2], \\ \vdots \\ c_n = \cdots = f[x_0, x_1, x_2, \dots, x_n]. \end{cases} \tag{33}$$

Diferenţa divizată faţa de o submulţime a nodurilor reţelei de discretizare se defineşte în mod recursiv:

$$f[x_i, x_{i+1}, \dots, x_{i+m}] = \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+m}] - f[x_i, x_{i+1}, \dots, x_{i+m-1}]}{x_{i+m} - x_i}. \tag{34}$$

Notes

Notes

Metoda Newton

Proprietăţi ale diferenţelor divizate:

- Variabilele independente pot fi permutate într-o diferenţă divizată fără ca valoarea funcţiei să se modifice.

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{f[x_0] - f[x_1]}{x_0 - x_1} = f[x_1, x_0] \qquad (35)$$

şi, pe baza relaţiei de recurenţă această proprietate poate fi generalizată pentru o mulţime oarecare de noduri.

Notes

Metoda Newton

Proprietăţi ale diferenţelor divizate:

- Diferenţa divizată faţă de o submulţime cu două puncte identice este derivata funcţiei:

$$f[x_0, x_0] = \lim_{x_1 \to x_0} \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f'(x_0). \qquad (36)$$

Legătura dintre diferenţele divizate ale unei funcţii şi derivatele sale poate fi generalizată.

Notes

Metoda Newton

Polinomul de interpolare Newton:

$$\begin{aligned} g(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ &+ \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}). \end{aligned} \tag{37}$$

Dacă $x_k \to x_0$, atunci $g(x)$ tinde către seria Taylor a funcţiei f în x_0:

$$g(x) \to f[x_0]+f[x_0, x_0](x-x_0)+f[x_0, x_0, x_0](x-x_0)^2+\cdots f[\underbrace{x_0, x_0, \ldots, x_0}_{n+1}](x-x_0)^n. \tag{38}$$

Notes

Metoda Newton

Avantaje:

- termeni succesivi ai polinomului interpolării Newton aproximează termeni corespunzători din dezvoltarea în serie Taylor $\Rightarrow$ se poate controla eroarea de trunchiere, efortul de calcul putând fi adaptat preciziei impuse soluţiei.
- atunci când se adaugă un punct suplimentar în reţeaua de interpolare, se poate porni de la interpolarea cu un grad mai scăzut şi trebuie doar adăugat un singur termen în sumă, nefiind necesară refacerea totală a calculelor, ci doar evaluarea unui singur termen suplimentar.

Notes

Metoda Newton

Fie $g(x)$ = interpolarea funcţiei $f(x)$ pe x_0, x_1, ..., x_n
Dacă se adaugă un nod nou x_{n+1}, noul polinom de interpolare:

$$h(x) = g(x) + f[x_0, x_1, \ldots, x_n, x_{n+1}](x - x_0)(x - x_1)\cdots(x - x_n). \quad (39)$$

Condiţia de interpolare suplimentară: $h(x_{n+1}) = f(x_{n+1}) \Rightarrow$

$$f(x_{n+1}) = g(x_{n+1}) + f[x_0, x_1, \ldots, x_n, x_{n+1}](x_{n+1} - x_0)(x_{n+1} - x_1)\cdots(x_{n+1} - x_n). \quad (40)$$

Deoarece x_{n+1} poate fi ales arbitrar, putem înlocui x_{n+1} cu x, deci

$$f(x) = g(x) + f[x_0, x_1, \ldots, x_n, x](x - x_0)(x - x_1)\cdots(x - x_n), \quad (41)$$

$\Rightarrow$ eroarea de trunchiere făcută la interpolarea funcţiei f pe reţeaua de noduri x_0, x_1, ..., x_n este

$$e(x) = f(x) - g(x) = f[x_0, x_1, \ldots, x_n, x] \prod_{k=0}^{n} (x - x_k). \quad (42)$$

Notes

Metoda Newton

$$e(x) = f(x) - g(x) = f[x_0, x_1, \ldots, x_n, x] \prod_{k=0}^{n} (x - x_k). \quad (43)$$

Dar

$$e(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (x - x_k). \quad (44)$$

$\Rightarrow$ există ξ a.î. $f[x_0, x_1, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}$.
Diferenţele divizate pe o reţea cu mai mult de două noduri reprezintă aproximări ale derivatelor de ordin superior, ordinul derivatei fiind cu unu mai mic decât numărul de noduri ale diviziunii.
În particular, dacă toate nodurile tind către $x_0 \Rightarrow$
$f[\underbrace{x_0, x_0, \ldots, x_0, x_0}_{n+2}] = \frac{f^{(n+1)}(x_0)}{(n+1)!}$.

Notes

Metoda Newton - algoritm

Etapa de pregătire: se calculează tabelul diferenţelor divizate:

x_0		x_1		x_2	$\cdots$	x_{n-1}		x_n
$y_0 = f[x_0]$		$y_1 = f[x_1]$		$y_2 = f[x_2]$	$\cdots$	$y_{n-1} = f[x_{n-1}]$		$y_n = f[x_n]$
	$f[x_0, x_1]$		$f[x_1, x_2]$		$\cdots$		$f[x_{n-1}, x_n]$	
		$f[x_0, x_1, x_2]$			$\cdots$	$f[x_{n-2}, x_{n-1}, x_n]$		
			$\vdots$					
				$\vdots$				
				$f[x_0, x_1, \ldots, x_n]$				

Sunt calculate $n(n+1)/2$ diferenţe divizate, din care utile pentru polinomul de interpolare sunt doar n.

Notes

Metoda Newton - algoritm

```
procedură Newton_pregătire(n, x, y, dd)
; pregăteşte diferenţele divizate din metoda Newton
întreg n                    ; dimensiunea problemei - nr. de intervale
tablou real x[n], y[n]      ; tabelul de valori, indici de la zero
; declaraţii - parametri de ieşire
tablou real dd[n][n]        ; diferenţele divizate
pentru j = 0, n
      dd(0, j) = y(j)
•
pentru i = 1, n
      pentru j = 0, n − i
            dd(i, j) = (dd(i − 1, j) − dd(i − 1, j − 1))/(x(j + i) − x(j))
      •
•
retur
```


Notes

Metoda Newton - algoritm

```
funcţie Newton_evaluare(n, x, y, dd, xcrt)
; evaluează polinomul de interpolare Newton în punctul xcrt
întreg n                    ; dimensiunea problemei - nr. de intervale
tablou real x[n], y[n]      ; tabelul de valori, indici de la zero
tablou real dd[n][n]        ; diferenţele divizate
real xcrt                   ; punctul de evaluat
real ycrt                   ; valoarea funcţiei în punctul de evaluat
ycrt = dd(n, 0)
pentru k = n − 1, 0, −1
        ycrt = dd(k, 0) + (xcrt − x_k) ∗ ycrt
•
întoarce ycrt
```


Notes

Metoda Newton - algoritm

Funcţia de evaluare se bazează pe scrierea polinomului sub forma

$$g(x) = c_0 + (x - x_0)\,[c_1 + (x - x_1)\,[c_2 + \cdots [c_{n-1} + c_n(x - x_n)] \cdots]] \,. \tag{45}$$

Efort de calcul:

- Etapa de pregătire: $\sum_{i=1}^{n} 2(n - i) = 2n(n - 1)/2$ operaţii $\Rightarrow$ $T_p = O(n^2)$.
- Etapa de evaluare: $T_e = O(3n)$
- Efort de calcul total în metoda Newton $T_N = O(n^2 + 3mn)$.

Efortul de calcul pentru metodele de interpolare globală.

Metoda	Pregătire	Evaluare
Clasică	$O(2n^3/3)$	$O(2mn)$
Lagrange	$O(2n^2)$	$O(5mn)$
Newton	$O(n^2)$	$O(3mn)$

Notes

Metoda Newton - algoritm

Concluzii:

1. Metodele clasică, Lagrange, Newton dau teoretic acelaşi rezultat pentru că polinomul de interpolare est unic.
2. Metoda Newton este cea mai eficientă din punct de vedere al timpului de pregătire, cât şi a celui de evaluare.
3. Avantajul major este însă acela că metoda Newton permite controlul erorii de trunchiere.

Evaluarea polinomului de interpolare cu controlul erorii de trunchiere:

Notes

Metoda Newton - algoritm

```
procedură Newton_evaluare2(n, x, y, dd, xcrt, ycrt, et)
; evaluează polinomul de interpolare Newton în punctul xcrt
; declaraţii - parametri de intrare
întreg n                      ; dimensiunea problemei - nr. de intervale
tablou real x[n], y[n]        ; tabelul de valori, indici de la zero
tablou real dd[n][n]          ; diferenţele divizate
real xcrt                     ; punctul de evaluat
real ycrt                     ; valoarea funcţiei în punctul de evaluat
real et                       ; eroarea de trunchiere
ycrt = dd(n − 1, 0)
pentru k = n − 2, 0, −1
        ycrt = dd(k, 0) + (xcrt − x_k) ∗ ycrt
•
et = dd(n, 0)
pentru k = 0, n − 1
        et = et ∗ (xcrt − x_k)
•
retur
```


Notes

Un exemplu simplu

x	-1	2	4
y	-6	9	49

Metoda clasică:

$$g(x) = c_0 + c_1 x + c_2 x^2, \tag{46}$$

Condiţiile de interpolare

$$\begin{aligned} g(-1) &= 6, \\ g(2) &= 9, \\ g(4) &= 49, \end{aligned} \tag{47}$$

$\Rightarrow$

$$\begin{array}{ccccccc} c_0 & - & c_1 & + & c_2 & = & 6, \\ c_0 & + & 2c_1 & + & 4c_2 & = & 9, \\ c_0 & + & 4c_1 & + & 16c_2 & = & 49. \end{array} \tag{48}$$

Soluţia acestui sistem este $c_0 = -7$, $c_1 = 2$, $c_2 = 3$, de unde $g(x) = 3x^2 + 2x - 7$.

Notes

Un exemplu simplu

x	-1	2	4
y	-6	9	49

Metoda Lagrange:

$$l_0(x) = \frac{(x-2)(x-4)}{(-1-2)(-1-4)} = \frac{x^2-6x+8}{15}, \tag{49}$$

$$l_1(x) = \frac{(x+1)(x-4)}{(2+1)(2-4)} = \frac{x^2-3x-4}{-6}, \tag{50}$$

$$l_2(x) = \frac{(x+1)(x-2)}{(4+1)(4-2)} = \frac{x^2-x-2}{10}, \tag{51}$$

iar polinomul de interpolare globală este

$$g(x) = -6l_0(x) + 9l_1(x) + 49l_2(x) = 3x^2 + 2x - 7. \tag{52}$$

Notes

Un exemplu simplu

x	-1	2	4
y	-6	9	49

Metoda Newton:

-1		2		4
$f[x_0] = -6$		$f[x_1] = 9$		$f[x_2] = 49$
	$f[x_0, x_1] = 5$		$f[x_1, x_2] = 20$	
		$f[x_0, x_1, x_2] = 3$		

$f[x_0, x_1] = (9+6)/(2+1) = 5$,
$f[x_1, x_2] = (49-9)/(4-2)$,$f[x_0, x_1, x_2] = (20-5)/(4+1)$
Polinomul de interpolare:

$$g(x) = -6 + 5(x+1) + 3(x+1)(x-2) = 3x^2 + 2x - 7. \quad (53)$$

Notes

Un exemplu simplu

Dacă adăugăm un nou punct în acest tabel

x	-1	2	4	3
y	-6	9	49	10

putem calcula uşor polinomul de gradul trei.

-1		2		4		3
-6		9		49		10
	5		20		39	
		3		19		
			4			

iar polinomul de interpolare este

$$h(x) = 3x^2 + 2x - 7 + 4(x+1)(x-2)(x-4) = 4x^3 - 17x^2 + 10x + 25.$$

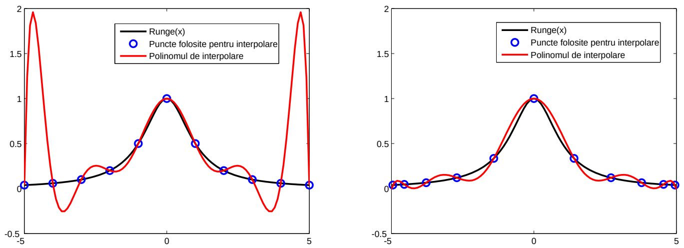
Termenul $e(x) = 4(x+1)(x-2)(x-4)$ reprezintă eroarea de trunchiere a polinomului de interpolare de grad doi $g(x) = 3x^2 + 2x - 7$ pentru tabelul de valori ce conţine patru puncte.

Notes

Interpolarea Chebyshev

Runge: $f : [-5, 5] \to \mathbb{R}$, $f(x) = 1/(1 + x^2)$, interpolarea pe o reţea echidistantă de puncte duce la oscilaţii la capetele polinomului de interpolare, cu atât mai mari cu cât gradul polinomului este mai mare .

Efectul de oscilaţie a polinomului de interpolare între nodurile reţelei de interpolare se numeşte efect Runge.



Efectul Runge: polinomul de intepolare are oscilaţii la capetele intervalului dacă gridul de discretizare este uniform (stânga). Efectul poate fi eliminat prin plasarea nodurilor de discretizare în conformitate cu rădăcinile polinoamelor

Notes

Interpolarea Chebyshev

Efectul lui Runge

- se poate explica prin faptul că eroarea de trunchiere dată de poate fi oricât de mare datorită faptului că $f^{(n+1)}(\xi)$ poate fi oricât de mare;
- se poate elimina dacă nodurile reţelei de interpolare se îndesesc către capetele domeniului.

Oscilaţii minime se obţin dacă plasarea nodurilor în interiorul domeniului de definiţie se face în concordanţă cu rădăcinile polinoamelor Chebyshev. În intervalul $[-1, 1]$, acestea sunt

$$x_i = \cos\left(\frac{2i+1}{2(n+1)}\pi\right), \quad i = 0, \ldots, n, \tag{54}$$

Într-un interval arbitrar $[a, b]$:

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{2i+1}{2(n+1)}\pi\right). \tag{55}$$

Notes

Interpolarea Chebyshev

Interpolarea Chebyshev este tot interpolarea polinomială globală numai că nodurile reţelei de interpolare sunt alese în concordanţă cu rădăcinile polinoamelor Chebyshev.

1. Limitarea acestei metode este aceea că se poate aplica numai funcţiilor definite prin cod şi nu tabelar.
2. În cazul funcţiilor date tabelar, pentru a obţine polinoame de interpolare care să nu prezinte efect Runge, este mai eficient dacă se foloseşte interpolarea pe porţiuni.

Notes

Referinţe

- [Ciuprina13a] Gabriela Ciuprina - Algoritmi numerici pentru calcule ştiinţifice în ingineria electrică , Editura MatrixROM, 2013, pag. 143-162.

  disponibilă la http://www.lmn.pub.ro/∼gabriela/books/AlgNr_MatrixRom2013.pdf
- [Buchanan92] James Buchanan, Peter Turner, *Numerical methods and analysis*, McGraw Hill, 1992.

Notes