

Interpolarea funcțiilor.

– Metode de interpolare globală. –

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 globală
 - Metoda clasică
 - Metoda Lagrange
 - Metoda Newton
- 3 Interpolarea Chebyshev

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: $g(\mathbf{x}_k) \approx \mathbf{y}_k$

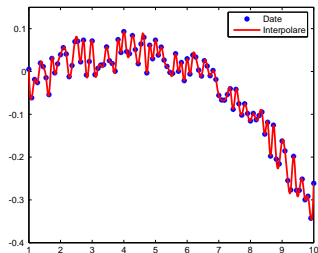
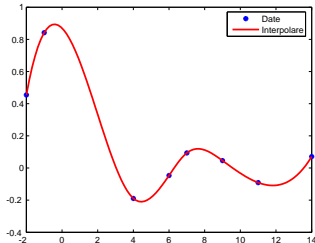
Preliminarii

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

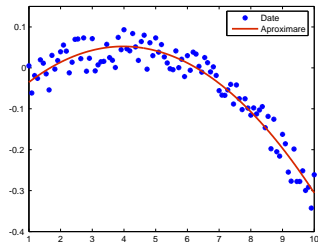
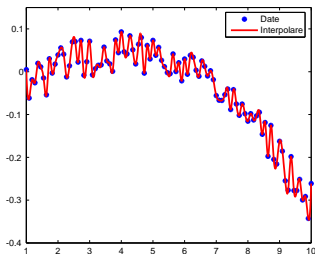
Exemple: interpolare



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

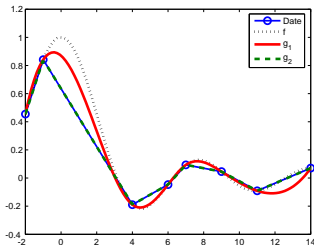
Normele discrete:

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

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

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

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:

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

Metode de interpolare globală

- **Metodele de interpolare globală** = metodele în care funcțiile de bază se definesc compact, printr-o singură expresie pe întreg domeniul de definiție al funcției de interpolat.
- Gradul polinomului de interpolare = numărul de puncte din tabelul de date - 1
- În funcție de cum se aleg funcțiile de bază se obțin diferite metode de interpolare globală.

Metoda clasică

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

Polinomul de interpolare:

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

Din impunerea condițiilor de interpolare \Rightarrow

$$\begin{cases} c_0 + c_1 x_0 + c_2 x_0^2 + \dots + c_n x_0^n = y_0 \\ c_0 + c_1 x_1 + c_2 x_1^2 + \dots + c_n x_1^n = y_1 \\ \dots \\ c_0 + c_1 x_n + c_2 x_n^2 + \dots + c_n x_n^n = y_n \end{cases} \quad (15)$$

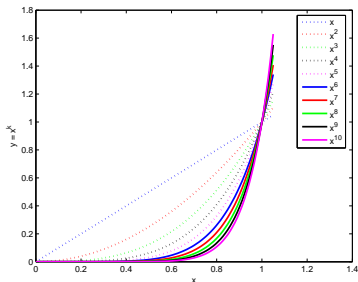
- **etapa de pregătire** = calculul coeficienților polinomului de interpolare
- **etapa de evaluare** = evaluarea propriu-zisă a polinomului de interpolare.

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 coeficienților sistemului este slab condiționată



Metoda Lagrange

Funcțiile de bază sunt **polinoamele Lagrange**

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

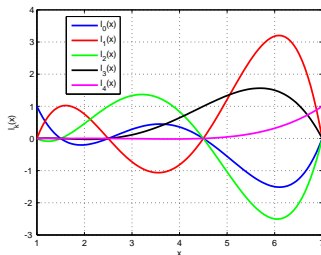
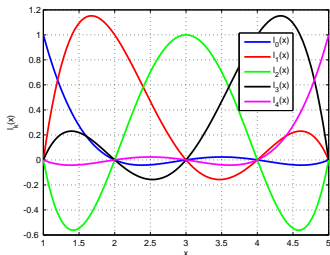
Polinomul de interpolare este

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

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

Metoda Lagrange

Polinoame Lagrange



Funcțiile Lagrange pentru o rețea de discretizare uniformă (stânga), respectiv neuniformă (dreapta).

Metoda Lagrange

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

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

\Rightarrow

$$c_j = y_j, \quad j = 0, \dots, n. \quad (20)$$

Expresia polinomului Lagrange:

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

Metoda Lagrange

Exemplu: dreapta ce trece prin 2 puncte

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

Exemplu: parabola ce trece prin 3 puncte

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

Implementarea formulelor de acest tip (fără pregătire)

$T_e = O(4n^2)$ pentru fiecare evaluare.

Efort de calcul total pentru evaluarea în m puncte de

$T_{L-fp} = O(4mn^2)$.

Metoda Lagrange - implementare

Se scoate factor comun forțat

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

polinomul de interpolare fiind

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

unde coeficienții notați

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

vor fi calculați în etapa de pregătire.

Metoda Lagrange - implementare

```
procedură Lagrange_pregătire( $n, x, y, \alpha$ )  
; pregătește coeficienții din metoda Lagrange  
întreg  $n$  ; dimensiunea problemei - numărul de intervale  
tablou real  $x[n], y[n]$  ; tabelul de valori, indici de la zero  
; declarații - parametri de ieșire  
tablou real  $\alpha[n]$  ; coeficienții  
pentru  $k = 0, n$   
     $\alpha_k = y_k$   
    pentru  $j = 0, n$   
        dacă  $j \neq k$  atunci  $\alpha_k = \alpha_k / (x_k - x_j)$   
    •  
•  
retur
```

Metoda Lagrange - implementare

```
funcție Lagrange_evaluare( $n, x, y, \alpha, xcrt$ )  
; evaluează polinomul de interpolare Lagrange în punctul  $xcrt$   
intreg  $n$  ; dimensiunea problemei - numărul de intervale  
tablou real  $x[n], y[n]$  ; tabelul de valori, indici de la zero  
tablou real  $\alpha[n]$  ; coeficienții  
real  $xcrt$  ; punctul de evaluat  
; alte declarații  
real  $p, s$   
 $p = 1$   
pentru  $k = 0, n$   
    dacă  $|xcrt - x_k| < \text{zeroul\_mașinii}()$  atunci întoarce  $y_k$   
         $p = p * (xcrt - x_k)$   
•  
 $s = 0$   
pentru  $k = 0, n$   
     $s = s + \alpha_k / (xcrt - x_k)$   
•  
întoarce  $s * p$ 
```

Metoda Lagrange - efort de calcul

Varianta cu pregătire

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

Varianta fără pregătire

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

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

Eroarea de trunchiere

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

$$e(x) = f(x) - g(x), \quad (27)$$

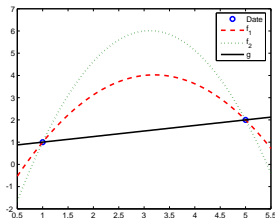
care poate fi privită ca o eroare de trunchiere deoarece interpolarea caută o aproximare într-un spațiu finit dimensional de funcții.

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

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

Eroarea de trunchiere

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



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

Metoda Newton

Funcții 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:

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

Condițiile 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

Coeficienții reprezintă diferențe divizate ale funcției față de submulțimi ale nodurilor de discretizare:

$$\left\{ \begin{array}{l} c_0 = y_0 = f[x_0] \\ c_1 = (y_1 - y_0)/(x_1 - x_0) = f[x_0, x_1] \\ c_2 = [y_2 - y_0 - (y_1 - y_0)/(x_1 - x_0)(x_2 - x_0)]/(x_2 - x_0)/(x_2 - x_1) = \\ \vdots \\ c_n = \dots = f[x_0, x_1, x_2, \dots, x_n]. \end{array} \right. \quad (33)$$

Diferența divizată față de o submulțime a nodurilor rețelei de discretizare se definește în mod recursiv:

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

Metoda Newton

Proprietăți ale diferențelor divizate:

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

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

și, pe baza relației de recurență această proprietate poate fi generalizată pentru o mulțime oarecare de noduri.

Metoda Newton

Proprietăți ale diferențelor divizate:

- Diferența divizată față de o submulțime cu două puncte identice este derivata funcției:

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

Legătura dintre diferențele divizate ale unei funcții și derivatele sale poate fi generalizată.

Metoda Newton

Polinomul de interpolare Newton:

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

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

$$g(x) \rightarrow f[x_0] + f[x_0, x_0](x - x_0) + f[x_0, x_0, x_0](x - x_0)^2 + \dots + \underbrace{f[x_0, x_0, \dots, x_0]}_{n+1}(x - x_0)^{n+1} \quad (38)$$

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 soluției.
- atunci când se adaugă un punct suplimentar în rețeaua de interpolare, se poate porni de la interpolarea cu un grad mai scăzut și trebuie doar adăugat un singur termen în sumă, nefiind necesară refacerea totală a calculelor, ci doar evaluarea unui singur termen suplimentar.

Metoda Newton

Fie $g(x)$ = interpolarea funcției $f(x)$ pe x_0, x_1, \dots, x_n

Dacă se adaugă un nod nou x_{n+1} , noul polinom de interpolare:

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

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

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

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

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

\Rightarrow eroarea de trunchiere făcută la interpolarea funcției f pe rețeaua de noduri x_0, x_1, \dots, x_n este

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

Metoda Newton

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

Dar

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

\Rightarrow există ξ a.î. $f[x_0, x_1, \dots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}$.

Diferențele divizate pe o rețea 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 $x_0 \Rightarrow$

$$f[\underbrace{x_0, x_0, \dots, x_0, x_0}_{n+2}] = \frac{f^{(n+1)}(x_0)}{(n+1)!}.$$

Metoda Newton - algoritm

Etapa de pregătire: se calculează tabelul diferențelor divizate:

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

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

Metoda Newton - algoritm

```
procedură Newton_pregătire( $n, x, y, dd$ )  
; pregătește diferențele divizate din metoda Newton  
întreg  $n$  ; dimensiunea problemei - nr. de intervale  
tablou real  $x[n], y[n]$  ; tabelul de valori, indici de la zero  
; declarații - parametri de ieșire  
tablou real  $dd[n][n]$  ; diferențele 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
```


Metoda Newton - algoritm

```
funcție 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  $dd[n][n]$  ; diferențele divizate  
real  $xcrt$  ; punctul de evaluat  
real  $ycrt$  ; valoarea funcției în punctul de evaluat  
 $ycrt = dd(n, 0)$   
pentru  $k = n - 1, 0, -1$   
     $ycrt = dd(k, 0) + (xcrt - x_k) * ycrt$   
•  
întoarce  $ycrt$ 
```

Metoda Newton - algoritm

Funcția de evaluare se bazează pe scrierea polinomului sub forma

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

Efort de calcul:

- Etapa de pregătire: $\sum_{i=1}^n 2(n - i) = 2n(n - 1)/2$ operații $\Rightarrow T_p = O(n^2)$.
- 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	$O(n^2)$	$O(3mn)$

Metoda Newton - algoritm

Concluzii:

- 1 Metodele clasică, Lagrange, Newton dau teoretic același rezultat pentru că polinomul de interpolare este unic.
- 2 Metoda Newton este cea mai eficientă din punct de vedere al timpului de pregătire, cât și 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:

Metoda Newton - algoritm

```
procedură Newton_evaluare2( $n, x, y, dd, xcrt, ycrt, et$ )  
; evaluează polinomul de interpolare Newton în punctul  $xcrt$   
; declarații - parametri de intrare  
întreg  $n$  ; dimensiunea problemei - nr. de intervale  
tablou real  $x[n], y[n]$  tabelul de valori, indici de la zero  
tablou real  $dd[n][n]$  ; diferențele divizate  
real  $xcrt$  ; punctul de evaluat  
real  $ycrt$  ; valoarea funcției 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
```

Un exemplu simplu

x	-1	2	4
y	-6	9	49

Metoda clasică:

$$g(x) = c_0 + c_1x + c_2x^2, \quad (46)$$

Condițiile de interpolare

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

\Rightarrow

$$\begin{aligned} c_0 - c_1 + c_2 &= 6, \\ c_0 + 2c_1 + 4c_2 &= 9, \\ c_0 + 4c_1 + 16c_2 &= 49. \end{aligned} \quad (48)$$

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

Un exemplu simplu

x	-1	2	4
y	-6	9	49

Metoda Lagrange:

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

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

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

iar polinomul de interpolare globală este

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

Un exemplu simplu

x	-1	2	4
y	-6	9	49

Metoda Newton:

$$\begin{array}{r}
 \begin{array}{ccc}
 -1 & & 2 & & & & 4 \\
 f[x_0] = -6 & & f[x_1] = 9 & & & & f[x_2] = 49 \\
 \hline
 & f[x_0, x_1] = 5 & & & f[x_1, x_2] = 20 & & \\
 & & f[x_0, x_1, x_2] = 3 & & & &
 \end{array}
 \end{array}$$

$$f[x_0, x_1] = (9 + 6)/(2 + 1) = 5,$$

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

Polinomul de interpolare:

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

Un exemplu simplu

Dacă adăugăm un nou punct în acest tabel

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

putem calcula ușor polinomul de gradul trei.

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

iar polinomul de interpolare este

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

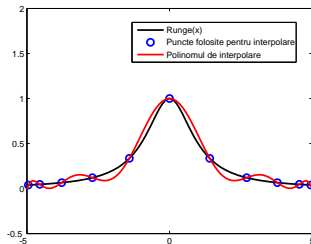
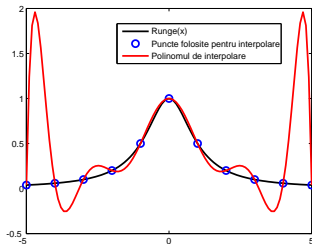
Termenul $e(x) = 4(x + 1)(x - 2)(x - 4)$ reprezintă eroarea de trunchiere a polinomului de interpolare de grad doi

$g(x) = 3x^2 + 2x - 7$ pentru tabelul de valori ce conține patru puncte.

Interpolarea Chebyshev

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

Efectul de oscilație a polinomului de interpolare între nodurile rețelei de interpolare se numește efect Runge.



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

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 rețelei de interpolare se îndesesc către capetele domeniului.

Oscilații minime se obțin dacă plasarea nodurilor în interiorul domeniului de definiție se face în concordanță cu rădăcinile polinoamelor Chebyshev. În intervalul $[-1, 1]$, acestea sunt

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

Într-un interval arbitrar $[a, b]$:

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

Interpolarea Chebyshev

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

- 1 Limitarea acestei metode este aceea că se poate aplica numai funcțiilor definite prin cod și nu tabelar.
- 2 În cazul funcțiilor date tabelar, pentru a obține polinoame de interpolare care să nu prezinte efect Runge, este mai eficient dacă se folosește **interpolarea pe porțiuni**.

Lectură obligatorie

- Cap.6 din

[1] Gabriela Ciuprina, Mihai Rebican, Daniel Ioan - Metode numerice in ingineria electrica - Indrumar de laborator pentru studentii facultatii de Inginerie electrica, Editura Printech, 2013, disponibil la

http://mn.lmn.pub.ro/indrumar/IndrumarMN_Printech2013.pdf