

Interpolare Lagrange

Radu Trîmbițaș

1 aprilie 2025

1 Forma clasică

Fie $f : [a, b] \rightarrow \mathbb{R}$, $x_i \in [a, b]$, $i = 0, \dots, m$. Dacă $x_i \neq x_j$, pentru $i \neq j$, atunci există un polinom unic de gradul m (numit polinomul de interpolare Lagrange), astfel încât:

$$(L_m f)(x_i) = f(x_i), i = 0, \dots, m.$$

Formula de interpolare Lagrange este

$$f = L_m f + R_m f,$$

unde L_m este polinomul de interpolare Lagrange:

$$(L_m f)(x) = \sum_{k=0}^m \ell_k(x) f(x_k), \quad (1)$$

ℓ_k sunt polinoamele fundamentale de interpolare Lagrange

$$\ell_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^m \frac{x - x_j}{x_k - x_j}, \quad (2)$$

iar R_m este termenul rest. Dacă $f \in C^{(m)}[\alpha, \beta]$ și există $f^{(m+1)}$ pe $[\alpha, \beta]$, atunci $\exists \xi \in (\alpha, \beta)$ a.î.

$$(R_m f)(x) = \frac{(x - x_0) \dots (x - x_m)}{(m+1)!} f^{(m+1)}(\xi), \quad (3)$$

unde $\alpha = \min\{x, x_0, \dots, x_m\}$, $\beta = \max\{x, x_0, \dots, x_m\}$.

Dacă valorile funcției sunt tabelate, evaluarea lui ℓ_k necesită $2(m-1)$ înmulțiri, o împărțire și $2m$ scăderi. Întreaga evaluare necesită $2m(m+1)$ * / și $m(2m+3)$ +|-, total $\Theta(m^2)$.

2 Algoritmul lui Aitken

Uneori gradul este necunoscut sau precizia dorită poate fi atinsă utilizând un număr mai mic de noduri. Să introducem notațiile:

$$\begin{aligned}(L_{m-1}f)_{1,m}(x) &= \sum_{k=1}^m \ell_k(x)f(x_k), \\ (L_{m-1}f)_{0,m-1}(x) &= \sum_{k=0}^{m-1} \ell_k(x)f(x_k), \\ (L_m f)_{0,m}(x) &= \sum_{k=0}^m \ell_k(x)f(x_k).\end{aligned}\tag{4}$$

Algoritmul lui Aitken se bazează pe relația

$$(L_m f)_{0,m}(x) = \frac{\begin{vmatrix} (L_{m-1}f)_{1,m}(x) & x_0 - x \\ (L_{m-1}f)_{0,m-1}(x) & x_m - x \end{vmatrix}}{x_m - x_0}.$$

Metoda generează tabela următoare:

$$\begin{array}{cccccc} x_0 & f_{0,0} & & & & \\ x_1 & f_{1,0} & f_{1,1} & & & \\ x_2 & f_{2,0} & f_{2,1} & f_{2,2} & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ x_i & f_{i,0} & f_{i,1} & f_{i,2} & \cdots & f_{i,i} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \ddots \\ x_n & f_{n,0} & f_{n,1} & f_{n,2} & \cdots & f_{n,i} & \cdots & f_{n,n} \end{array}$$

unde $f_{i,0} = f(x_i)$, $i = 0, \dots, m$, și

$$f_{i,j+1} = \frac{1}{x_i - x_j} \begin{vmatrix} f_{j,j} & x_j - x \\ f_{i,j} & x_i - x \end{vmatrix}.\tag{5}$$

Se verifică ușor că $(L_i f)(x) = f_{i+1,i+1}$, $i = 0, \dots, n-1$, datorită unicității polinomului de interpolare Lagrange. Dacă interpolarea Lagrange converge, atunci $(f_{i,i})_{i \in \mathbb{N}}$ converge către $f(x)$ și $|f_{i,i} - f_{i-1,i-1}| \rightarrow 0$ când $i \rightarrow \infty$, deci relația $|f_{i,i} - f_{i-1,i-1}| \leq \varepsilon$ ar putea fi utilizată drept criteriu de oprire.

Algoritmul poate fi accelerat dacă sortăm nodurile crescător după distanța lor la x , i.e. $|x_i - x| \leq |x_j - x|$, dacă $i < j$.

Intrare: $m \in \mathbb{N}$, x , x_i , $f_i \in \mathbb{R}$, $i = 0, \dots, m$, $\varepsilon > 0$.

Ieșire: $f_{i,i}$.

P1. Sortează x_i crescător după $a_i = |x - x_i|$.

P2. For $i = 0, \dots, m$ set $f_{i,1} := f(x_i)$.

P3. For $i = 1, \dots, m$ do

P3.1. For $j = 0, \dots, i - 1$ do
 $y_{i,j} := x_i - x_j$;
 $f_{i,j+1} := ((x - x_i) * f_{jj} - (x - x_j) * f_{ij}) / y_{ij}$;
P3.2. If $|f_{i,i} - f_{i-1,i-1}| \leq \varepsilon$ go to P4.
P4. Extrage $f_{i,i}$.

3 Interpolare Lagrange baricentrică

Forma clasică a interpolării Lagrange are următoarele dezavantaje:

1. fiecare evaluare a lui $p(x)$ necesită $\Theta(m^2)$ adunări și înmulțiri;
2. adăugarea unei noi perechi de date (x_{m+1}, f_{m+1}) necesită reluarea tuturor calculelor.
3. procesul de calcul este numeric instabil.

Metoda lui Newton, odată ce s-a generat tabele de diferențe divizate, necesită un timp $\Theta(m)$, dar este instabilă.

3.1 O formulă Lagrange îmbunătățită

Notăm cu $f_i = f(x_i)$. Vom rescrie formulele (1)+(2) astfel ca $(L_m f)(x)$ să poată fi evaluat și actualizat cu $O(m)$ operații. Introducând

$$\ell(x) = (x - x_0)(x - x_1) \cdots (x - x_m) \quad (6)$$

ℓ_j se poate scrie ca $\ell_j(x) = \ell(x)/(x - x_j)$. Definind ponderile baricentrice prin

$$w_j = \frac{1}{\prod_{k \neq j} (x_j - x_k)}, \quad j = 0, \dots, m, \quad (7)$$

adică, $w_j = 1/\ell'(x_j)$, putem scrie ℓ_j sub forma

$$\ell_j(x) = \ell(x) \frac{w_j}{x - x_j}.$$

Acum PIL se scrie

$$(L_m f)(x) = \ell(x) \sum_{j=0}^m \frac{w_j}{x - x_j} f_j. \quad (8)$$

Avantajul este că putem calcula interpolantul Lagrange cu o formulă ce necesită $O(m^2)$ flops pentru calculul unor cantități independente de x , numerele w_j , urmate de $O(m)$ flops pentru evaluarea lui p , odată ce aceste numere sunt cunoscute.

Din (8) rezultă că actualizarea polinomului de interpolare la inserția unui nod nou necesită următoarele calcule:

- se împarte fiecare w_j , $j = 0..m$, prin $x_j - x_{m+1}$ (un flop pentru fiecare punct), cu un cost de $m + 1$ flops;
- se calculează w_{m+1} cu formula (7) cu alte $m + 1$ flops.

3.2 Metoda baricentrică

Interpolând funcția constantă 1 obținem

$$1 = \sum_{j=0}^m \ell_j(x) = \ell(x) \sum_{j=0}^m \frac{w_j}{x - x_j}. \quad (9)$$

Împărțind (8) cu expresia de mai sus și simplificând cu $\ell(x)$, obținem

$$p(x) = \frac{\sum_{j=0}^m \frac{w_j}{x - x_j} f_j}{\sum_{j=0}^m \frac{w_j}{x - x_j}}, \quad (10)$$

numită *formula baricentrică*.

La fel ca în (8), în (10) se poate adăuga o nouă pereche de date (x_{m+1}, f_{m+1}) și actualiza w_j în $O(m)$ flops.

3.3 Distribuții remarcabile ale nodurilor

În cazul unor noduri particulare se pot da formule explicite pentru ponderile baricentrice w_j . Pentru noduri echidistante în intervalul $[-1, 1]$, la distanța $h = 2/m$, se obține $w_j = (-1)^m \binom{m}{j} / (h^m m!)$, care după anularea (simplificarea) factorilor independenți de j ne dă

$$w_j = (-1)^j \binom{m}{j}. \quad (11)$$

Același rezultat se obține și pentru un interval arbitrar $[a, b]$, deoarece formula originală pentru w_j se înmulțește cu $2^m(b-a)^{-m}$, dar acest factor poate fi înlăturat prin simplificare.

Familia de *puncte Cebîșev* se poate obține proiectând puncte egal spațiate pe cercul unitate pe intervalul $[-1, 1]$. Pornind de la formula

$$w_j = \frac{1}{\ell'(x_j)}, \quad (12)$$

se pot obține formule explicite pentru ponderile w_j .

Punctele Cebîșev de speța I sunt date de

$$x_j = \cos \frac{(2j+1)\pi}{2m+2}, \quad j = 0, \dots, m.$$

Simplificând factorii independenți de j se obține

$$w_j = (-1)^j \sin \frac{(2j+1)\pi}{2m+2}. \quad (13)$$

Punctele Cebîșev de speța II sunt date de

$$x_j = \cos \frac{j\pi}{m}, \quad j = 0, \dots, m,$$

iar ponderile corespunzătoare sunt

$$w_j = (-1)^j \delta_j, \quad \delta_j = \begin{cases} 1/2, & j = 0 \text{ sau } j = m, \\ 1, & \text{altfel.} \end{cases}$$

Dăm codul MATLAB pentru interpolarea Lagrange baricentrică.

Funcția `barycentricInterpolation` calculează valorile polinomului de interpolare Lagrange folosind metoda baricentrică.

```
function ff=barycentricInterpolation(x,y,xx,c)
%BARYCENTRICINTERPOLATION - barycentric Lagrange interpolation
%call ff=barycentricInterpolation(x,y,xx,c)
%x - nodes
%y - function values
%xx - interpolation points
%c - barycentric weights
%ff - values of interpolation polynomial

n=length(x)-1;
numer = zeros(size(xx));
denom = zeros(size(xx));
exact = zeros(size(xx)); %test if xx=nodes
for j=1:n+1
    xdiff = xx-x(j);
    temp = c(j)./xdiff;
    numer = numer+temp*y(j);
    denom = denom+temp;
    exact(xdiff==0) = j;
end
ff = numer ./ denom;
jj = find(exact);
ff(jj) = y(exact(jj));
```

Funcția `barycentricweights` calculează ponderile baricentrice.

```
function c = barycentricweights( x )
%BARYCENTRICWEIGHTS - compute barycentric weights(coefficients)
%call c = barycentricweights( x )
%x - nodes
%c - weights

n=length(x)-1;
c=ones(1,n+1);
```

```

for j=1:n+1
    c(j)=prod(x(j)-x([1:j-1,j+1:n+1]));
end
c=1./c;
end

```

În cazul nodurilor Cebîșev de speța a doua sursa MATLAB este

```

function ff=ChebLagrange(y,xx,a,b)
%CHEBLAGRANGE - Lagrange interpolation for Chebyshev #2 points- barycentric
%call ff=ChebLagrange(y,xx,a,b)
%y - function values;
%xx - evaluation points
%a,b - interval
%ff - values of Lagrange interpolation polynomial

n = length(y)-1;
if nargin==2
    a=-1; b=1;
end
c = [1/2; ones(n-1,1); 1/2].*(-1).^((0:n)');
x = sort(cos((0:n)'*pi/n))*(b-a)/2+(a+b)/2;
ff=barycentricInterpolation(x,y,xx,c);
end

```

Probleme

1. Implementați o rutină pentru calculul valorilor polinomului de interpolare Lagrange când se dau punctele, nodurile și valorile funcției în noduri.
2. Reprezentați grafic polinoamele fundamentale când se dau gradul și nodurile.
3. Reprezentați pe același grafic f și $L_m f$.
4. Dându-se x , f , m și nodurile, aproximați $f(x)$ utilizând interpolarea Lagrange.
5. Implementați metoda baricentrică, în cazul general și pentru noduri Cebîșev #1 și #2.

Probleme practice

1. Datele de mai jos dau populația SUA în perioada 1900 – 2000 (în milioane de locuitori)

t	y
1900	75.995
1910	91.972
1920	105.711
1930	123.203
1940	131.669
1950	150.697
1960	179.323
1970	203.212
1980	226.505
1990	249.633
2000	281.422
2010	308.786

Approximați populația din 1975 și 2018.

2. Fie

$$f(x) = e^{x^2-1}.$$

Aproximați $f(1.25)$ utilizând valorile lui f în 1, 1.1, 1.2, 1.3 și 1.4 și dați o delimitare a erorii.

3. Aproximați $\sqrt{115}$ cu 3 zecimale exacte prin interpolare Lagrange.
4. Dați contraexemple pentru convergența interpolării Lagrange și studiați-le grafic:
- (a) contraexemplul lui Runge $f : [-5, 5] \rightarrow \mathbb{R}$, $f(x) = \frac{1}{1+x^2}$;
- (b) contraexemplul lui Bernstein $g : [-1, 1] \rightarrow \mathbb{R}$, $g(x) = |x|$;

ambele cu noduri echidistante și noduri Cebîșev de speța a doua.

5. Aproximați $\sin 5^\circ$ și $\cos 5^\circ$ folosind interpolarea Lagrange și tabela de valori

x	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$
$\sin x$	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1
$\cos x$	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0

Probleme suplimentare

- Se consideră funcția cu valori complexe $f : [-1, 1] \rightarrow \mathbb{C}$, data de $f(x) = (3 + \sin 10\pi x + \sin 61e^{0.8 \sin \pi x + 0.7}) e^{i\pi x}$. Să se reprezinte grafic valorile polinomului său de interpolare Lagrange pentru 671 de noduri Cebîșev de speța a doua în cel puțin 10000 de puncte din $[-1, 1]$. (Indicație: `plot(z)` pentru z complex este echivalent cu `plot(real(z), imag(z))`.)
- Implementați interpolarea Lagrange în MATLAB, în forma clasică și forma baricentrică, fără a folosi cicluri `for`.