

# Formula lui Taylor și aplicații

## O cărămidă importantă a analizei numerice

Radu T. Trîmbițaș

UBB

6 martie 2023

# Polinomul lui Taylor I

- Fie  $f : I \rightarrow \mathbb{R}$ ,  $f \in C^{n+1}(I)$ ,  $a \in \text{int}(I)$ . Dorim să găsim un polinom  $P$  de grad minim care să verifice condițiile

$$P(a) = f(a), P'(a) = f'(a), \dots, P^{(n)}(a) = f^{(n)}(a). \quad (1)$$

- Căutăm  $P$  sub forma

$$P(x) = a_0 + a_1(x - a) + \dots + a_n(x - a)^n \quad (2)$$

# Polinomul lui Taylor II

- Din (1) și (2) obținem

$$P(a) = a_0 = f(a)$$

$$P'(a) = a_1 = f'(a)$$

⋮

$$P^{(k)}(a) = k!a_k = f^{(k)}(a)$$

⋮

$$P^{(n)}(a) = n!a_n = f^{(n)}(a),$$

de unde rezultă

$$a_0 = f(a), a_1 = f'(a), \dots, a_k = \frac{f^{(k)}(a)}{k!}, \dots, a_n = \frac{f^{(n)}(a)}{n!} \quad (3)$$

# Polinomul lui Taylor III

- **Unicitatea:** presupunem că există  $Q \neq P$  care verifică (1). Notăm  $H := P - Q$ . Obținem

$$H(a) = 0, H'(a) = 0, \dots, H^{(n)}(a) = 0,$$

adică  $H \equiv 0$  ( $H$  identic nul)

- De ce se pune condiția  $P$  de grad minim?
- $P$  se va numi **polinomul lui Taylor** de gradul  $n$ , atașat funcției  $f$  în punctul  $a$  și se va nota cu  $(T_n f)(x)$

# Formula lui Taylor

- I interval,  $f : I \rightarrow \mathbb{R}$  o funcție derivabilă de  $n$  ori în punctul  $a \in I$ .  
**Polinomul lui Taylor** de gradul  $n$ , atașat funcției  $f$  în punctul  $a$ :

$$(T_n f)(x) = f(a) + \frac{x-a}{1!} f'(a) + \cdots + \frac{(x-a)^n}{n!} f^{(n)}(a)$$

- **Restul** de ordinul  $n$  al formulei lui Taylor în punctul  $x$

$$(R_n f)(x) = f(x) - (T_n f)(x)$$

- **formula lui Taylor** de ordinul  $n$  pentru funcția  $f$  în vecinătatea punctului  $a$ :

$$f(x) = (T_n f)(x) + (R_n f)(x)$$

sau

$$\begin{aligned} f(x) = & f(a) + \frac{x-a}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(a) + \cdots + \frac{(x-a)^n}{n!} f^{(n)}(a) \\ & + (R_n f)(x) \end{aligned}$$

# Expresii ale restului

- Are loc

$$(R_n f)(x) = \frac{(x-a)^n}{n!} \omega(x), \text{ cu } \lim_{x \rightarrow a} \omega(x) = 0.$$

- Dacă  $f \in C^{n+1}(I)$ , atunci  $\exists \theta \in (0, 1)$  astfel încât

$$(R_n f)(x) = \frac{(x-a)^{n+1} f^{(n+1)}(a + \theta(x-a))}{(n+1)!}$$

(restul în forma Lagrange)

$$(R_n f)(x) = \frac{(x-a)^{n+1} (1-\theta)^n f^{(n+1)}(a + \theta(x-a))}{n!}$$

(restul în forma Cauchy)

$$(R_n f)(x) = \int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t) dt$$

(restul în formă integrală)

# Demonstrația expresiei restului I

- Pornim de la expresia restului și o integrăm prin părți de  $n$  ori

$$\begin{aligned} \int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t) dt &= f^{(n)}(t) \frac{(x-t)^n}{n!} \Big|_a^x + \int_a^x f^{(n)}(t) \frac{(x-t)^{n-1}}{(n-1)!} dt \\ &= -f^{(n)}(a) \frac{(x-a)^n}{n!} + f^{(n-1)}(t) \frac{(x-t)^{n-1}}{(n-1)!} \Big|_a^x + \int_a^x f^{(n-1)}(t) \frac{(x-t)^{n-2}}{(n-2)!} dt \\ &\vdots \\ &= -f^{(n)}(a) \frac{(x-a)^n}{n!} - f^{(n-1)}(a) \frac{(x-a)^{n-1}}{(n-1)!} - \dots - f'(a) \frac{x-a}{1!} + \int_a^x f'(t) dt \\ &= -f^{(n)}(a) \frac{(x-a)^n}{n!} - f^{(n-1)}(a) \frac{(x-a)^{n-1}}{(n-1)!} - \dots - f'(a) \frac{x-a}{1!} + f(x) - f(a) \end{aligned}$$

## Demonstrația expresiei restului II

- de aici se obține

$$f(x) = f(a) + \frac{x-a}{1!}f(a) + \frac{(x-a)^2}{2!}f''(a) + \cdots + \frac{(x-a)^n}{n!}f^{(n)}(a) + \underbrace{\int_a^x \frac{(x-t)^n}{n!}f^{(n+1)}(t)dt}_{(R_n f)(x)}$$

- Pentru a obține forma Lagrange a restului folosim teorema a două de medie a calculului integral: fie  $u, v : [\alpha, \beta] \rightarrow \mathbb{R}$ ,  $u, v$  continue și  $u$  are semn constant pe  $[\alpha, \beta]$ . Atunci există  $\xi \in [\alpha, \beta]$  astfel încât

$$\int_{\alpha}^{\beta} u(t)v(t)dt = v(\xi) \int_{\alpha}^{\beta} u(t)dt$$

## Demonstrația expresiei restului III

- Alegând  $\alpha = a$ ,  $\beta = x$ ,  $u(t) = \frac{(x-t)^n}{n!}$ ,  $v(t) = f^{(n+1)}(t)$  vom obține

$$\begin{aligned}(R_n f)(x) &= \int_a^x \frac{(x-t)^n}{n!} f^{(n+1)}(t) dt = f^{(n+1)}(\xi) \int_a^x \frac{(x-t)^n}{n!} dt \\&= f^{(n+1)}(\xi) \left[ -\frac{(x-t)^{n+1}}{(n+1)!} \right] \Big|_a^x = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-a)^{n+1}\end{aligned}$$

## O formă utilă

Dacă  $f \in C^{n+1}(I)$ , dezvoltând  $f(x+h)$  în jurul lui  $x$ ,

$$f(x+h) = \sum_{k=0}^n \frac{f^{(k)}(x)}{k!} h^k + (R_n f)(x),$$

iar o formă pentru rest este

$$(R_n f)(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} h^{n+1},$$

unde  $\xi$  este între  $x$  și  $x+h$ .

# Formula lui Maclaurin

- Dacă în formula lui Taylor se ia  $a = 0$ , se obține formula lui Maclaurin

$$f(x) = f(0) + xf'(0) + \cdots + \frac{x^n}{n!} f^{(n)}(0) + (R_n f)(x),$$

unde

$$(R_n f)(x) = \frac{x^{n+1}}{(n+1)!} f^{(n+1)}(\theta x), \quad \theta \in (0, 1).$$

- Exemple de dezvoltări uzuale

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + R_n(x); \quad (4)$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + R_{2n+1}(x); \quad (5)$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + R_{2n}(x); \quad (6)$$

## Alte dezvoltari uzuale

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \cdots + (-1)^n \frac{x^{n+1}}{n+1} + R_{n+1}(x); \quad (7)$$

$$(1+x)^k = 1 + \binom{k}{1}x + \binom{k}{2}x^2 + \cdots + \binom{k}{n}x^n + R_n(x), \quad (8)$$

unde

$$\binom{k}{n} = \frac{k(k-1)\dots(k-n+1)}{n!}.$$



Brook Taylor (1685-1731)



Colin Maclaurin (1698-1768)

# Aplicații I

## Problemă

Să se scrie formula lui MacLaurin pentru funcția  $f : [-a, \infty) \rightarrow \mathbb{R}$ ,  $f(x) = \sqrt{a+x}$ ,  $a > 0$ .

**Soluție.** Scriem  $f(x) = \sqrt{a+x} = \sqrt{a}\left(1 + \frac{x}{a}\right)^{\frac{1}{2}}$ ; se obține

$$\begin{aligned}f(x) = \sqrt{a} &\left[ 1 + \frac{1}{2} \frac{x}{a} + (-1)^1 \frac{1}{2^2} \frac{1}{2!} \left(\frac{x}{a}\right)^2 + (-1)^2 \frac{1}{2^3} \frac{1}{3!} \left(\frac{x}{a}\right)^3 + \dots \right. \\&\quad \left. + (-1)^{n-1} \frac{1 \cdot 3 \cdot 5 \dots (2n-3)}{n! 2^n} \left(\frac{x}{a}\right)^n + (R_n f)(x) \right].\end{aligned}$$



## Aplicații II

### Problemă

Să se determine numărul natural  $n$  astfel ca pentru  $a = 0$  și  $f : \mathbb{R} \rightarrow \mathbb{R}$ ,  $f(x) = e^x$ ,  $T_n f$  să aproximeze  $f$  în  $[-1, 1]$  cu trei zecimale exacte.

**Soluție.** Impunem condiția  $| (R_n f)(x) | = \left| \frac{x^{n+1} e^{\theta x}}{(n+1)!} \right| < 10^{-3}$ . Deoarece  $\theta x < 1$ ,  $e^{\theta x} < e < 3$ , avem

$$\left| \frac{x^{n+1}}{(n+1)!} e^{\theta x} \right| < \frac{3}{(n+1)!} < 10^{-3} \Rightarrow n = 6.$$

În particular, luând  $x = 1$ , obținem

$$e - \left( 1 + \frac{1}{1!} + \cdots + \frac{1}{6!} \right) < \frac{1}{1000}.$$

# Aplicații III

## Problemă

Să se aproximeze  $\sqrt[3]{999}$  cu 12 zecimale exacte.

**Soluție.** Avem

$$\sqrt[3]{999} = 10 \left(1 - \frac{1}{1000}\right)^{\frac{1}{3}}.$$

Folosim formula (8) pentru  $k = 1/3$ ,  $x = -\frac{1}{1000}$ . Într-o serie alternată modulul erorii este mai mic decât modulul primului termen neglijat.

$$|(R_n f)(x)| < \left| \left(\frac{\frac{1}{3}}{n}\right) 10^{-3n} \right|.$$

Pentru  $n = 4$ , avem

$$|(R_n f)(x)| < \frac{10}{243} 10^{-12} = \frac{1}{24300000000000} = 4.1152 \times 10^{-14}. \blacksquare$$

## Formula lui Taylor bidimensională

- Pentru o funcție  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ , o expresie a formulei lui Taylor este

$$f(a+h, b+k) = \sum_{i=0}^n \frac{1}{i!} \left( h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^i f(a, b) + R_n(h, k) \quad (9)$$

$$R_n(h, k) = \frac{1}{(n+1)!} \left( h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^{n+1} f(a + \theta h, b + \theta k),$$

cu  $\theta \in (0, 1)$ . Semnificația termenilor diferențiali este

$$\left( h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^0 f(a, b) = f(a, b)$$

$$\left( h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^1 f(a, b) = \left( h \frac{\partial f}{\partial x} + k \frac{\partial f}{\partial y} \right)(a, b)$$

$$\left( h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^2 f(a, b) = \left( h^2 \frac{\partial^2 f}{\partial x^2} + 2hk \frac{\partial^2 f}{\partial x \partial y} + k^2 \frac{\partial^2 f}{\partial y^2} \right)(a, b).$$

# Formula lui Taylor bidimensională

Teorema următoare precizează condițiile de aplicabilitate ale formulei (9).

## Teoremă

*Dacă toate derivatele parțiale de ordinul  $n + 1$  ale lui  $f$  sunt continue în dreptunghiul definit de  $|x - a| \leq |h|$  și  $|y - b| \leq k$ , atunci există  $\theta \in (0, 1)$  pentru care are loc (9).*

# Un exemplu

## Problemă

Scriți dezvoltarea Taylor a lui  $f(x, y) = \cos xy$ , pentru  $n = 1$ .

**Soluție.** Aplicând formula (9) se obține

$$\cos[(a+h)(b+k)] = \cos ab - (hb + ak) \sin ab + R_1(h, k),$$

unde  $R_1$  este suma a trei termeni

$$\begin{aligned} & -\frac{1}{2}h^2(b+\theta k)^2 \cos [(a+\theta h)(b+\theta k)] \\ & - hk \{(a+\theta h)(b+\theta k) \cos [(a+\theta h)(b+\theta k)] + \sin [(a+\theta h)(b+\theta k)]\} \\ & -\frac{1}{2}k^2(a+\theta h)^2 \cos [(a+\theta h)(b+\theta k)]. \end{aligned}$$



# Exemple

- Exemple Maple [exempletaylor.html](#)
- Exemplu Maple Taylor 2D [taylor2d.html](#)
- Exemple MATLAB Symbolic Math Toolbox [extaylor.html](#)

# Aproximare rațională

- Se numește *funcție rațională de ordin  $(n, m)$*  o funcție de forma

$$r(x) = \frac{p(x)}{q(x)},$$

unde  $p$  este un polinom de grad  $n$ , iar  $q$  este un polinom de grad  $m$ .

- Numărul  $N = m + n$  se numește gradul funcției raționale
- Sunt  $n + 1 + m + 1 = m + n + 2$  coeficienți, dar numai  $n + m + 1$  sunt independenți (putem împărti cu unul dintre coeficienții nenului)
- Polinoamele pot fi considerate cazuri particulare de funcții raționale cu grad  $q(x) = 0$ .

# Aproximare Padé

- Este un analog rațional al formulei lui Taylor
- Fie o funcție rațională de forma de ordin  $(m, n)$

$$R(x) = \frac{p(x)}{q(x)} = \frac{p_0 + p_1x + \cdots + p_nx^n}{q_0 + q_1x + \cdots + q_mx^m}$$

- Dacă  $q_0 \neq 0$ , putem împărți cu el; wlog  $q_0 = 1$ ; sunt coeficienți  $m + n + 1$  independenți
- Se numește *aproximare Padé* a lui  $f$  de ordin  $(n, m)$  (de grad  $N = n + m$ ) o funcție rațională  $R$  de ordin  $(n, m)$  care verifică

$$f^{(k)}(0) - R^{(k)}(0) = 0, \quad k = 0, \dots, m + n \quad (10)$$

- Pentru  $m = 0$   $R$  este polinomul MacLaurin de grad  $n$

# Calculul aproximantei I

- Considerăm diferența

$$\begin{aligned}f(x) - R(x) &= f(x) - \frac{p(x)}{q(x)} = \frac{f(x)q(x) - p(x)}{q(x)} \\&= \frac{f(x)\sum_{k=0}^m q_k x^k - \sum_{k=0}^n p_k x^k}{\sum_{k=0}^m q_k x^k}\end{aligned}$$

- Considerăm dezvoltarea Mac Laurin a lui  $f$ ,  $f(x) = \sum_{k=0}^{\infty} c_k x^k$  și o înlocuim în formula precedentă

$$f(x) - R(x) = \frac{\left(\sum_{k=0}^{\infty} c_k x^k\right) \left(\sum_{k=0}^m q_k x^k\right) - \sum_{k=0}^n p_k x^k}{\sum_{k=0}^m q_k x^k}$$

- $q_k = ?, k = 1, \dots, m$ ;  $p_k = ?, k = 0, \dots, n$

## Calculul aproximantei II

- (14)  $\Rightarrow$

$$\Delta = (c_0 + c_1x + \cdots)(1 + q_1x + \cdots + q_mx^m) - p_0 + p_1x + \cdots + p_nx^n \quad (11)$$

nu are nici un termen de grad  $< N$ .

- Definim  $p_{n+1} = p_{n+2} = \cdots = p_N = 0$  și  
 $q_{m+1} = q_{m+2} = \cdots = q_N = 0$
- coeficientul lui  $x^k$  din (15) se scrie mai compact

$$\left( \sum_{i=0}^k c_i q_{k-i} \right) - p_k$$

- Pentru  $k = 0, 1, \dots, N$  el trebuie să fie nul; se obține sistemul

$$\sum_{i=0}^k c_i q_{k-i} = p_k, \quad k = 0, 1, \dots, N \quad (12)$$

cu necunoscutele  $p_0, \dots, p_n; q_1, \dots, q_m$

# Rezolvarea sistemului în practică I

- Ultimele  $m$  ecuații ne dau, ținând cont că  $q_0 = 1$  și  $p_j = 0$ , pentru  $j > n$

$$\sum_{i=0}^{k-1} c_i q_{k-i} = -c_k, \quad k = n+1, \dots, n+m \quad (13)$$

- Odată determinați coeficienții  $q_1, \dots, q_m$  coeficienții  $p_k$ ,  $k = 0, \dots, n$  se obțin cu

$$p_k = \sum_{i=0}^k c_i q_{k-i}$$

## Rezolvarea sistemului în practică II

- Matricea  $A$  a sistemului (17) este matrice Toeplitz (diagonalele sunt constante)

$$A = \begin{bmatrix} c_n & c_{n-1} & \cdots & c_{n-(m-2)} & c_{n-(m-1)} \\ c_{n+1} & c_n & c_{n-1} & \cdots & c_{n-(m-2)} \\ c_{n+2} & c_{n+1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & c_n & c_{n-1} \\ c_{n+(m-1)} & \cdots & c_{n+2} & c_{n+1} & c_n \end{bmatrix}$$

## Exemplu I

- Aproximarea Padé de ordin  $(1, 1)$  pentru  $f(x) = e^x$ ;
- Avem  $c_k = \frac{1}{k!}$ ;  $N = 1 + 1 = 2$ ; forma aproximantei este

$$R = \frac{p_0 + p_1 x}{1 + q_1 x}$$

- Calculăm numărătorul lui  $f - R$

$$\begin{aligned} & \left(1 + x + \frac{x^2}{2}\right) (1 + q_1 x) - (p_0 + p_1 x) \\ &= \frac{1}{2} q_1 x^3 + \left(q_1 + \frac{1}{2}\right) x^2 + (q_1 - p_1 + 1) x + (1 - p_0) \end{aligned}$$

## Exemplu II

- Sistemul (16) devine

$$\begin{cases} 1 - p_0 = 0 \\ q_1 - p_1 + 1 = 0 \\ q_1 + \frac{1}{2} = 0 \end{cases}$$

Soluția  $p_0 = 1$ ,  $q_1 = -\frac{1}{2}$ ,  $p_1 = \frac{1}{2}$ .

- Am obținut aproximanta

$$R = \frac{1 + \frac{1}{2}x}{1 - \frac{1}{2}x}$$

- Eroarea

$$e^x - \frac{1 + \frac{1}{2}x}{1 - \frac{1}{2}x} = \frac{-\frac{1}{12}x^3 + \dots}{1 - \frac{1}{2}x}$$

# Avantaje

- În loc să utilizăm polinoame de grad mare, putem utiliza câturi de polinoame de grad mic;
- Poate aproxima funcții cu singularități;
- Dă adesea aproximări mai bune decât seriile Taylor trunchiate;
- Uneori funcționează chiar și atunci când seria Taylor nu converge!

# Aproximare rațională

- Se numește *funcție rațională de ordin  $(n, m)$*  o funcție de forma

$$r(x) = \frac{p(x)}{q(x)},$$

unde  $p$  este un polinom de grad  $n$ , iar  $q$  este un polinom de grad  $m$ .

- Numărul  $N = m + n$  se numește gradul funcției raționale
- Sunt  $n + 1 + m + 1 = m + n + 2$  coeficienți, dar numai  $n + m + 1$  sunt independenți (putem împărti cu unul dintre coeficienții nenului)
- Polinoamele pot fi considerate cazuri particulare de funcții raționale cu grad  $q(x) = 0$ .

# Aproximare Padé

- Este un analog rațional al formulei lui Taylor
- Fie o funcție rațională de forma de ordin  $(m, n)$

$$R(x) = \frac{p(x)}{q(x)} = \frac{p_0 + p_1x + \cdots + p_nx^n}{q_0 + q_1x + \cdots + q_mx^m}$$

- Dacă  $q_0 \neq 0$ , putem împărți cu el; wlog  $q_0 = 1$ ; sunt  $m+n+1$  coeficienți independenți
- Se numește *aproximare Padé* a lui  $f$  de ordin  $(n, m)$  (de grad  $N = n+m$ ) o funcție rațională  $R$  de ordin  $(n, m)$  care verifică

$$f^{(k)}(0) - R^{(k)}(0) = 0, \quad k = 0, \dots, m+n \quad (14)$$

- Pentru  $m = 0$   $R$  este polinomul MacLaurin de grad  $n$

# Calculul aproximantei I

- Considerăm diferența

$$\begin{aligned}f(x) - R(x) &= f(x) - \frac{p(x)}{q(x)} = \frac{f(x)q(x) - p(x)}{q(x)} \\&= \frac{f(x)\sum_{k=0}^m q_k x^k - \sum_{k=0}^n p_k x^k}{\sum_{k=0}^m q_k x^k}\end{aligned}$$

- Considerăm dezvoltarea Mac Laurin a lui  $f$ ,  $f(x) = \sum_{k=0}^{\infty} c_k x^k$  și o înlocuim în formula precedentă

$$f(x) - R(x) = \frac{\left(\sum_{k=0}^{\infty} c_k x^k\right) \left(\sum_{k=0}^m q_k x^k\right) - \sum_{k=0}^n p_k x^k}{\sum_{k=0}^m q_k x^k}$$

- $q_k = ?, k = 1, \dots, m; p_k = ?, k = 0, \dots, n$

## Calculul aproximantei II

- (14)  $\Rightarrow$

$$\Delta = (c_0 + c_1x + \cdots)(1 + q_1x + \cdots + q_mx^m) - p_0 + p_1x + \cdots + p_nx^n \quad (15)$$

nu are nici un termen de grad  $< N$ .

- Definim  $p_{n+1} = p_{n+2} = \cdots = p_N = 0$  și  
 $q_{m+1} = q_{m+2} = \cdots = q_N = 0$
- coeficientul lui  $x^k$  din (15) se scrie mai compact

$$\left( \sum_{i=0}^k c_i q_{k-i} \right) - p_k$$

- Pentru  $k = 0, 1, \dots, N$  el trebuie să fie nul; se obține sistemul

$$\sum_{i=0}^k c_i q_{k-i} = p_k, \quad k = 0, 1, \dots, N \quad (16)$$

cu necunoscutele  $p_0, \dots, p_n; q_1, \dots, q_m$

# Rezolvarea sistemului în practică I

- Ultimele  $m$  ecuații ne dau, ținând cont că  $q_0 = 1$  și  $p_j = 0$ , pentru  $j > n$

$$\sum_{i=0}^{k-1} c_i q_{k-i} = -c_k, \quad k = n+1, \dots, n+m \quad (17)$$

- Odată determinați coeficienții  $q_1, \dots, q_m$  coeficienții  $p_k$ ,  $k = 0, \dots, n$  se obțin cu

$$p_k = \sum_{i=0}^k c_i q_{k-i}$$

## Rezolvarea sistemului în practică II

- Matricea  $A$  a sistemului (17) este matrice Toeplitz (diagonalele sunt constante)

$$A = \begin{bmatrix} c_n & c_{n-1} & \cdots & c_{n-(m-2)} & c_{n-(m-1)} \\ c_{n+1} & c_n & c_{n-1} & \cdots & c_{n-(m-2)} \\ c_{n+2} & c_{n+1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & c_n & c_{n-1} \\ c_{n+(m-1)} & \cdots & c_{n+2} & c_{n+1} & c_n \end{bmatrix}$$

## Exemplu I

- Aproximarea Padé de ordin  $(1, 1)$  pentru  $f(x) = e^x$ ;
- Avem  $c_k = \frac{1}{k!}$ ;  $N = 1 + 1 = 2$ ; forma aproximantei este

$$R = \frac{p_0 + p_1 x}{1 + q_1 x}$$

- Calculăm numărătorul lui  $f - R$

$$\begin{aligned} & \left(1 + x + \frac{x^2}{2}\right) (1 + q_1 x) - (p_0 + p_1 x) \\ &= \frac{1}{2} q_1 x^3 + \left(q_1 + \frac{1}{2}\right) x^2 + (q_1 - p_1 + 1) x + (1 - p_0) \end{aligned}$$

## Exemplu II

- Sistemul (16) devine

$$\begin{cases} 1 - p_0 = 0 \\ q_1 - p_1 + 1 = 0 \\ q_1 + \frac{1}{2} = 0 \end{cases}$$

Soluția  $p_0 = 1$ ,  $q_1 = -\frac{1}{2}$ ,  $p_1 = \frac{1}{2}$ .

- Am obținut aproximanta

$$R = \frac{1 + \frac{1}{2}x}{1 - \frac{1}{2}x}$$

- Eroarea

$$e^x - \frac{1 + \frac{1}{2}x}{1 - \frac{1}{2}x} = \frac{-\frac{1}{12}x^3 + \dots}{1 - \frac{1}{2}x}$$

# Avantaje

- În loc să utilizăm polinoame de grad mare, putem utiliza câturi de polinoame de grad mic;
- Poate aproxima funcții cu singularități;
- Dă adesea aproximări mai bune decât seriile Taylor trunchiate;
- Uneori funcționează chiar și atunci când seria Taylor nu converge!

# Bibliografie I

- W. Cheney, David Kinkaid, *Numerical Mathematics and Computing*, 6th edition, Brooks/Cole, 2008
- J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, 2nd ed., Springer Verlag, 1992.
- C. Ueberhuber, *Numerical Computation. Methods, Software and Analysis*, vol. I, II, Springer Verlag, Berlin, Heidelberg, New York, 1997.
- E. W. Cheney, *Introduction to Approximation Theory*, 2nd ed., AMS 1982

# Teoria erorilor și aritmetică în virgulă flotantă

## Erorile sunt omniprezente

Radu Tiberiu Trîmbițaș

Universitatea “Babeș-Bolyai”

7 martie 2023

# Tipuri de erori

Aprecierea preciziei rezultatelor calculelor este un obiectiv important în Analiza numerică. Se disting mai multe tipuri de erori care pot limita această precizie:

- ① **erori în datele de intrare** - sunt în afara (dincolo de) controlului calculelor. Ele se pot datora, de exemplu, imperfecțiunilor inerente ale măsurătorilor fizice.
- ② **erori de rotunjire** - apar dacă se fac calcule cu numere a căror reprezentare se restrânge la un număr finit de cifre.
- ③ **erori de aproximare** -multe metode nu dau soluția exactă a problemei  $P$ , ci a unei probleme mai simple  $\tilde{P}$ , care aproximează  $P$ : integralele se aproximează prin sume finite, derivatele prin diferențe (divizate), etc. Aceste erori se numesc **erori de discretizare**.

## Exemplu de eroare de aproximare

- (P) Dorim să approximăm

$$e = 1 + \frac{1}{1!} + \cdots + \frac{1}{n!} + \dots$$

- Problema se înlocuiește cu problema mai simplă  $(\tilde{P})$  a însumării unui număr finit de termeni - eroare se trunchiere

$$(\tilde{P}) \quad e = 1 + \frac{1}{1!} + \cdots + \frac{1}{n!}.$$

- În acest capitol ne interesează doar erorile în datele de intrare și erorile de rotunjire.

# Probleme numerice

- Combinarea dintre o problemă matematică (PM), (de natură constructivă) și specificațiile de precizie ale rezultatului (SP) se numește **problemă numerică**.

# Probleme numerice

- Combinarea dintre o problemă matematică (PM), (de natură constructivă) și specificațiile de precizie ale rezultatului (SP) se numește **problemă numerică**.
- Exemplu: Fie  $f : \mathbb{R} \rightarrow \mathbb{R}$  și  $x \in \mathbb{R}$ . Dorim să calculăm  $y = f(x)$ . În general  $x$  nu este reprezentabil în calculator; din acest motiv vom lucra cu o aproximare  $x^*$  a sa,  $x^* \approx x$ . De asemenea este posibil ca  $f$  să nu poată fi calculată exact; vom înlocui  $f$  cu o aproximantă a sa  $f_A$ . Valoarea calculată în calculator va fi  $f_A(x^*)$ . Deci problema numerică este următoarea:

# Probleme numerice

- Combinarea dintre o problemă matematică (PM), (de natură constructivă) și specificațiile de precizie ale rezultatului (SP) se numește **problemă numerică**.
- Exemplu: Fie  $f : \mathbb{R} \rightarrow \mathbb{R}$  și  $x \in \mathbb{R}$ . Dorim să calculăm  $y = f(x)$ . În general  $x$  nu este reprezentabil în calculator; din acest motiv vom lucra cu o aproximare  $x^*$  a sa,  $x^* \approx x$ . De asemenea este posibil ca  $f$  să nu poată fi calculată exact; vom înlocui  $f$  cu o aproximantă a sa  $f_A$ . Valoarea calculată în calculator va fi  $f_A(x^*)$ . Deci problema numerică este următoarea:

PM.

dându-se  $x$  și  $f$ , să se calculeze  $f(x)$ ;

# Probleme numerice

- Combinarea dintre o problemă matematică (PM), (de natură constructivă) și specificațiile de precizie ale rezultatului (SP) se numește **problemă numerică**.
- Exemplu: Fie  $f : \mathbb{R} \rightarrow \mathbb{R}$  și  $x \in \mathbb{R}$ . Dorim să calculăm  $y = f(x)$ . În general  $x$  nu este reprezentabil în calculator; din acest motiv vom lucra cu o aproximare  $x^*$  a sa,  $x^* \approx x$ . De asemenea este posibil ca  $f$  să nu poată fi calculată exact; vom înlocui  $f$  cu o aproximantă a sa  $f_A$ . Valoarea calculată în calculator va fi  $f_A(x^*)$ . Deci problema numerică este următoarea:

PM. dându-se  $x$  și  $f$ , să se calculeze  $f(x)$ ;

SP.  $|f(x) - f_A(x^*)| < \varepsilon$ ,  $\varepsilon$  dat.

- $X$  spațiu liniar normat,  $A \subseteq X$ ,  $x \in X$ . Un element  $x^* \in A$  se numește **aproximantă** a lui  $x$  din  $A$  (notație  $x^* \approx x$ ).
- $x^* \approx x$  o aproximantă a lui  $x$ , diferența  $\Delta x = x - x^*$  se numește **eroare**, iar

$$\|\Delta x\| = \|x^* - x\| \quad (1)$$

se numește **eroare absolută**.

- Raportul

$$\delta x = \frac{\|\Delta x\|}{\|x\|}, \quad x \neq 0 \quad (2)$$

se numește **eroare relativă**.

- Deoarece în practică  $x$  este necunoscut, se folosește aproximarea  $\delta x = \frac{\|\Delta x\|}{\|x^*\|}$ . Dacă  $\|\Delta x\|$  este mic comparativ cu  $\|x^*\|$ , atunci approximanta este bună.

# Eroarea propagată

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $x = (x_1, \dots, x_n)$ ,  $x^* = (x_1^*, \dots, x_n^*)$ . Dorim să evaluăm eroarea absolută și relativă  $\Delta f$  și respectiv  $\delta f$  când se aproximează  $f(x)$  prin  $f(x^*)$ .
- Aceste erori se numesc **erori propagate**, deoarece ne spun cum se propagă eroarea inițială (absolută sau relativă) pe parcursul calculării lui  $f$ .
- Presupunem  $x = x^* + \Delta x$ ,  $\Delta x = (\Delta x_1, \dots, \Delta x_n)$ . Aplicăm formula lui Taylor

$$\begin{aligned}\Delta f &= f(x_1^* + \Delta x_1, \dots, x_n^* + \Delta x_n) - f(x_1^*, \dots, x_n^*) \\ &= \sum_{i=1}^n \Delta x_i \frac{\partial f}{\partial x_i^*}(x_1^*, \dots, x_n^*) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \Delta x_i \Delta x_j \frac{\partial^2 f}{\partial x_i^* \partial x_j^*}(\theta),\end{aligned}$$

$$\theta \in [(x_1^*, \dots, x_n^*), (x_1^* + \Delta x_1, \dots, x_n^* + \Delta x_n)].$$

## Eroarea propagată II

- neglijând termenii de ordinul al doilea (mici) obținem

$$\Delta f \approx \sum_{i=1}^n \Delta x_i \frac{\partial f}{\partial x_i^*}(x_1^*, \dots, x_n^*). \quad (3)$$

- Pentru eroarea relativă avem

$$\begin{aligned} \delta f &= \frac{\Delta f}{f} \approx \sum_{i=1}^n \Delta x_i \frac{\frac{\partial f}{\partial x_i^*}(x^*)}{f(x^*)} \\ &= \sum_{i=1}^n \delta x_i \frac{x_i^* \frac{\partial f}{\partial x_i^*}(x^*)}{f(x^*)} \end{aligned} \quad (4)$$

## Eroarea propagată III

- Problema inversă: cu ce precizie trebuie approximate datele pentru ca rezultatul să aibă o precizie dată?
- Adică, dându-se  $\varepsilon > 0$ , cât trebuie să fie  $\Delta x_i$  sau  $\delta x_i$ ,  $i = \overline{1, n}$  astfel încât  $\Delta f$  sau  $\delta f < \varepsilon$ ?
- **principiul efectelor egale**: se presupune că toți termenii care intervin în (3) sau (4) au același efect, adică

$$\frac{\partial f}{\partial x_1^*}(x^*)\Delta x_1 = \dots = \frac{\partial f}{\partial x_n^*}(x^*)\Delta x_n.$$

- se obține

$$\Delta x_i \approx \frac{\Delta f}{n \left| \frac{\partial f}{\partial x_i^*}(x^*) \right|}. \quad (5)$$

$$\delta x_i = \frac{\delta f}{n \left| \frac{x_i^* \frac{\partial}{\partial x_i^*} f(x^*)}{f(x^*)} \right|}. \quad (6)$$

## Exemple

**Exemplu.** Găsiți o margine a erorii absolute și relative pentru volumul sferei  $V = \frac{\pi d^3}{6}$  cu diametrul egal cu  $3.7\text{cm} \pm 0.04\text{cm}$  și  $\pi \approx 3.14$ .

- Calculăm derivatele parțiale

$$\frac{\partial V}{\partial \pi} = \frac{1}{6}d^3 = 8.44, \quad \frac{\partial V}{\partial d} = \frac{1}{2}\pi d^2 = 21.5.$$

- Aplicând formula (3) și definiția erorii relative obținem:

$$\Delta V = \left| \frac{\partial V}{\partial \pi} \right| |\Delta \pi| + \left| \frac{\partial V}{\partial d} \right| |\Delta d| = 8.44 + 21.5 \cdot 0.05 \approx 1.0888,$$

$$\delta_V = \frac{1.0888}{274} \approx 4\%.$$

## Exemple - continuare

**Exemplu.** Un cilindru are raza  $R \approx 2m$ , înălțimea  $H \approx 3m$ . Cu ce erori absolute trebuie determinate  $R$ ,  $H$  și  $\pi$  astfel încât  $V$  să poată fi calculat cu o eroare  $< 0.1m^3$ .

Se aplică principiul efectelor egale (5):

$$V = \pi R^2 H, \quad \Delta V = 0.1m^3,$$

$$\frac{\partial V}{\partial \pi} = R^2 H = 12, \quad \frac{\partial V}{\partial R} = 2\pi RH = 37.7, \quad \frac{\partial V}{\partial H} = \pi R^2 = 12.6.$$

$n = 3$ , erorile absolute ale argumentelor:

$$\Delta \pi \approx \frac{\Delta V}{3 \frac{\partial V}{\partial \pi}} = \frac{0.1}{3 \cdot 12} < 0.003,$$

$$\Delta R \approx \frac{0.1}{3 \cdot 37.7} < 0.001,$$

$$\Delta H \approx \frac{0.1}{3 \cdot 12.6} < 0.003.$$

# Aritmetică în virgulă flotantă

## Parametrii reprezentării

- Parametrii reprezentării în virgulă flotantă sunt următoarele numere întregi
  - **baza**  $\beta$  (întotdeauna pară);
  - **precizia**  $p$ ;
  - **exponentul maxim**  $e_{\max}$ ;
  - **exponentul minim**  $e_{\min}$ ;
- În general, un număr în virgulă flotantă se reprezintă sub forma

$$x = \pm d_0.d_1d_2 \dots d_{p-1} \times \beta^e, \quad 0 \leq d_i < \beta, \quad e_{\min} \leq e \leq e_{\max} \quad (7)$$

$d_0.d_1d_2 \dots d_{p-1}$  - **semnificant** sau **fracție** sau **mantisă**,  $e$  **exponent**.

- Valoarea lui  $x$  este

$$\pm (d_0 + d_1\beta^{-1} + d_2\beta^{-2} + \dots + d_{p-1}\beta^{-(p-1)})\beta^e. \quad (8)$$

# Parametrii reprezentării

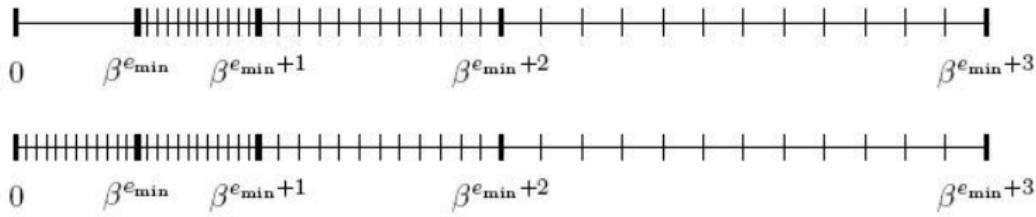
- Unicitatea se asigură prin **normalizare**: se modifică reprezentarea (nu valoarea) astfel încât  $d_0 \neq 0$ .
- Zero se reprezintă ca  $1.0 \times \beta^{e_{\min}-1}$
- Ordinea numerică ușuală a numerelor reale nenegative corespunde ordinii lexicografice a reprezentării lor flotante (cu exponentul în stânga semnificantului).
- **număr în virgulă flotantă** = număr real care poate fi reprezentat exact în virgulă flotantă

# Numere denormalizate

- După normalizarea semnificanților ramâne un „gol” între 0 și  $\beta^{e_{\min}}$
- Aceasta poate avea ca efect  $x - y = 0$  chiar dacă  $x \neq y$ , iar un fragment de cod de tipul **if**  $x \neq y$  **then**  $z = 1/(x - y)$  poate eșua
- Soluție: se admit semnificanți nenormalizați când exponentul este  $e_{\min}$  (gradual underflow). Aceste numere se numesc **numere denormalizate**. Ele garantează că

$$x = y \iff x - y = 0$$

- Distribuția fără denormalizare și cu denormalizare



# Parametrii reprezentării

- Mulțimea numerelor în virgulă flotantă pentru un set de parametri date ai reprezentării se va nota cu

$$\mathbb{F}(\beta, p, e_{\min}, e_{\max}, \text{denorm}), \quad \text{denorm} \in \{\text{true}, \text{false}\}.$$

- Această mulțime nu coincide cu  $\mathbb{R}$  din următoarele motive:
  - este o submulțime finită a lui  $\mathbb{Q}$ ;
  - pentru  $x \in \mathbb{R}$  putem avea  $|x| > \beta \times \beta^{e_{\max}}$  (depășire superioară) sau  $|x| < 1.0 \times \beta^{e_{\min}}$  (depășire inferioară).
- Operațiile aritmetice uzuale pe  $\mathbb{F}(\beta, p, e_{\min}, e_{\max}, \text{denorm})$  se notează cu  $\oplus, \ominus, \otimes, \oslash$ , iar funcțiile uzuale cu SIN, COS, EXP, LN, SQRT și.a.m.d.  $(\mathbb{F}, \oplus, \otimes)$  nu este corp deoarece

$$(x \oplus y) \oplus z \neq x \oplus (y \oplus z) \quad (x \otimes y) \otimes z \neq x \otimes (y \otimes z)$$
$$(x \oplus y) \otimes z \neq x \otimes z \oplus y \otimes z.$$

## Erori

- Eroarea relativă
- **ulps** – units in the last place (unități în ultima poziție): dacă  $z = d_0.d_1d_2 \dots d_{p-1} \dots \times \beta^e$ , atunci eroarea este

$$|d_0.d_1d_2 \dots d_{p-1} - z/\beta^e| \beta^{p-1} \text{ulps.}$$

- Eroarea relativă ce corespunde la  $\frac{1}{2}$ ulps este

$$\frac{1}{2}\beta^{-p} \leq \frac{1}{2} \text{ulps} \leq \frac{\beta}{2}\beta^{-p},$$

căci eroarea absolută este  $\underbrace{0.0 \dots 0}_{p} \beta' \times \beta^e$ , cu  $\beta' = \frac{\beta}{2}$ . Valoarea  $\text{eps} = \frac{\beta}{2}\beta^{-p}$  se numește **epsilon-ul mașinii**.

- Echivalent rezoluția relativă (distanța relativă între doi vecini)

# Rotunjire

- Rotunjirea implicită se face după regula cifrei pare: dacă  $x = d_0.d_1 \dots d_{p-1}d_p \dots$  și  $d_p > \frac{\beta}{2}$  rotunjirea se face în sus, dacă  $d_p < \frac{\beta}{2}$  rotunjirea se face în jos, iar dacă  $d_p = \frac{\beta}{2}$  și printre cifrele eliminate există una nenulă rotunjirea se face în sus, iar în caz contrar ultima cifră păstrată este pară.
- Alte tipuri de rotunjiri: în jos, în sus, spre zero, trunchiere

## Aritmetică în virgulă flotantă

- Definim  $\text{fl}(x)$  ca fiind cea mai apropiată aproximare în virgulă flotantă a lui  $x$
- Din definiția  $\text{eps}$  avem pentru eroarea relativă:  
$$\forall x \in \mathbb{R}, \exists \epsilon \text{ cu } |\epsilon| \leq \text{eps astfel încât } \text{fl}(x) = x(1 + \epsilon)$$
- Rezultatul unei operații  $\odot$  în virgulă flotantă este  $\text{fl}(a \circ b)$
- Dacă  $\text{fl}(a \circ b)$  este cel mai apropiat număr în virgulă flotantă de  $a \circ b$ , operațiile aritmetice se rotunjesc corect (standardul IEEE o face), ceea ce ne conduce la următoarea proprietate:

*Pentru orice numere în virgulă flotantă  $x, y$ , există  $\epsilon$  cu  $|\epsilon| \leq \text{eps}$  astfel încât*

$$x \odot y = (x \circ y)(1 + \epsilon)$$

**numită axioma fundamentală a aritmeticii în virgulă flotantă**

- Rotunjire la cel mai apropiat par în caz de ambiguitate

# Anularea

- Din formulele pentru eroarea relativă (4), dacă  $x \approx x(1 + \delta_x)$  și  $y \approx y(1 + \delta_y)$ , avem următoarele expresii pentru erorile relative ale operațiilor în virgulă flotantă:

$$\delta_{xy} = \delta_x + \delta_y \quad (9)$$

$$\delta_{x/y} = \delta_x - \delta_y \quad (10)$$

$$\delta_{x+y} = \frac{x}{x+y}\delta_x + \frac{y}{x+y}\delta_y \quad (11)$$

- Singura operație critică din punct de vedere al erorii este scăderea a două cantități apropiate  $x \approx y$ , caz în care  $\delta_{x-y} \rightarrow \infty$ .
- Acest fenomen se numește **anulare**
- Figura 1 dă o explicație intuitivă

# Explicarea intuitivă a anulării

$x$	$=$	1	0	1	1	0	0	1	0	1	$b$	$b$	$g$	$g$	$g$	$g$
$y$	$=$	1	0	1	1	0	0	1	0	1	$b'$	$b'$	$g$	$g$	$g$	$g$
$x-y$	$=$	0	0	0	0	0	0	0	0	$b''$	$b''$	$g$	$g$	$g$	$g$	
	$=$	$b''$	$b''$	$g$	$g$	$g$	$g$	?	?	?	?	?	?	?	?	?

Figura: Anularea

## Anularea II

- Anularea este de două tipuri:
  - ① **benignă**, când se scad două cantități exacte
  - ② **catastrofală**, când se scad două cantități deja rotunjite.
- Programatorul trebuie să fie conștient de posibilitatea apariției anulării și să încerce să o evite.
- Expresiile în care apare anularea trebuie rescrise, iar o anulare catastrofală trebuie întotdeauna transformată în una benignă.

## Anularea III

- **Exemplu.** Dacă  $a \approx b$ , atunci expresia  $a^2 - b^2$  se transformă în  $(a - b)(a + b)$ . Forma inițială este de preferat în cazul când  $a \gg b$  sau  $b \gg a$ .
- **Exemplu.** Dacă anularea apare într-o expresie cu radicali, se amplifică cu conjugata:

$$\sqrt{x + \delta} - \sqrt{x} = \frac{\delta}{\sqrt{x + \delta} + \sqrt{x}}, \quad \delta \approx 0.$$

- **Exemplu.** Diferența valorilor unei funcții pentru argumente apropiate se transformă folosind formula lui Taylor:

$$f(x + \delta) - f(x) = \delta f'(x) + \frac{\delta^2}{2} f''(x) + \dots \quad f \in C^n[a, b].$$

## Anularea IV

La ecuația de gradul al doilea  $ax^2 + bx + c = 0$ , anularea poate să apară dacă  $b^2 \gg 4ac$ . Formulele uzuale

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (12)$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (13)$$

pot să conducă la anulare astfel: pentru  $b > 0$  anularea apare la calculul lui  $x_1$ , iar pentru  $b < 0$  anularea apare la calculul lui  $x_2$ . Remediul este să amplificăm cu conjugata

$$x_1 = \frac{2c}{-b - \sqrt{b^2 - 4ac}} \quad (14)$$

$$x_2 = \frac{2c}{-b + \sqrt{b^2 - 4ac}} \quad (15)$$

și să utilizăm în primul caz formulele (14) și (13), iar în al doilea caz (12) și (15). [.../demo/html/ecgr2.html](http://demo/html/ecgr2.html)

# Teorema asupra pierderii preciziei

- Problemă: Câte cifre semnificative se pierd la scăderea  $x - y$  când  $x$  este apropiat de  $y$ ?
- Apropierea lui  $x$  de  $y$  este măsurată convenabil de  $1 - \frac{y}{x}$ .

## Teoremă (Loss of precision theorem)

Fie  $x$  și  $y$  NVF normalize, unde  $x > y > 0$ . Dacă

$$2^{-p} \leq 1 - \frac{y}{x} \leq 2^{-q}$$

pentru  $p, q \in \mathbb{N}$ , atunci se pierd cel puțin  $q$  și cel mult  $p$  cifre binare semnificative la scăderea  $x - y$ .

# Teorema asupra pierderii preciziei - demonstrație

## Demonstrație.

Vom demonstra partea a doua, lăsând prima parte ca exercițiu. Fie  $x = s \times 2^n$ ,  $y = r \times 2^m$  NVF normalize. Deoarece  $y < x$ ,  $y$  va trebui deplasat înaintea scăderii, pentru a avea același exponent ca  $x$ . Deci,  $y = (s2^{m-n}) \times 2^n$  și

$$x - y = (r - s2^{m-n}) \times 2^n$$

Semnificantul satisfacă

$$r - s2^{m-n} = r \left(1 - \frac{r \times 2^m}{s \times 2^n}\right) = r \left(1 - \frac{y}{x}\right) < 2^{-q}$$

Deci, pentru normalizarea reprezentării lui  $x - y$ , este nevoie de o deplasare de  $q$  biți la stânga. Astfel se introduc cel puțin  $q$  zerouri false la capătul drept al semnificantului. Aceasta înseamnă o pierdere a preciziei de cel puțin  $q$  biți.



# Reducerea rangului I

## Exemplu

Pentru  $\sin x$ , câți biți semnificativi se pierd la reducerea la intervalul  $[0, 2\pi)$ ?

**Soluție.** Dându-se  $x > 2\pi$ , vom determina întregul  $n$  ce satisface  $0 \leq x - 2n\pi < 2\pi$ . Apoi la evaluare vom utiliza periodicitatea  $f(x) = f(x - 2n\pi)$ . La scăderea  $x - 2n\pi$ , va fi o pierdere de precizie. Conform teoremei 1 se vor pierde cel puțin  $q$  biți dacă

$$1 - \frac{2n\pi}{x} \leq 2^{-q}$$

Deoarece

$$1 - \frac{2n\pi}{x} = \frac{x - 2n\pi}{x} < \frac{2\pi}{x}$$

## Reducerea rangului II

conchidem că cel puțin  $q$  biți se pierd dacă  $2\pi/x < 2^{-q}$ , sau echivalent, dacă  $2^q < x/2\pi$ . ■

**Exemplu numeric.** Să se calculeze  $\sin(12532.14)$ .

Avem  $\sin(12532.14) = \sin(12532.14 - 2k\pi)$ , cu  $k = 1994$  și  $12532.14 - 2k\pi \approx 3.47$  și rezultatul va fi eronat. Dacă reducerea s-ar fi putut face cu precizie mai bună și rezultatul ar fi fost mai bun. MATLAB dă  $\sin(12532.14) = -0.321113319309938$  și  $\sin(3.47) = -0.322535900322479$ .

# Standardele IEEE

- Există două standarde diferite pentru calculul în virgulă flotantă:
  - IEEE 754 care prevede  $\beta = 2$
  - IEEE 854 care permite  $\beta = 2$  sau  $\beta = 10$ , dar lasă o mai mare libertate de reprezentare.
- Parametrii standardului IEEE 754

	Precizia			
	Simplă	Simplă extinsă	Dublă	Dublă extinsă
p	24	$\geq 32$	53	$\geq 64$
$e_{\max}$	+127	$\geq +1023$	+1023	$\geq +16383$
$e_{\min}$	-126	$\leq -1022$	-1022	$\leq -16382$
dim. exponent	8	$\geq 11$	11	$\geq 15$
dim. număr	32	$\geq 43$	64	$\geq 79$

Tabela: Parametrii reprezentării flotante

bit ascuns -  $d_0 = 1$ , deci nu trebuie reprezentat fizic

Motivele pentru formatele extinse sunt:

- ① o mai bună precizie;
- ② pentru conversia din binar în zecimal și invers este nevoie de 9 cifre în simplă precizie și de 17 cifre în dublă precizie.

Motivul pentru care  $|e_{min}| < e_{max}$  este acela că  $1/2^{e_{min}}$  nu trebuie să dea depășire.

Operațiile  $\oplus, \ominus, \otimes, \oslash$  trebuie să fie **exact rotunjite**. Precizia aceasta se asigură cu două cifre de gardă și un bit suplimentar.

Reprezentarea exponentului se numește **reprezentare cu exponent deplasat**, adică în loc de  $e$  se reprezintă  $e + D$ , unde  $D$  este fixat la alegerea reprezentării.

$D = 127$  pentru simplă precizie și  $D = 1023$  pentru dublă precizie.

# Precizia cvadruplă

După IEEE 754-2008

- $p = 113$  biți (112+1 bit ascuns);
- dim. exponent=15 biți
- $e_{max} = 16383$ ,  $e_{min} = -16382$
- deplasamentul  $D = 16383$
- dim. număr 128

# Cantități speciale

Exponent	Semnificant	Ce reprezintă
$e = e_{min} - 1$	$f = 0$	$\pm 0$ zero cu semn
$e = e_{min} - 1$	$f \neq 0$	$0.f \times 2^{e_{min}}$ Numere denormalizate
$e_{min} \leq e \leq e_{max}$		$1.f \times 2^e$
$e = e_{max} + 1$	$f = 0$	$\pm\infty$ infinit
$e = e_{max} + 1$	$f \neq 0$	NaN NaN-uri

## Cantități speciale

**NaN.** Avem de fapt o familie de valori NaN, operațiile ilegale sau nedeterminate conduc la NaN:  $\infty + (-\infty)$ ,  $0 \times \infty$ ,  $0/0$ ,  $\infty/\infty$ ,  $x \text{ REM } 0$ ,  $\infty \text{ REM } y$ ,  $\sqrt{x}$  pentru  $x < 0$ . Dacă un operand este NaN rezultatul va fi tot NaN.

**Infinit.** Operațiile cu  $\infty$  se definesc ca limite, ex:  $1/0 = \infty$ ,  $-1/0 = -\infty$ . Valorile infinite dau posibilitatea continuării calculului, lucru mai sigur decât abortarea sau returnarea celui mai mare număr reprezentabil.  $\frac{x}{1+x^2}$  pentru  $x = \infty$  dă rezultatul 0.

**Zero cu semn.** Avem doi de 0:  $+0, -0$ ; relațiile  $+0 = -0$  și  $-0 < +\infty$  sunt adevărate. Avantaje: tratarea simplă a depășirilor inferioare și discontinuităților. Se face distincție între  $\log 0 = -\infty$  și  $\log x = \text{NaN}$  pentru  $x < 0$ . Fără 0 cu semn nu s-ar putea face distincție la logaritm între un număr negativ care dă depășire superioară și 0.

## IEEE Simplă precizie, exemple

$s$	$e + D$	$f$	Cantitate
0	11111111	000001000000000000000000	NaN
1	11111111	00100010000100101010101	NaN
0	11111111	000000000000000000000000	$\infty$
0	10000001	101000000000000000000000	$+2^{129-127} \cdot 1.101 = 6.5$
0	10000000	000000000000000000000000	$+2^{128-127} \cdot 1.0 = 2$
0	00000001	000000000000000000000000	$+2^{1-127} \cdot 1.0 = 2^{-126}$
0	00000000	100000000000000000000000	$+2^{-126} \cdot 0.1 = 2^{-127}$
0	00000000	000000000000000000000001	$+2^{-126} \cdot 2^{-23} = 2^{-149}$
0	00000000	000000000000000000000000	$+0$
1	00000000	000000000000000000000000	$-0$
1	10000001	101000000000000000000000	$-2^{129-127} \cdot 1.101 = -6.5$
1	11111111	000000000000000000000000	$-\infty$

Pentru virgulă flotantă în MATLAB vezi [.../demo/html/fpdemo.html](http://.../demo/html/fpdemo.html)



**William Kahan**, eminent matematician și informatician, contribuții importante la studiul metodelor precise și eficiente de rezolvare a problemelor numerice pe calculatoare cu precizie finită. A fost principalul arhitect al standardului IEEE 754. Distins cu premiul Turing al ACM în 1989, Fellow al ACM din 1994. Profesor la Universitatea Berkeley, California

# Eșecul rachetei Patriot I



- Eșecul unui sistem de rachete antirachetă Patriot în timpul războiului din Golf din 1991 s-a datorat unei erori de conversie software.
- Ceasul sistemului măsura timpul în zecimi de secundă, dar îl memora într-un registru de 24 de biți, provocându-se astfel erori de rotunjire.
- Datele din câmp au arătat că sistemul poate eșua să urmărească și să intercepteze o rachetă după 20 de ore de funcționare și deci sistemul ar necesita rebootare.

## Eșecul rachetei Patriot II

- După 100 de ore de funcționare, eșecul sistemului a cauzat moartea a 28 de soldați americani aflați într-o cazarmă din Dhahran, Arabia Saudită, deoarece nu a reușit să intercepteze o rachetă Scud irakiană. Deoarece numărul 0.1 are o dezvoltare infinită în binar (este o fracție periodică), valoarea din registrul de 24 de biți este eronată

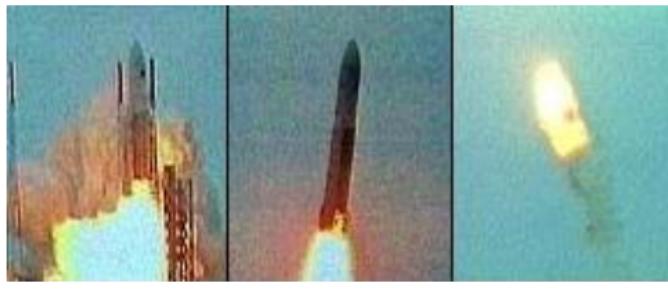
$$(0.00011001100110011001100)_2 \approx 0.95 \times 10^{-7}.$$

Eroarea de timp după o sută de ore a fost de 0.34 secunde. Viteza rachetei Scud este de 3750 mile/oră, rezultând o eroare în distanță de aproximativ 573.59 m.

Vezi [../demo/html/patriotmaple.html](#) și  
[../demo/html/patriot.html](#)

# Explozia rachetei Ariane 5

- În 1996, racheta Ariane 5 lansată de Agenția Spațială Europeană a explodat la 40 de secunde după lansarea de la Kourou, Guyana Franceză.
- Investigația de după incident a arătat că componenta orizontală a vitezei a necesitat conversia unui număr flotant în dublă precizie într-un întreg pe 16 biți.
- Deoarece numărul era mai mare decât 32,767, cel mai mare întreg reprezentabil pe 16 biți, componentele de control au intrat în procedura de autodistrugere. Valoarea rachetei și a încărcăturii a fost de 500 de milioane de dolari.



## Referințe WWW

Se pot găsi informații adiționale pe World Wide Web la adresa  
<http://www.ima.umn.edu/~arnold/disasters/> sau la  
<http://www5.in.tum.de/~huckle/bugse.html>. Există și alte  
consemnări ale calamităților ce ar fi putut fi evitate printr-o programare  
mai atentă, în special la utilizarea AVF.

# Condiționarea unei probleme

- Putem gândi o problemă ca o aplicație

$$f : \mathbb{R}^m \rightarrow \mathbb{R}^n, \quad y = f(x). \quad (16)$$

- Ne interesează sensibilitatea aplicației într-un punct dat  $x$  la mici perturbații ale argumentului, adică cât de mare sau cât de mică este perturbația lui  $y$  comparativ cu perturbația lui  $x$ .
- În particular, dorim să măsurăm gradul de sensibilitate printr-un singur număr, numărul de condiționare al aplicației  $f$  în punctul  $x$ . Vom presupune că  $f$  este calculată exact, cu precizie infinită.
- **Condiționarea lui  $f$  este deci o proprietate inherentă a funcției  $f$  și nu depinde de nici o considerație algoritmică legată de implementarea sa.**

## Condiționarea unei probleme

- Aceasta nu înseamnă că determinarea condiționării unei probleme este nerelevantă pentru orice soluție algoritmică a problemei.
- Soluția calculată cu (16),  $y^*$  (utilizând un algoritm specific și aritmetică în virgulă flotantă) este (și acest lucru se poate demonstra) soluția unei probleme „apropiate”

$$y^* = f(x^*) \quad (17)$$

cu

$$x^* = x + \delta \quad (18)$$

- distanța  $\|\delta\| = \|x^* - x\|$  poate fi estimată în termeni de precizie a mașinii
- dacă știm cât de tare sau cât de slab reacționează aplicația la mici perturbații, cum ar fi  $\delta$  în (18), putem spune ceva despre eroarea  $y^* - y$  a soluției cauzată de această perturbație.

## Condiționarea unei probleme

Fie  $x = [x_1, \dots, x_m]^T \in \mathbb{R}^m$ ,  $y = [y_1, \dots, y_n]^T \in \mathbb{R}^n$ ,  $y_\nu = f_\nu(x_1, \dots, x_m)$ ,  $\nu = \overline{1, n}$  —  $y_\nu$  va fi privit ca o funcție de o singură variabilă  $x_\mu$

$$\gamma_{\nu\mu} = (\text{cond}_{\nu\mu} f)(x) = \left| \frac{x_\mu \frac{\partial f_\nu}{\partial x_\mu}}{f_\nu(x)} \right|. \quad (19)$$

Aceasta ne dă o matrice de numere de condiționare (vezi și (4))

$$\Gamma(x) = \begin{pmatrix} \frac{x_1 \frac{\partial f_1}{\partial x_1}}{f_1(x)} & \cdots & \frac{x_m \frac{\partial f_1}{\partial x_m}}{f_1(x)} \\ \vdots & \ddots & \vdots \\ \frac{x_1 \frac{\partial f_n}{\partial x_1}}{f_n(x)} & \cdots & \frac{x_m \frac{\partial f_n}{\partial x_m}}{f_n(x)} \end{pmatrix} =: [\gamma_{\nu\mu}(x)] \quad (20)$$

și vom lua ca **număr de condiționare**

$$(\text{cond } f)(x) = \|\Gamma(x)\|. \quad (21)$$

# Condiționarea unei probleme

Altfel.

$$\|\Delta y\| = \|f(x + \Delta x) - f(x)\| \leq \|\Delta x\| \left\| \frac{\partial f}{\partial x} \right\|$$

unde

$$J(x) = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^n \times \mathbb{R}^m \quad (22)$$

este matricea jacobiană a lui  $f$

$$\frac{\|\Delta y\|_\infty}{\|y\|_\infty} \leq \frac{\|x\|_\infty \left\| \frac{\partial f}{\partial x} \right\|_\infty}{\|f(x)\|_\infty} \cdot \frac{\|\Delta x\|}{\|x\|_\infty}. \quad (23)$$

## Cazul unidimensional

- Pentru  $m = n = 1$  și  $x \neq 0, y \neq 0$

$$(\text{cond } f)(x) = \left| \frac{xf'(x)}{f(x)} \right|.$$

- Dacă  $x = 0 \wedge y \neq 0$  se consideră eroarea absolută pentru  $x$  și eroarea relativă pentru  $y$

$$(\text{cond } f)(x) = \left| \frac{f'(x)}{f(x)} \right|;$$

- Pentru  $y = 0 \wedge x \neq 0$  se ia eroarea absolută pentru  $y$  și eroarea relativă pentru  $x$
- Pentru  $x = y = 0$ , se iau erorile absolute

$$(\text{cond } f)(x) = f'(x).$$

# Condiționarea absolută

- Număr de condiționare absolută al unei probleme diferențiabile  $f$  în  $x$ :

$$\hat{\kappa} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|} = \|J(x)\|$$

unde  $J(x) = [J_{ij}] = [\partial f_i / \partial x_j]$ , este jacobianul, iar norma este indușă de normele lui  $\delta f$  și  $\delta x$

- în cazul unidimensional

$$\hat{\kappa} = |f'(x)|.$$

# Exemple

- **Exemplu:** Funcția  $f(x) = \alpha x$

- număr de condiționare absolută  $\hat{\kappa} = \|J\| = \alpha$
- număr de condiționare relativă  $(\text{cond } f)(x) = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{\alpha}{\alpha x/x} = 1$

- **Exemplu:** Funcția  $f(x) = \sqrt{x}$

- număr de condiționare absolută  $\hat{\kappa} = \|J\| = \frac{1}{2\sqrt{x}}$
- număr de condiționare relativă  
 $(\text{cond } f)(x) = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{1/(2\sqrt{x})}{\sqrt{x}/x} = \frac{1}{2}$

- **Exemplu:** Funcția  $f(x) = x_1 - x_2$  (cu norma  $\infty$ )

- număr de condiționare absolută  $\hat{\kappa} = \|J\| = \|(-1, 1)^T\| = 2$
- număr de condiționare relativă

$$(\text{cond } f)(x) = \frac{\|J\|}{\|f(x)\| / \|x\|} = \frac{2}{|x_1 - x_2| / \max\{|x_1|, |x_2|\}}$$

- prost condiționată dacă  $x_1 \approx x_2$  (anulare)

# Precizia

- Pentru o funcție dată  $g(n)$  vom nota cu  $\Theta(g(n))$  mulțimea de funcții
$$\Theta(g(n)) = \{f(n) : \exists c_1, c_2, n_0 > 0 \quad 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \\ \forall n \geq n_0\}.$$
- Scriem  $f(n) = \Theta(g(n))$  pentru a indica  $f(n) \in \Theta(g(n))$ . Spunem că  $g(n)$  este o *margine asimptotică strânsă* (*assymptotically tight bound*) pentru  $f(n)$ .
- Definiția mulțimii  $\Theta(g(n))$  necesită ca fiecare membru al ei să fie *asimptotic nenegativ*, adică  $f(n) \geq 0$  când  $n$  este suficient de mare.

- Pentru o funcție dată  $g(n)$  vom nota cu  $O(g(n))$  mulțimea de funcții

$$O(g(n)) = \{f(n) : \exists c, n_0 \quad 0 \leq f(n) \leq cg(n), \quad \forall n \geq n_0\}.$$

- *margine asymptotică superioară*
- Pentru a indica faptul că  $f(n)$  este un membru al lui  $O(g(n))$  scriem  $f(n) = O(g(n))$ .
- Observăm că  $f(n) = \Theta(g(n)) \implies f(n) = O(g(n))$ , sau  $\Theta(g(n)) \subseteq O(g(n))$
- Una dintre proprietățile ciudate ale notației este aceea că  $n = O(n^2)$ .

- Pentru o funcție dată  $g(n)$  vom nota prin  $\Omega(g(n))$  mulțimea de funcții

$$\Omega(g(n)) = \{f(n) : \exists c, n_0 \quad 0 \leq cg(n) \leq f(n), \quad \forall n \geq n_0\}.$$

- *margine asimptotică inferioară*
- Din definițiile notațiilor asimptotice se obține imediat:

$$f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \wedge f(n) = \Omega(g(n)).$$

- Spunem că funcțiile  $f$  și  $g : \mathbb{N} \rightarrow \mathbb{R}$  sunt *asimptotic echivalente*, notație  $\sim$  dacă

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1.$$

- Extinderea notațiilor asimptotice la mulțimea numerelor reale este naturală. De exemplu  $f(t) = O(g(t))$  înseamnă că există o constantă pozitivă  $C$  astfel încât pentru orice  $t$  suficient de apropiat de o limită subînțeleasă (de exemplu  $t \rightarrow \infty$  sau  $t \rightarrow 0$ ) avem

$$|f(t)| \leq Cg(t). \tag{24}$$

# Precizia

- Considerăm un *algoritm*  $\tilde{f}$  pentru *problema*  $f$
- Un calcul  $\tilde{f}(x)$  are *eroarea absolută*  $\|\tilde{f}(x) - f(x)\|$  și *eroarea relativă*

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}$$

- Algoritmul este **precis** dacă (pentru orice  $x$ )

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\text{eps})$$

unde  $O(\text{eps})$  este “de ordinul  $\text{eps}$ ” (vezi slide-ul următor)

- Constanta din  $O(\text{eps})$  poate fi foarte mare pentru multe probleme, căci datorită erorilor de rotunjire nu utilizăm nici chiar un  $x$  corect.

## Detalii asupra notațiilor asimptotice

- Notația  $\varphi(t) = O(\psi(t))$  înseamnă că există o constantă  $C$  a.î. pentru  $t$  apropiat de o limită (de obicei 0 sau  $\infty$ ),  $|\varphi(t)| \leq C\psi(t)$
- **Exemplu:**  $\sin^2 t = O(t^2)$  când  $t \rightarrow 0$  înseamnă  $|\sin^2 t| \leq Ct^2$  pentru un anumit  $C$
- Dacă  $\varphi$  depinde de variabile adiționale, notația

$$\varphi(s, t) = O(\psi(t)) \quad \text{uniform în } s$$

înseamnă că există o constantă  $C$  a.î.  $|\varphi(s, t)| \leq C\psi(t)$  pentru orice  $s$

- **Exemplu:**  $(\sin^2 t)(\sin^2 s) = O(t^2)$  uniform când  $t \rightarrow 0$ , dar nu dacă  $\sin^2 s$  este înlocuit cu  $s^2$
- În margini de forma  $\|\tilde{x} - x\| \leq C\kappa(A)\text{eps} \|x\|$ ,  $C$  nu depinde de  $A$  sau  $b$ , dar poate depinde de dimensiunea  $m$

# Stabilitatea

- Un algoritm  $\tilde{f}$  pentru problema  $f$  este stabil dacă pentru orice  $x$  există un  $\tilde{x}$  cu proprietatea

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\text{eps})$$

a.î.

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\text{eps})$$

# Stabilitatea

- Un algoritm  $\tilde{f}$  pentru problema  $f$  este **stabil** dacă pentru orice  $x$  există un  $\tilde{x}$  cu proprietatea

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\text{eps})$$

a.î.

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\text{eps})$$

- “Răspuns aproape corect la problemă aproape exactă”

# Stabilitatea

- Un algoritm  $\tilde{f}$  pentru problema  $f$  este **stabil** dacă pentru orice  $x$  există un  $\tilde{x}$  cu proprietatea

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\text{eps})$$

a.i.

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\text{eps})$$

- “Răspuns aproape corect la problemă aproape exactă”
- Un algoritm  $\tilde{f}$  pentru problema  $f$  este **regresiv stabil** dacă pentru orice  $x$  există un  $\tilde{x}$  cu proprietatea

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\text{eps})$$

a.i.

$$\tilde{f}(x) = f(\tilde{x}).$$

# Stabilitatea

- Un algoritm  $\tilde{f}$  pentru problema  $f$  este **stabil** dacă pentru orice  $x$  există un  $\tilde{x}$  cu proprietatea

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\text{eps})$$

a.î.

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\text{eps})$$

- “Răspuns aproape corect la problemă aproape exactă”
- Un algoritm  $\tilde{f}$  pentru problema  $f$  este **regresiv stabil** dacă pentru orice  $x$  există un  $\tilde{x}$  cu proprietatea

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\text{eps})$$

a.î.

$$\tilde{f}(x) = f(\tilde{x}).$$

- “Răspuns corect la problemă aproape exactă”

# Stabilitatea AVF

- Cele două axiome ale AVF implică stabilitatea regresivă a operației ⊖
  - (1)  $\forall x \in \mathbb{R}, \exists \epsilon \text{ cu } |\epsilon| \leq \text{eps a.i. } \text{fl}(x) = x(1 + \epsilon)$
  - (2) Pentru orice NVF  $x, y$ , există  $\epsilon$  cu  $|\epsilon| \leq \text{eps a.i.}$   
 $x \odot y = (x \circ y)(1 + \epsilon)$
- **Exemplu:** Scăderea  $f(x_1, x_2) = x_1 - x_2$  cu algoritmul

$$\tilde{f}(x_1, x_2) = \text{fl}(x_1) \ominus \text{fl}(x_2)$$

- (1) implică existența  $|\epsilon_1|, |\epsilon_2| \leq \text{eps a.i.}$

$$\text{fl}(x_1) = x_1(1 + \epsilon_1), \quad \text{fl}(x_2) = x_2(1 + \epsilon_2)$$

## Stabilitatea AVF II

(continuarea exemplului)

- (2) implică existența  $|\epsilon_3| \leq \text{eps}$  a.î.

$$\text{fl}(x_1) \ominus \text{fl}(x_2) = (\text{fl}(x_1) - \text{fl}(x_2))(1 + \epsilon_3)$$

- Combinând, rezultă existența  $|\epsilon_4|, |\epsilon_4| \leq 2\text{eps} + O(\text{eps}^2)$  a.î.

$$\begin{aligned}\text{fl}(x_1) \ominus \text{fl}(x_2) &= (x_1(1 + \epsilon_1) - x_2(1 + \epsilon_2))(1 + \epsilon_3) \\ &= x_1(1 + \epsilon_1)(1 + \epsilon_3) - x_2(1 + \epsilon_2)(1 + \epsilon_3) \\ &= x_1(1 + \epsilon_4) - x_2(1 + \epsilon_5)\end{aligned}$$

- Deci,  $\text{fl}(x_1) - \text{fl}(x_2) = \tilde{x}_1 - \tilde{x}_2$

# Stabilitatea AVF III

- **Exemplu:** Produsul  $f(x, y) = x^*y$  calculat cu  $\otimes$  și  $\oplus$  este regresiv stabil

# Stabilitatea AVF III

- **Exemplu:** Produsul  $f(x, y) = x^*y$  calculat cu  $\otimes$  și  $\oplus$  este regresiv stabil
- **Exemplu:** Produsul exterior  $f(x, y) = xy^*$  calculat cu  $\otimes$  nu este regresiv stabil (în afară de cazul când  $\tilde{f}$  are rangul 1)

# Stabilitatea AVF III

- **Exemplu:** Produsul  $f(x, y) = x^*y$  calculat cu  $\otimes$  și  $\oplus$  este regresiv stabil
- **Exemplu:** Produsul exterior  $f(x, y) = xy^*$  calculat cu  $\otimes$  nu este regresiv stabil (în afară de cazul când  $\tilde{f}$  are rangul 1)
- **Exemplu:**  $f(x) = x + 1$  calculat cu  $\tilde{f}(x) = \text{fl}(x) \oplus 1$  nu este regresiv stabil (considerăm  $x \approx 0$ )

# Stabilitatea AVF III

- **Exemplu:** Produsul  $f(x, y) = x^*y$  calculat cu  $\otimes$  și  $\oplus$  este regresiv stabil
- **Exemplu:** Produsul exterior  $f(x, y) = xy^*$  calculat cu  $\otimes$  nu este regresiv stabil (în afară de cazul când  $\tilde{f}$  are rangul 1)
- **Exemplu:**  $f(x) = x + 1$  calculat cu  $\tilde{f}(x) = \text{fl}(x) \oplus 1$  nu este regresiv stabil (considerăm  $x \approx 0$ )
- **Exemplu:**  $f(x, y) = x + y$  calculat cu  $\tilde{f}(x, y) = \text{fl}(x) \oplus \text{fl}(y)$  este regresiv stabil

## Teoremă (Precizia unui algoritm regresiv stabil)

Dacă se utilizează un algoritm regresiv stabil pentru a rezolva problema  $f$  cu numărul de condiționare  $\text{cond}(f)(x)$ , eroarea relativă satisfacă

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O((\text{cond } f)(x)\text{eps})$$

### Demonstrație.

Stabilitatea regresivă înseamnă  $\tilde{f}(x) = f(\tilde{x})$ , pentru un anumit  $\tilde{x}$  a. î.  
 $\frac{\|\tilde{x} - x\|}{\|x\|} = O(\text{eps})$ . Definiția numărului de condiționare ne dă

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = ((\text{cond } f)(x) + o(1)) \frac{\|\tilde{x} - x\|}{\|x\|}$$

unde  $o(1) \rightarrow 0$  la fel ca  $\text{eps} \rightarrow 0$ . Combinând aceste două se obține rezultatul dorit.



# Bibliografie I

-  James Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
-  W. Gautschi, *Numerical Analysis. An Introduction*, Birkhäuser, Basel, 1997.
-  D. Goldberg, *What every computer scientist should know about floating-point arithmetic*, Computing Surveys **23** (1991), no. 1, 5–48.
-  Nicholas J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
-  M. L. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*, SIAM, Philadelphia, 2001.
-  J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, 2nd ed., Springer Verlag, 1992.

## Bibliografie II

- ☞ C. Überhuber, *Computer-Numerik*, vol. 1, 2, Springer Verlag, Berlin, Heidelberg, New-York, 1995.
- ☞ C. Ueberhuber, *Numerical Computation. Methods, Software and Analysis*, vol. I, II, Springer Verlag, Berlin, Heidelberg, New York, 1997.

# Metode directe pentru sisteme de ecuații liniare

## Eliminare gaussiană, descompunere LU, Cholesky

Radu T. Trîmbițaș

Universitatea "Babeș-Bolyai"

6 aprilie 2020

# Rezolvarea sistemelor liniare

- În notație matricială, un sistem se scrie sub forma

$$Ax = b$$

unde  $A \in \mathbb{K}^{n \times n}$ ,  $b \in \mathbb{K}^{n \times 1}$ .

- Soluția  $x = A^{-1}b$
- În majoritatea problemelor practice inversarea este nenecesară și nerecomandabilă
- Exemplu extrem,  $n = 1$ :  $7x = 21$ , cu soluția  $x = \frac{21}{7} = 3$ , o operație /
- Rezolvat prin inversare ne dă  
 $x = 7^{-1} \cdot 21 = 0.1428571 \cdot 21 = 3.0000002$ , două operații /\*
- Considerații similare se aplică și la sisteme cu mai multe ecuații și chiar la sisteme cu aceeași matrice  $A$  și membri drepti diferiți

# Un exemplu numeric I

- Considerăm exemplul

$$\begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \\ 6 \end{bmatrix}$$

- $0.3E_1 + E_2 \rightarrow E_2$ ,  $-0.5E_1 + E_3 \rightarrow E_3$ . Coeficientul 10 al lui  $x_1$  se numește *pivot*, iar cantitățile -0.3 și 0.5 obținute prin împărțirea coeficienților lui  $x_1$  din celelalte ecuații se numesc *multiplicatori*

$$\begin{bmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 2.5 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 6.1 \\ 2.5 \end{bmatrix}$$

## Un exemplu numeric II

- Pasul 2, eliminarea lui  $x_2$  din a treia ecuație. Pivotul, coeficientul lui  $x_2$ ,  $-0.1$  este mai mic decât alți coeficienți. Din acest motiv ecuațiile 2 și 3 trebuie interschimbate, operație numită *pivotare*. În acest caz nu este necesară, dar în general este crucială.

$$\begin{bmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & -0.1 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 2.5 \\ 6.1 \end{bmatrix}$$

- Pasul 3,  $0.04E_2 + E_3 \rightarrow E_3$  (Cât ar fi multiplicatorul fără interschimbare?)

$$\begin{bmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 2.5 \\ 6.2 \end{bmatrix}$$

## Un exemplu numeric III

- Ultima ecuație

$$6.2x_3 = 6.2$$

ne dă  $x_3 = 1$ .

- Înlocuind în a doua ecuație

$$2.5x_2 + 5 \cdot 1 = 2.5 \implies x_2 = -1.$$

- Înlocuind în prima ecuație

$$10x_1 + (-7)(-1) = 7 \implies x_1 = 0.$$

- Soluția este

$$x = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

# Un exemplu numeric IV

- Verificare

$$\begin{bmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \\ 6 \end{bmatrix}.$$

- Algoritmul poate fi exprimat mai compact în formă matricială, fie

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & -0.04 & 1 \end{bmatrix}, U = \begin{bmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{bmatrix}, P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Matricea  $L$  este matricea multiplicatorilor,  $U$  este matricea finală,  $P$  descrie pivotarea și

$$LU = PA$$

- Matricea originală poate fi exprimată ca produs de matrice cu o structură mai simplă.

# Factorizare LU

- Algoritmul folosit aproape universal pentru rezolvarea sistemelor liniare este eliminarea gaussiană.
- Cercetările din perioada 1955-1965 au evidențiat aspecte ale EG neînțelese până atunci: alegerea pivotilor și interpretarea corectă a efectului erorilor de rotunjire
- EG are două stadii, triunghiularizarea matricei inițiale și substituția inversă

# Matrice de permutare și triunghiulare I

- O matrice de permutare se obține din matricea identică prin permutări de linii sau coloane. O astfel de matrice are exact un 1 pe fiecare linie și coloană și în rest 0.

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

- Efect:  $PA$  permutare de linii,  $AP$  permutare de coloane
- MATLAB utilizează și vectori de permutare ca indici de linie sau coloane; fie  $p = [4 \ 1 \ 3 \ 2]$ ,  $P*A$  și  $A(p, :)$  sunt echivalente, la fel  $A*P$  și  $A(:, p)$ . Notația vectorială este mai rapidă și utilizează mai puțină memorie.

# Matrice de permutare și triunghiulare II

- Soluția sistemului  $Px = b$ ,  $P$  matrice de permutare, este  $x = P^T b$ , adică o rearanjare a componentelor lui  $b$ .
- O matrice triunghiulară superior are toate elementele nenule deasupra diagonalei principale sau pe ea, adică  $a_{ij} = 0$  dacă  $i > j$ .
- Analog se definesc și matricele triunghiulare inferior. La rezolvarea sistemelor liniare sunt importante și matricele triunghiulare inferior care au toate elementele de pe diagonala principală egale cu 1 *unit lower triangular matrices*.
- Sistemele liniare cu matricea triunghiulară sunt ușor de rezolvat în timp  $\Theta(m^2)$ .
- **Măsura complexității -Numărul de operații în virgulă flotantă.**

# Algoritm pentru sistem triunghiular superior

```
function x = backsubst(U,b)
%BACKSUBST - backward substitution
%U - upper triangular matrix
%b - right hand side vector

n=length(b);
x=zeros(size(b));
for k=n:-1:1
    x(k)=(b(k)-U(k,k+1:n)*x(k+1:n))/U(k,k);
end
```

# Algoritm pentru sistem triunghiular inferior

```
function x=forwardsubst(L,b)
%FORWARDSUBST - forward substitution
%L - lower triangular matrix
%b - right hand side vector

x=zeros(size(b));
n=length(b);
for k=1:n
    x(k)=(b(k)-L(k,1:k-1)*x(1:k-1))/L(k,k);
end
```

# Descompunerea LU

- Transformă  $A \in \mathbb{C}^{m \times m}$  într-o matrice triunghiulară superior,  $U$  scăzând multiplii de linii
- Fiecare  $L_i$  introduce zerouri sub diagonală în coloana  $i$ :

$$\underbrace{L_{m-1} \dots L_2 L_1}_{{L^{-1}}} A = U \implies A = LU \text{ unde } L = L_1^{-1} L_2^{-1} \dots L_{m-1}^{-1}$$

$$\begin{array}{c}
 \left[ \begin{array}{cccc} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{array} \right] \xrightarrow{L_1} \left[ \begin{array}{cccc} \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \end{array} \right] \xrightarrow{L_2} \left[ \begin{array}{cccc} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \end{array} \right] \xrightarrow{L_3} \left[ \begin{array}{cccc} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times \end{array} \right] \\
 A \qquad \qquad \qquad L_1 A \qquad \qquad \qquad L_2 L_1 A \qquad \qquad \qquad L_3 L_2 L_1 A
 \end{array}$$

- “Triunghiularizare triunghiulară”

# Matricele $L_k$

- La pasul  $k$  se elimină elementele de sub  $A_{kk}$ :

$$x_k = [ x_{11} \quad \cdots \quad x_{kk} \quad x_{k+1,k} \quad \cdots \quad x_{mk} ]^T$$

$$L_k x_k = [ x_{11} \quad \cdots \quad x_{kk} \quad 0 \quad \cdots \quad 0 ]^T$$

- Multiplicatorii  $\ell_{jk} = x_{jk} / x_{kk}$  apar în  $L_k$ :

$$L_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\ell_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -\ell_{mk} & & 1 \end{bmatrix}$$

# Construcția lui $L$

- Matricea  $L$  conține toți multiplicatorii într-o singură matrice (cu semne +)

$$L = L_1^{-1} L_2^{-1} \dots L_{m-1}^{-1} = \begin{bmatrix} 1 & & & & & \\ \ell_{21} & 1 & & & & \\ \ell_{31} & \ell_{32} & 1 & & & \\ \vdots & \vdots & \ddots & \ddots & & \\ \ell_{m1} & \ell_{m2} & \cdots & \ell_{m,m-1} & 1 \end{bmatrix}$$

- Definim  $\ell_k = [0, \dots, 0, \ell_{k+1,k}, \dots, \ell_{m,k}]^T$ . Atunci  $L_k = I - \ell_k e_k^*$ .
  - Avem  $L_k^{-1} = I + \ell_k e_k^*$ , deoarece  $e_k^* \ell_k = 0$  și  $(I - \ell_k e_k^*) (I + \ell_k e_k^*) = I - \ell_k e_k^* \ell_k e_k^* = I$
  - De asemenea,  $L_k^{-1} L_{k+1}^{-1} = I + \ell_k e_k^* + \ell_{k+1} e_{k+1}^*$ , deoarece  $e_k^* \ell_{k+1} = 0$  și  $(I + \ell_k e_k^*) (I + \ell_{k+1} e_{k+1}^*) = I + \ell_k e_k^* + \ell_{k+1} e_{k+1}^* + \ell_k e_k^* \ell_{k+1} e_{k+1}^*$

# Eliminare gaussiană fără pivotare

- Se factorizează  $A \in \mathbb{C}^{m \times m}$  în  $A = LU$

## Eliminare gaussiană fără pivot

```

 $U := A; L = I;$ 
for  $k := 1$  to  $m - 1$  do
    for  $j := k + 1$  to  $m$  do
         $\ell_{jk} := u_{jk} / u_{kk};$ 
         $u_{j,k:m} := u_{j,k:m} - \ell_{jk} u_{k,k:m};$ 

```

- Ciclul interior poate fi scris utilizând operații matriciale în loc de cicluri **for**
- Număr de operații (complexitatea)
  $\sim \sum_{k=1}^m 2(m-k)(m-k) \sim 2 \sum_{k=1}^m k^2 \sim \frac{2}{3}m^3$

# Eliminare gaussiană cu produs exterior

- Ciclul interior poate fi scris cu operații matriciale în loc de for

## Eliminare gaussiană cu produs exterior

```
for k := 1 to m - 1 do
    rows := k + 1 : m;
    Arows,k := Arows,k / Ak,k;
    Arows,rows := Arows,rows - Arows,k Ak,rows;
```

Matricea  $S = A_{k+1:m,k+1:m} - A_{k+1:m,k} A_{k,k+1:m}$  se numește **complement Schur** al lui  $A$  în raport cu  $a_{k,k}$ .

# Necesitatea pivotării I

- EG aşa cum a fost prezentată este instabilă.
- Pentru anumite matrice EG poate eşua, datorită tentativei de împărţire la zero

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

- Matricea este nesingulară și bine condiționată;  
 $condA = \frac{3+\sqrt{5}}{2} \approx 2.168$
- Fenomenul este mai general; este evidențiat de o ușoară perturbație a lui  $A$

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

## Necesitatea pivotării II

- Acum procesul nu eșuează; se obține (în aritmetică exactă)

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

- În aritmetică în virgulă flotantă, dublă precizie,  $1 - 10^{20}$  nu este reprezentabil exact, el se va rotunji la  $-10^{20}$
- Factorii calculați ai descompunerii vor fi

$$\tilde{L} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad \tilde{U} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}$$

- Produsul  $\tilde{L}\tilde{U}$  nu este apropiat de  $A$

$$\tilde{L}\tilde{U} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \cdot \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix}$$

# Necesitatea pivotării III

- Rezolvând sistemul

$$\tilde{L}\tilde{U}x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

se obține  $\tilde{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ , dar soluția corectă este  $x = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

- Explicație: *EG nu este nici regresiv stabilă, nici stabilă (ca algoritm de factorizare). Mai mult, matricele triunghiulare obținute pot fi foarte prost condiționate, introducându-se astfel o sursă suplimentară de instabilitate.*
- **Observație:** Dacă un pas al unui algoritm nu este regresiv stabil, algoritmul întreg poate fi instabil.

# Pivotare

- La pasul  $k$ , am utilizat elementul  $k, k$  al matricei ca pivot și am introdus zerouri în coloana  $k$  a liniilor rămase

$$\left[ \begin{array}{ccccc} \times & \times & \times & \times & \times \\ & \mathbf{x}_{kk} & \times & \times & \times \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \end{array} \right] \rightarrow \left[ \begin{array}{ccccc} \times & \times & \times & \times & \times \\ & \mathbf{x}_{kk} & \times & \times & \times \\ \mathbf{0} & \times & \times & \times & \\ \mathbf{0} & \times & \times & \times & \\ \mathbf{0} & \times & \times & \times & \end{array} \right]$$

- Dar, orice alt element  $i \geq k$  din coloana  $k$  poate fi utilizat ca pivot:

$$\left[ \begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \\ & \mathbf{x}_{ik} & \times & \times & \times \\ \times & \times & \times & \times & \end{array} \right] \rightarrow \left[ \begin{array}{ccccc} \times & \times & \times & \times & \times \\ \mathbf{0} & \times & \times & \times & \\ \mathbf{0} & \times & \times & \times & \\ & \mathbf{x}_{ik} & \times & \times & \times \\ \mathbf{0} & \times & \times & \times & \end{array} \right]$$

# Pivotare

- De asemenea, se poate utiliza orice altă coloană  $j \geq k$ :

$$\left[ \begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \mathbf{x}_{ij} & \times & \times & \times \\ \times & \times & \times & \times & \times \end{array} \right] \longrightarrow \left[ \begin{array}{ccccc} \times & \times & \times & \times & \times \\ \times & \mathbf{0} & \times & \times & \times \\ \times & \mathbf{0} & \times & \times & \times \\ \times & x_{ij} & \times & \times & \times \\ \times & \mathbf{0} & \times & \times & \times \end{array} \right]$$

- Alegând diferenți pivoți ne asigurăm că putem evita pivoții nuli sau foarte mici
- În loc să utilizăm pivoți în poziții diferite, putem interschimba linii sau coloane și să utilizăm algoritmul standard ([pivotare](#))
- O implementare concretă poate face pivotarea indirect, fără a muta fizic datele

# Pivotare parțială

- Alegerea pivoților dintre toți candidații valizi este costisitoare (**pivotare completă**)
- Considerăm doar pivoții din coloana  $k$  și interschimbăm liniile (**pivotare parțială**)

$$\begin{array}{c}
 \left[ \begin{array}{ccccx} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \\ \mathbf{x}_{ik} & \times & \times & \times & \\ \times & \times & \times & \times & \end{array} \right] \xrightarrow{P_1} \left[ \begin{array}{ccccc} \times & \times & \times & \times & \times \\ \mathbf{x}_{ik} & \times & \times & \times & \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \end{array} \right] \xrightarrow{L_1} \left[ \begin{array}{ccccc} \times & \times & \times & \times & \times \\ \mathbf{x}_{ik} & \times & \times & \times & \\ \mathbf{0} & \times & \times & \times & \\ \mathbf{0} & \times & \times & \times & \\ \mathbf{0} & \times & \times & \times & \end{array} \right] \\
 \text{Selectie pivot} \qquad \qquad \qquad \text{Interschimbare linii} \qquad \qquad \qquad \text{Eliminare}
 \end{array}$$

- Cu operații matriceale:

$$L_{m-1} P_{m-1} \dots L_2 P_2 L_1 P_1 A = U$$

# Factorizarea $PA = LU$

- Pentru a combina toți  $L_k$  și toți  $P_k$  în forma dorită de noi, rescriem factorizarea precedentă sub forma

$$\begin{aligned} L_{m-1}P_{m-1} \dots L_2P_2L_1P_1A &= U \\ (L'_m \dots L'_2L'_1)(P_{m-1} \dots P_2P_1)A &= U \end{aligned}$$

unde

$$L'_k = P_{m-1} \dots P_{k+1}L_kP_{k+1}^{-1} \dots P_{m-1}^{-1}$$

- Aceasta ne dă factorizare (descompunerea) LU a lui  $A$

$$PA = LU$$

# Eliminarea gaussiană cu pivotare parțială

- Factorizează  $A \in \mathbb{C}^{m \times m}$  în  $PA = LU$

## Eliminare gaussiană cu pivotare parțială

$U := A; L := I; P := I;$

**for**  $k := 1$  **to**  $m - 1$  **do**

Alege  $i \geq k$  care maximizează  $|u_{ik}|$ ;

$u_{k,k:m} \leftrightarrow u_{i,k:m}$ ; {interschimbare}

$\ell_{k,1:k-1} \leftrightarrow \ell_{i,1:k-1}$ ;

$P_{k,:} \leftrightarrow P_{i,:}$ ;

**for**  $j := k + 1$  **to**  $m$  **do**

$\ell_{jk} := u_{jk} / u_{kk}$ ;

$u_{j,k:m} := u_{j,k:m} - \ell_{jk} u_{k,k:m}$ ;

Algoritmul devine mai eficient dacă se fac toate calculele *în situ* (*în matricea A*).

# Cod MATLAB pentru descompunerea LUP

```

function [L,U,P]=lup(A)
%LUP - LUP decomposition of A
%permute effectively lines

[m,n]=size(A);
P=zeros(m,n);
piv=(1:m)';
for i=1:m-1
    %pivoting
    [pm,kp]=max(abs(A(i:m,i)));
    kp=kp+i-1;
    %line interchange
    if i~=kp
        A([i,kp],:)=A([kp,i],:);
        piv([i,kp])=piv([kp,i]);
    end
    %Schur complement
    lin=i+1:m;
    A(lin,i)=A(lin,i)/A(i,i);
    A(lin,lin)=A(lin,lin)-...
        A(lin,i)*A(i,lin);
    end;
    for i=1:m
        P(i,piv(i))=1;
    end;
    U=triu(A);
    L=tril(A,-1)+eye(m);
end

```

# Exemplu

- Rezolvați sistemul

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 2 & 4 & 2 \end{bmatrix} x = \begin{bmatrix} 3 \\ 4 \\ 8 \end{bmatrix}$$

prin descompunere LUP.

- Soluție:** Avem

$$\left[ \begin{array}{c|cccc} 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 2 \\ 3 & 2 & 4 & 2 \end{array} \right] \sim \left[ \begin{array}{c|ccc} 3 & 2 & 4 & 2 \\ 2 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 \end{array} \right] \sim \left[ \begin{array}{c|ccc} 3 & 2 & 4 & 2 \\ 2 & \frac{1}{2} & 1 & 2 \\ 1 & \frac{1}{2} & 1 & 1 \end{array} \right]$$

$$\left[ \begin{array}{c|ccc} 3 & 2 & 4 & 2 \\ 2 & \frac{1}{2} & -1 & 1 \\ 1 & \frac{1}{2} & -1 & 0 \end{array} \right] \sim \left[ \begin{array}{c|ccc} 3 & 2 & 4 & 2 \\ 2 & \frac{1}{2} & -1 & 1 \\ 1 & \frac{1}{2} & 1 & 0 \end{array} \right] \sim \left[ \begin{array}{c|ccc} 3 & 2 & 4 & 2 \\ 2 & \frac{1}{2} & -1 & 1 \\ 1 & \frac{1}{2} & 1 & -1 \end{array} \right].$$

## Exemplu (continuare)

- Deci

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 1 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & 4 & 2 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

- Sistemele triunghiulare corespunzătoare sunt

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 1 & 1 \end{bmatrix} y = Pb = \begin{bmatrix} 8 \\ 4 \\ 3 \end{bmatrix},$$

cu soluția  $y = [8, 0, -1]^T$

## Exemplu (continuare)

- și

$$\begin{bmatrix} 2 & 4 & 2 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix} x = \begin{bmatrix} 8 \\ 0 \\ -1 \end{bmatrix},$$

cu soluția  $x = [1, 1, 1]^T$ .

- Verificare

$$PA = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 2 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$LU = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 4 & 2 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}.$$

# Pivotare totală

- Dacă se selectează pivoți din coloane diferite, sunt necesare matrice de permutare la stânga  $Q_k$ :

$$L_{m-1} P_{m-1} \cdots L_2 P_2 L_1 P_1 A Q_1 Q_2 \cdots Q_{m-1} = U$$

$$(L'_{m-1} \cdots L'_2 L'_1) (P_{m-1} \cdots P_2 P_1) A (Q_1 Q_2 \cdots Q_{m-1}) = U$$

- Punem

$$L = (L'_{m-1} \cdots L'_2 L'_1)^{-1}$$

$$P = P_{m-1} \cdots P_2 P_1$$

$$Q = Q_1 Q_2 \cdots Q_{m-1}$$

pentru a obține

$$PAQ = LU$$



Liu Hui c. 220 –c. 280  
 Matematician chinez, a  
 discutat eliminarea  
 „gaussiană” în comentariile  
 sale asupra lucrării „Cele nouă  
 capitole ale artei matematice”  
 263 AD



Carl Friedrich Gauss 1777-1855  
 Matematică, astronomie,  
 geodezie, magnetism  
 1809 GE  
 (Ca adolescent în  
 Braunschweig a descoperit  
 teorema binomială,  
 reciprocitatea pătratică, media  
 aritmetică-geometrică... )  
 1807-1855: Universitatea din  
 Göttingen

# Stabilitatea LU fără pivotare

- Pentru  $A = LU$  calculată fără pivotare:

$$\tilde{L}\tilde{U} = A + \delta A, \quad \frac{\|\delta A\|}{\|L\| \|U\|} = O(\text{eps})$$

- Eroare se referă la  $\tilde{L}\tilde{U}$ , nu la  $\tilde{L}$  sau  $\tilde{U}$
- Notă: la numitor apare  $\|L\| \|U\|$ , nu  $\|A\|$
- $\|L\|$  și  $\|U\|$  pot fi arbitrar de mari, de exemplu

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

- Deci, algoritmul este **nestabil**

# Stabilitatea LU cu pivotare

- Dacă se face pivotare, toate elementele lui  $L$  sunt  $\leq 1$  în modul, deci  $\|L\| = O(1)$
- Pentru a măsura creșterea lui  $U$ , se introduce factorul de creștere

$$\rho = \frac{\max_{ij} |u_{ij}|}{\max_{ij} |a_{ij}|}$$

care implică  $\|U\| = O(\rho \|A\|)$

- Pentru descompunerea  $PA = LU$  calculată cu pivotare:

$$\widetilde{L}\widetilde{U} = PA + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\rho \text{eps})$$

- Dacă  $\rho = O(1)$ , atunci algoritmul este regresiv stabil

# Factorul de creștere I

- Considerăm matricea

$$\begin{bmatrix} 1 & & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ -1 & -1 & 1 & & \\ -1 & -1 & -1 & 1 & \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & 1 \\ & 1 & 2 \\ & 1 & 4 \\ & 1 & 8 \\ & & 16 \end{bmatrix}$$

- Nu apare nici o pivotare, deci aceasta este o factorizare  $PA = LU$
- Factorul de creștere  $\rho = 16 = 2^{m-1}$  (se poate arăta că acesta este cazul cel mai nefavorabil)
- Deci,  $\rho = 2^{m-1} = O(1)$ , uniform, pentru toate matricele de dimensiune  $m$
- Regresiv stabil conform definiției, dar rezultatul poate fi inutil
- Totuși, nu se știe exact de ce, factorii de creștere sunt mici în practică

## Factorul de creștere II

- **Conjectură:** factorii de creștere de ordin mai mare ca  $1/2$  sunt rari în practică
- adică, pentru orice  $\alpha > 1/2$  și  $M > 0$ , probabilitatea evenimentului  $\rho > m^\alpha$  este mai mică decât  $m^{-M}$ , pentru  $m$  suficient de mare.

$$\forall \alpha > \frac{1}{2} \forall M > 0 \exists m_0, \forall m > m_0 P(\rho > m^\alpha) < m^{-M}.$$

- Problemă deschisă: conjectura este adevărată sau falsă?

# Matrice SPD

- Reamintim:
  - $A \in \mathbb{R}^{m \times m}$  este simetrică dacă  $a_{ij} = a_{ji}$ , sau  $A = A^T$
  - $A \in \mathbb{C}^{m \times m}$  este hermitiană dacă  $a_{ij} = \bar{a}_{ji}$ , sau  $A = A^*$
- O matrice hermitiană  $A$  este hermitian pozitiv definită dacă  $x^*Ax > 0$  pentru  $x \neq 0$ 
  - $x^*Ax$  este întotdeauna real deoarece  $x^*Ay = \overline{y^*Ax}$
  - Simetric pozitiv definită, sau SPD, pentru matrice reale
- dacă  $A$  este  $m \times m$  PD și  $X$  are rang maxim, atunci  $X^*AX$  este PD
  - Deoarece  $(X^*AX)^* = X^*AX$ , și dacă  $x \neq 0$  atunci  $Xx \neq 0$  și  $x^*(X^*AX)x = (Xx)^*A(Xx) > 0$
  - Orice submatrice principală a lui  $A$  este PD, și orice element diagonal  $a_{ii} > 0$
- matricele PD au valori proprii reale pozitive și vectori proprii ortonormali

# Factorizarea Cholesky

- Se elimină sub pivot și la dreapta pivotului (datorită simetriei):

$$\begin{aligned} A &= \begin{bmatrix} a_{11} & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ w/\alpha & I \end{bmatrix} \begin{bmatrix} \alpha & w^*/\alpha \\ 0 & K - ww^*/a_{11} \end{bmatrix} \\ &= \begin{bmatrix} \alpha & 0 \\ w/\alpha & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^*/a_{11} \end{bmatrix} \begin{bmatrix} \alpha & w^*/\alpha \\ 0 & I \end{bmatrix} = R_1^* A_1 R_1 \end{aligned}$$

unde  $\alpha = \sqrt{a_{11}}$

- $K - ww^*/a_{11}$  este o submatrice principală a matricei PD  $R_1^{*-1} A R_1^{-1}$ , deci *elementul ei din stânga sus este pozitiv*

# Factorizarea Cholesky

- Se aplică recursiv și se obține

$$A = (R_1^* R_2^* \dots R_m^*)(R_m \dots R_2 R_1) = R^* R, \quad r_{ii} > 0$$

- Existența și unicitatea: orice matrice HPD are o factorizare Cholesky unică
  - Algoritmul recursiv de pe slide-ul precedent nu eșuează niciodată
  - Rezultă și unicitatea, deoarece  $\alpha = \sqrt{a_{11}}$  este determinat unic (dat) la fiecare pas și la fel, întreaga linie  $w/\alpha$

# Algoritmul de factorizare Cholesky

- Factorizează matricea HPD  $A \in \mathbb{C}^{m \times m}$  în  $A = R^*R$ :

## Factorizare Cholesky

$R := A;$

**for**  $k := 1$  **to**  $m$  **do**

**for**  $j := k + 1$  **to**  $m$  **do**

$$R_{j,j:m} := R_{j,j:m} - R_{k,j:m} \overline{R_{k,j}} / R_{k,k}$$

$$R_{k,k:m} := R_{k,k:m} / \sqrt{R_{k,k}}$$

- Complexitatea (număr de operații)

$$\sum_{k=1}^m \sum_{j=k+1}^m 2(m-j) \sim 2 \sum_{k=1}^m \sum_{j=1}^k j \sim \sum_{k=1}^m k^2 \sim \frac{m^3}{3}$$

# Exemplu

- Să se rezolve sistemul

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 5 & 3 \\ 1 & 3 & 3 \end{bmatrix} x = \begin{bmatrix} 4 \\ 10 \\ 7 \end{bmatrix}$$

folosind descompunerea Cholesky.

- **Soluție:** Calculând radicalii pivotilor și complementele Schur se obține

$$B = \begin{bmatrix} 1 & 2 & 1 \\ 5 & 3 \\ 3 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 \\ 2 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 \\ 1 \end{bmatrix}.$$

## Exemplu

- Sistemele corespunzătoare sunt:

$$\begin{bmatrix} 1 & & \\ 2 & 1 & \\ 1 & 1 & 1 \end{bmatrix} y = \begin{bmatrix} 4 \\ 10 \\ 7 \end{bmatrix}$$

cu soluția  $y = [ 4 \ 2 \ 1 ]^T$

- și

$$\begin{bmatrix} 1 & 2 & 1 \\ & 1 & 1 \\ & & 1 \end{bmatrix} x = \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix},$$

cu soluția  $x = [ 1 \ 1 \ 1 ]^T$ .

# Stabilitatea

- Factorul Cholesky calculat  $\tilde{R}$  satisfacă

$$\tilde{R}^* \tilde{R} = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\text{eps})$$

algoritmul este regresiv stabil

- Dar, eroarea în  $\tilde{R}$  poate fi mare ,

$$\|\tilde{R} - R\| / \|R\| = O(\kappa(A)\text{eps})$$

- Rezolvare  $Ax = b$  pentru HPD  $A$  și cu două substituții
  - Numărul de operații Cholesky  $\sim m^3/3$
  - Algoritm regresiv stabil:

$$(A + \Delta A)\tilde{x} = b, \quad \frac{\|\Delta A\|}{\|A\|} = O(\text{eps})$$



John von Neumann  
(1903-1957)



André Louis Cholesky  
(1875-1918)

# Backslash în MATLAB

- $x = A \backslash b$  pentru  $A$  densă realizează următorii pași
- ① Dacă  $A$  este triunghiulară superior sau inferior se rezolvă prin substituție inversă sau directă

# Backslash în MATLAB

- $x = A \backslash b$  pentru  $A$  densă realizează următorii pași
- ① Dacă  $A$  este triunghiulară superior sau inferior se rezolvă prin substituție inversă sau directă
- ② Dacă  $A$  este o permutare a unei matrice triunghiulare, se rezolvă prin substituție (utilă pentru  $[L, U] = lu(A)$  căci  $L$  este permutată)

# Backslash în MATLAB

- $x = A \backslash b$  pentru  $A$  densă realizează următorii pași
- ① Dacă  $A$  este triunghiulară superior sau inferior se rezolvă prin substituție inversă sau directă
- ② Dacă  $A$  este o permutare a unei matrice triunghiulare, se rezolvă prin substituție (utilă pentru  $[L, U] = lu(A)$  căci  $L$  este permutată)
- ③ Dacă  $A$  este simetrică sau hermitiană

# Backslash în MATLAB

- $x = A \backslash b$  pentru  $A$  densă realizează următorii pași
- ① Dacă  $A$  este triunghiulară superior sau inferior se rezolvă prin substituție inversă sau directă
- ② Dacă  $A$  este o permutare a unei matrice triunghiulare, se rezolvă prin substituție (utilă pentru  $[L, U] = lu(A)$  căci  $L$  este permutată)
- ③ Dacă  $A$  este simetrică sau hermitiană
  - Se verifică dacă toate elementele diagonale sunt pozitive

# Backslash în MATLAB

- $x = A \backslash b$  pentru  $A$  densă realizează următorii pași
- ① Dacă  $A$  este triunghiulară superior sau inferior se rezolvă prin substituție inversă sau directă
- ② Dacă  $A$  este o permutare a unei matrice triunghiulare, se rezolvă prin substituție (utilă pentru  $[L, U] = lu(A)$  căci  $L$  este permutată)
- ③ Dacă  $A$  este simetrică sau hermitiană
  - Se verifică dacă toate elementele diagonale sunt pozitive
  - Se încearcă cu Cholesky; dacă se termină cu succes se rezolvă prin substituție

# Backslash în MATLAB

- $x = A \backslash b$  pentru  $A$  densă realizează următorii pași
- ① Dacă  $A$  este triunghiulară superior sau inferior se rezolvă prin substituție inversă sau directă
- ② Dacă  $A$  este o permutare a unei matrice triunghiulare, se rezolvă prin substituție (utilă pentru  $[L, U] = lu(A)$  căci  $L$  este permutată)
- ③ Dacă  $A$  este simetrică sau hermitiană
  - Se verifică dacă toate elementele diagonale sunt pozitive
  - Se încearcă cu Cholesky; dacă se termină cu succes se rezolvă prin substituție
- ④ Dacă  $A$  este Hessenberg, se reduce la o matrice triunghiulară superior și apoi se rezolvă prin substituție inversă

# Backslash în MATLAB

- $x = A \backslash b$  pentru  $A$  densă realizează următorii pași
- ① Dacă  $A$  este triunghiulară superior sau inferior se rezolvă prin substituție inversă sau directă
- ② Dacă  $A$  este o permutare a unei matrice triunghiulare, se rezolvă prin substituție (utilă pentru  $[L, U] = lu(A)$  căci  $L$  este permutată)
- ③ Dacă  $A$  este simetrică sau hermitiană
  - Se verifică dacă toate elementele diagonale sunt pozitive
  - Se încearcă cu Cholesky; dacă se termină cu succes se rezolvă prin substituție
- ④ Dacă  $A$  este Hessenberg, se reduce la o matrice triunghiulară superior și apoi se rezolvă prin substituție inversă
- ⑤ Dacă  $A$  este pătratică, se factorizează  $PA = LU$  și se rezolvă prin substituție inversă

# Backslash în MATLAB

- $x = A \backslash b$  pentru  $A$  densă realizează următorii pași
- ① Dacă  $A$  este triunghiulară superior sau inferior se rezolvă prin substituție inversă sau directă
- ② Dacă  $A$  este o permutare a unei matrice triunghiulare, se rezolvă prin substituție (utilă pentru  $[L, U] = lu(A)$  căci  $L$  este permutată)
- ③ Dacă  $A$  este simetrică sau hermitiană
  - Se verifică dacă toate elementele diagonale sunt pozitive
  - Se încearcă cu Cholesky; dacă se termină cu succes se rezolvă prin substituție
- ④ Dacă  $A$  este Hessenberg, se reduce la o matrice triunghiulară superior și apoi se rezolvă prin substituție inversă
- ⑤ Dacă  $A$  este pătratică, se factorizează  $PA = LU$  și se rezolvă prin substituție inversă
- ⑥ Dacă  $A$  nu este pătratică, se face factorizare QR cu metoda Householder, și se rezolvă problema de aproximare în sensul celor mai mici pătrate

# Descompunere QR

- Fie  $A \in \mathbb{C}^{m \times n}$ . Se numește **descompunere QR** a lui  $A$  perechea de matrice  $(Q, R)$  unde  $Q \in \mathbb{C}^{m \times n}$  este unitară,  $R \in \mathbb{C}^{n \times n}$  este triunghiulară superior și  $A = QR$ .

# Triunghiularizare Householder

- Metoda lui Householder înmulțește cu matrice unitare pentru a transform matricea într-o triunghiulară; de exemplu la primul pas:

$$Q_1 A = \begin{bmatrix} r_{11} & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \times & \cdots & \times \end{bmatrix}$$

- La sfârșit, am obținut un produs de matrice ortogonale

$$\underbrace{Q_n \dots Q_2 Q_1}_{Q^*} A = R$$

- “Triunghiularizare ortogonală”

# Introducerea de zerouri

- $Q_k$  introduce zerouri sub diagonală în coloana  $k$
- Păstrează zerourile introduse anterior

$$\begin{array}{c}
 \left[ \begin{array}{ccc} \times & \times & \times \\ \times & \times & \times \end{array} \right] \xrightarrow{Q_1} \left[ \begin{array}{ccc} \times & \times & \times \\ \mathbf{0} & \times & \times \end{array} \right] \xrightarrow{Q_2} \left[ \begin{array}{ccc} \times & \times & \times \\ & \times & \times \\ \mathbf{0} & \times & \times \\ \mathbf{0} & \times & \times \\ \mathbf{0} & \times & \times \end{array} \right] \xrightarrow{Q_3} \left[ \begin{array}{ccc} \times & \times & \times \\ & \times & \times \\ & & \times \\ \mathbf{0} & & \mathbf{0} \\ \mathbf{0} & & \mathbf{0} \end{array} \right] \\
 A \qquad \qquad \qquad Q_1 A \qquad \qquad \qquad Q_2 Q_1 A \qquad \qquad \qquad Q_3 Q_2 Q_1 A
 \end{array}$$

# Refectori Householder

- Fie  $Q_k$  de forma

$$Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}$$

unde  $I$  este  $(k - 1) \times (k - 1)$  și  $F$  este  $(m - k + 1) \times (m - k + 1)$

- Creăm reflectorul Householder  $F$  care introduce zerouri:

$$x = \begin{bmatrix} \times \\ \times \\ \vdots \\ \times \end{bmatrix} \quad Fx = \begin{bmatrix} \|x\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\| e_1$$

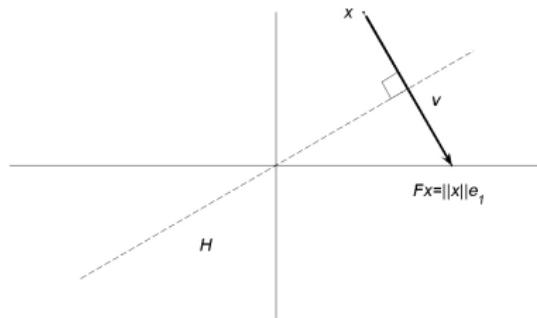
# Refectori Householder-Ideea

- Ideea: reflectăm în raport cu hiperplanul  $H$ , ortogonal pe  $v = \|x\|_2 e_1 - x$ , aplicând matricea unitară

$$F = I - 2 \frac{vv^*}{v^*v}$$

- A se compara cu proiectoarele

$$P_{\perp v} = I - \frac{vv^*}{v^*v}$$



# Determinarea reflectorului

- reflexie Householder:  $P = I - 2uu^T$ ,  $\|u\|_2 = 1$ ;  $P$  simetrică și ortogonală, deoarece  $P = P^T$  și

$$PP^T = (I - 2uu^T)(I - 2uu^T) = I - 4uu^T + 4uu^Tuu^T = I$$

- Dorim  $Px = [c, 0, \dots, 0]^T = ce_1$  (anulăm toate componentele lui  $x$  exceptând prima)

$$Px = x - 2u(u^T x) = ce_1 \implies u = \frac{1}{2u^T x} (x - ce_1)$$

$$\|x\|_2 = \|Px\|_2 = |c|$$

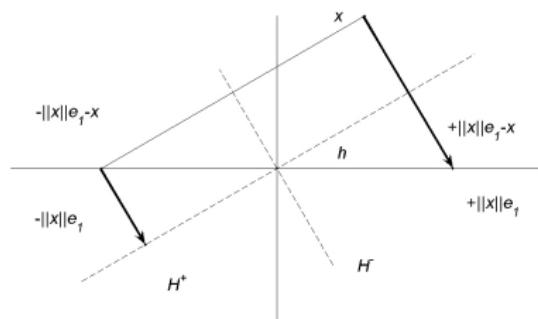
- obținem  $u$  paralel cu  $\tilde{u} = x \pm \|x\|_2 e_1$ , deci  $u = \tilde{u} / \|\tilde{u}\|_2$ . Orice alegere de semn corespunde; vom alege

$$\tilde{u} = [x_1 + sign(x_1) \|x\|_2, x_2, \dots, x_n]^T, \quad u = \tilde{u} / \|\tilde{u}\|_2.$$

## Alegerea reflectorului

- Putem aplica reflexia oricărui multiplu  $z$  al lui  $\|x\| e_1$  cu  $|z| = 1$
  - Proprietăți numerice mai bune pentru  $\|v\|$  mare, de exemplu  
 $v = \text{sign}(x_1) \|x\| e_1 + x$

- Notă:  $\text{sign}(0) = 1$ , dar în MATLAB,  $\text{sign}(0) == 0$



# Algoritmul lui Householder

- Calculează factorul  $R$  al descompunerii QR a matricei  $m \times n$   $A$  ( $m \geq n$ )
- Lasă rezultatul în  $A$ , memorând vectorii de reflexie  $v_k$  pentru utilizare ulterioară

## Factorizare QR prin metoda Householder

```
for k := 1 to n do
    x := Ak:m,k;
    vk := sign(x1)||x||2e1 + x;
    vk := vk / ||vk||2;
    Ak:m,k:n = Ak:m,k:n - 2vk(vk*Ak:m,k:n)
```

# Aplicarea sau obținerea lui $Q$

- Calculăm  $Q^*b = Q_n \dots Q_2 Q_1 b$  și  $Qx = Q_1 Q_2 \dots Q_n x$  implicit
- Pentru a crea  $Q$  explicit, aplicăm pentru  $x = I$

## Calculul implicit al lui $Q^*b$

**for**  $k := 1$  **to**  $n$  **do**

$$b_{k:m} = b_{k:m} - 2v_k (v_k^* b_{k:m});$$

## Calculul implicit al lui $Qx$

**for**  $k := n$  **downto** 1 **do**

$$x_{k:m} = x_{k:m} - 2v_k (v_k^* x_{k:m});$$

# Complexitatea QR-Householder

- Cea mai mare parte a efortului

$$A_{k:m,k:n} = A_{k:m,k:n} - 2v_k (v_k^* A_{k:m,k:n})$$

- Operații pe iterație:
  - $2(m - k)(n - k)$  pentru produsele scalare  $v_k^* A_{k:m,k:n}$
  - $(m - k)(n - k)$  pentru produsul exterior  $2v_k(\dots)$
  - $(m - k)(n - k)$  pentru scăderea  $A_{k:m,k:n} - \dots$
  - $4(m - k)(n - k)$  total
- Încluzând ciclul exterior, totalul devine

$$\sum_{k=1}^n 4(m - k)(n - k) = 4 \sum_{k=1}^n (mn - k(m + n) + k^2)$$

$$\sim 4mn^2 - 4(m + n)n^2/2 + 4n^3/3 = 2mn^2 - 2n^3/3$$



**Figura:** Alston S. Householder (1904-1993), matematician american. Contribuții importante: biologie matematică, algebră liniară numerică. Cartea sa "The Theory of Matrices in Numerical Analysis" a avut un mare impact asupra dezvoltării analizei numerice și a informaticii.



**Figura:** James Wallace Givens (1910-1993) Pionier al algebrei liniare numerice și informaticii

## Exemplu

*Calculați descompunerea QR a matricei*

$$A = \begin{bmatrix} 3 & 1 \\ 4 & 1 \end{bmatrix}.$$

**Soluție.** Reflexia pentru prima coloană este  $P = I - 2uu^T$ . Vectorul  $u$  se determină astfel:

$$\tilde{u} = \begin{bmatrix} x_1 + sign(x_1) \|x\|_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 + 5 \\ 4 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \end{bmatrix}.$$

$$\|\tilde{u}\| = \sqrt{8^2 + 4^2} = 4\sqrt{5}$$

$$u = \frac{\tilde{u}}{\|\tilde{u}\|} = \begin{bmatrix} \frac{2\sqrt{5}}{5} \\ \frac{\sqrt{5}}{5} \end{bmatrix}.$$

Matricea de reflexie este

$$\begin{aligned}
 P &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - 2 \begin{bmatrix} \frac{2\sqrt{5}}{5} \\ \frac{\sqrt{5}}{5} \end{bmatrix} \begin{bmatrix} \frac{2\sqrt{5}}{5} \\ \frac{\sqrt{5}}{5} \end{bmatrix}^T \\
 &= \begin{bmatrix} -\frac{3}{5} & -\frac{4}{5} \\ -\frac{4}{5} & \frac{3}{5} \end{bmatrix} = Q^T.
 \end{aligned}$$

Se obține:

$$\begin{aligned}
 Q &= \begin{bmatrix} -\frac{3}{5} & -\frac{4}{5} \\ -\frac{4}{5} & \frac{3}{5} \end{bmatrix} \\
 R &= Q^T A = \begin{bmatrix} -\frac{3}{5} & -\frac{4}{5} \\ -\frac{4}{5} & \frac{3}{5} \end{bmatrix} \cdot \begin{bmatrix} 3 & 1 \\ 4 & 1 \end{bmatrix} = \begin{bmatrix} -5 & -\frac{7}{5} \\ 0 & -\frac{1}{5} \end{bmatrix}.
 \end{aligned}$$



# Rotații Givens I

- O rotație Givens

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

rotește un vector  $x \in \mathbb{R}^2$  în sens trigonometric cu unghiul  $\theta$

- Rotația Givens cu unghiul  $\theta$  în coordonatele  $i$  și  $j$  se obține cu ajutorul matricei de mai sus, punând elementele ei în liniile și coloanele  $i$  și  $j$  și în rest elementele matricei unitate.

## Rotații Givens II

$$R(i, j, \theta) := \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & \cos \theta & \sin \theta & \\ i & & & -\sin \theta & \cos \theta & \\ & j & & & & \ddots & \\ & & & & & & 1 \\ & & & & & & & 1 \end{pmatrix}$$

# Rotații Givens III

- Dându-se  $x$ ,  $i$  și  $j$  putem anula  $x_j$  alegând  $\cos \theta$  și  $\sin \theta$  astfel încât

$$\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} \sqrt{x_i^2 + x_j^2} \\ 0 \end{bmatrix},$$

adică

$$\cos \theta = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad \sin \theta = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}.$$

- Algoritmul QR bazat pe rotații Givens este analog algoritmului bazat pe reflexii Householder, dar când anulăm coloana  $i$  se anulează un element la un moment dat.

# Teorema de punct fix a lui Banach

## Principiul contracției

Radu Trîmbițăș

UBB

6 aprilie 2020

# Introducere

În matematică, teorema de punct fix a lui Banach [1] (cunoscută și sub numele de teorema aplicației contractive sau principiul contracției) este un instrument important în teoria spațiilor metrice. Ea garantează existența și unicitatea punctelor fixe ale unor clase de aplicații ale unui spațiu metric în el însuși și furnizează o metodă constructivă de determinare a acestor puncte fixe.

## Definiție

Fie  $(X, d)$  un spațiu metric. O aplicație  $T : X \rightarrow X$  se numește **contracție** pe  $X$ , dacă există  $q \in [0, 1)$  astfel încât oricare ar fi  $x, y \in X$

$$d(T(x), T(y)) \leq qd(x, y). \quad (1)$$

## Definiție

Fie  $T : X \rightarrow X$ . Punctul  $x \in X$  se numește **punct fix** al lui  $T$  dacă  $T(x) = x$ .

# Enunțul

## Teoremă (Banach, 1922)

*Fie  $(X, d)$  un spațiu metric complet nevid și  $T : X \rightarrow X$  o contracție. Atunci  $T$  admite un punct fix unic  $x^* \in X$  (i.e.  $T(x^*) = x^*$ ). Mai mult,  $x^*$  poate fi determinată prin **metoda aproximăriilor succesive**: se pornește de la un element arbitrar  $x_0 \in X$  și se definește sirul  $\{x_n\}$  prin  $x_n = T(x_{n-1})$ ; atunci  $x_n \rightarrow x^*$ .*

## Observație

*Inegalitățile următoare sunt echivalente și descriu viteza de convergență:*

$$d(x^*, x_n) \leq \frac{q^n}{1-q} d(x_1, x_0)$$

$$d(x^*, x_{n+1}) \leq \frac{q}{1-q} d(x_{n+1}, x_n)$$

$$d(x^*, x_{n+1}) \leq q d(x^*, x_n).$$

# Demonstrație I

- Demonstrația este din [2].
- Din inegalitatea triunghiului,  $\forall x, y \in X$

$$\begin{aligned} d(x, y) &\leq d(x, T(x)) + d(T(x), T(y)) + d(T(y), y) \\ &\leq d(x, T(x)) + qd(x, y) + d(T(y), y). \end{aligned}$$

- Exprimând  $d(x, y)$  avem

$$d(x, y) \leq \frac{d(x, T(x)) + d(T(y), y)}{1 - q} \quad (2)$$

- $x, y$  puncte fixe  $\Rightarrow d(x, y) = 0 \Rightarrow x = y$  (deci unicitatea)

## Demonstrație II

- Fie  $T^n$  prin compunerea lui  $T$  cu ea însăși de  $n$  ori; observăm prin inducție că ea satisface condiția (1) cu constanta  $q^n$ . Rămîne să arătăm că pentru orice  $x_0 \in X$ , sirul  $\{T^n(x_0)\}$  este fundamental și converge către un punct  $x^* \in X$ , care, așa cum s-a observat mai sus, este punct fix al lui  $T$ . Dacă în inegalitatea (2) înlocuim  $x$  cu  $T^n(x_0)$  și  $y$  cu  $T^m(x_0)$ , obținem

$$\begin{aligned}
 d(T^n(x_0), T^m(x_0)) &\leq \frac{d(T(T^n(x_0)), T^n(x_0)) + d(T(T^m(x_0)), T^m(x_0))}{1 - q} \\
 &= \frac{d(T^n(T(x_0)), T^n(x_0)) + d(T^m(T(x_0)), T^m(x_0))}{1 - q} \\
 &\leq \frac{q^n d(T(x_0), x_0) + q^m d(T(x_0), x_0)}{1 - q} \\
 &= \frac{q^n + q^m}{1 - q} d(T(x_0), x_0)
 \end{aligned}$$

## Demonstrație III

- Deoarece  $q < 1$ , ultima expresie converge către zero cînd  $n, m \rightarrow \infty$ , deci  $\{T^n(x_0)\}$  este Cauchy.
- Făcînd  $m \rightarrow \infty$ , se obține delimitarea

$$d(T^n(x_0), x^*) \leq \frac{q^n}{1-q} d(T(x_0), x_0).$$



**Figura:** Stefan Banach (1892 - 1945) a fondat analiza funcțională modernă și a avut contribuții majore în teoria spațiilor liniare topologice, teoria măsurii, integrare, serii ortogonale.

# Bibliografie

-  Banach, S., Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales, *Fund. Math.* 3(1922), 133–181.
-  Palais, R., A simple proof of the Banach contraction principle, *J. fixed point theory appl.* 2 (2007), 221–223

# Analiză matricială și condiționarea unui sistem liniar

Norme, convergență, condiționare

Radu Trîmbițaș

UBB

21 martie 2021

# Elemente de analiză matricială

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

- ▶  $A \in \mathbb{C}^{m \times m}$ ,  $A^T$  transpusa lui  $A$ ,  $A^*$  transpusa conjugată a lui  $A$
- ▶ polinomul  $p(\lambda) = \det(A - \lambda I)$  **polinomul caracteristic** al lui  $A$ ; rădăcinile lui se numesc valori proprii ale lui  $A$
- ▶  $Ax = \lambda x$ ,  $\lambda \in \mathbb{C}$  **valoare proprie**,  $x \neq 0$  **vector proprie**
- ▶ Valoarea  $\rho(A) = \max\{|\lambda| : \lambda$  valoare proprie a lui  $A\}$ 
  - **raza spectrală** a matricei  $A$ .

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Matrice speciale

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

## ► O matrice se numește

- ▶ **normală**, dacă  $AA^* = A^*A$
- ▶ **unitară**, dacă  $AA^* = A^*A = I$
- ▶ **ortogonală**, dacă  $AA^T = A^TA = I$ ,  $A$  reală
- ▶ **hermitiană**, dacă  $A^* = A$
- ▶ **simetrică**, dacă  $A^T = A$ ,  $A$  reală

## ► O **normă matricială** este o aplicație $\|\cdot\| : \mathbb{C}^{m \times m} \rightarrow \mathbb{R}$ , care pentru orice matrice $A, B$ și orice scalar $\alpha \in \mathbb{C}$ verifică

- (NM1)  $\|A\| \geq 0$ ,  $\|A\| = 0 \Leftrightarrow A = 0$
- (NM2)  $\|\alpha A\| = |\alpha| \|A\|$
- (NM3)  $\|A + B\| \leq \|A\| + \|B\|$
- (NM4)  $\|AB\| \leq \|A\| \|B\|$

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

## Norme matriciale - exemple

- fiind dată o normă vectorială  $\|\cdot\|$  pe  $\mathbb{C}^n$ , aplicația

$$\|\cdot\| : \mathbb{C}^{m \times n} \rightarrow \mathbb{R}$$

$$\|A\| = \sup_{\substack{v \in \mathbb{C}^n \\ v \neq 0}} \frac{\|Av\|}{\|v\|} = \sup_{\substack{v \in \mathbb{C}^n \\ \|v\| \leq 1}} \|Av\| = \sup_{\substack{v \in \mathbb{C}^n \\ \|v\|=1}} \|Av\|$$

este o normă matricială numita **normă matricială subordonată** (normei vectoriale date) sau **normă indușă** (de normă vectorială) sau **normă naturală**.

- Orice normă subordonată verifică  $\|I\| = 1$
- Un exemplu important de normă nesubordonată (neindusă) este norma Frobenius

$$\|A\|_F = \left( \sum_i \sum_j |a_{ij}|^2 \right)^{1/2} = (\text{tr}(A^* A))^{1/2}.$$

- $\|I\|_F = \sqrt{n}$ , deci norma Frobenius nu este subordonată

## Teoremă

Fie  $A \in \mathbb{C}^{m \times m}$ . Atunci

$$\|A\|_1 = \sup_{v \in \mathbb{C}^m \setminus \{0\}} \frac{\|Av\|_1}{\|v\|_1} = \max_j \sum_i |a_{ij}|$$

$$\|A\|_2 = \sup_{v \in \mathbb{C}^m \setminus \{0\}} \frac{\|Av\|_2}{\|v\|_2} = \sqrt{\rho(AA^*)} = \sqrt{\rho(A^*A)} = \|A^*\|_2$$

$$\|A\|_\infty = \sup_{v \in \mathbb{C}^m \setminus \{0\}} \frac{\|Av\|_\infty}{\|v\|_\infty} = \max_i \sum_j |a_{ij}|$$

Dacă  $A$  este normală ( $AA^* = A^*A$ ), atunci  $\|A\|_2 = \rho(A)$ .

# Exemple

Fie matricele

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 3 \end{bmatrix}, B = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 2 \\ 2 & 1 & -2 \end{bmatrix}.$$

Normele uzuale ale lui  $A$  și  $B$  vor fi

$$\|A\|_1 = 5, \|A\|_\infty = 6,$$

$$\|A\|_2 = \frac{\sqrt{29} + \sqrt{17}}{2} \approx 4.7541, \|A\|_F = \sqrt{23}$$

$$\|B\|_1 = 6, \|B\|_\infty = 7,$$

$$\|B\|_2 \approx 42986, \|B\|_F = 2\sqrt{7}.$$

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Condiționarea unui sistem liniar 1

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

- ▶ Care este condiționarea problemei: dându-se  $A \in \mathbb{C}^{m \times m}$  și  $b \in \mathbb{C}^{m \times 1}$  să se rezolve sistemul  $Ax = b$ .
- ▶ Considerăm exemplul (Wilson)

$$\begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}$$

cu soluția  $\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$ .

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Condiționarea unui sistem liniar 2

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

- ▶ Perturbăm membrul drept

$$\begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \begin{bmatrix} x_1 + \Delta x_1 \\ x_2 + \Delta x_2 \\ x_3 + \Delta x_3 \\ x_4 + \Delta x_4 \end{bmatrix} = \begin{bmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{bmatrix}$$

- ▶ soluția  $\begin{bmatrix} 9.2 & -12.6 & 4.5 & -1.1 \end{bmatrix}^T$ .
- ▶ o eroare (relativă) de  $1/200$  în date  $\rightarrow$  eroare relativă de  $10/1$  (amplificare a erorii relative de 2000 de ori)

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Condiționarea unui sistem liniar 3

## ► Perturbăm matricea

$$\begin{bmatrix} 10 & 7 & 8.1 & 7.2 \\ 7.08 & 5.04 & 6 & 5 \\ 8 & 9.98 & 9.89 & 9 \\ 6.99 & 4.99 & 9 & 9.98 \end{bmatrix} \begin{bmatrix} x_1 + \Delta x_1 \\ x_2 + \Delta x_2 \\ x_3 + \Delta x_3 \\ x_4 + \Delta x_4 \end{bmatrix} = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}$$

- soluția  $\begin{bmatrix} -81 & 137 & -34 & 22 \end{bmatrix}^T$ .
- Din nou, o variație mică în datele de intrare modifică complet rezultatul
- Matricea are un aspect „bun”, ea este simetrică, determinantul ei este 1, iar inversa ei este

$$\begin{bmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{bmatrix}$$

# Estimarea numărului de condiționare

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

- ▶ Considerăm sistemul parametrizat, cu parametrul  $t$

$$(A + t\Delta A)x(t) = b + t\Delta b, \quad x(0) = x^*.$$

- ▶  $A$  nesingulară  $\implies$  funcția  $x$  este diferențiabilă în  $t = 0$  și  $x'(0) = A^{-1}(\Delta b - \Delta Ax^*)$ .
- ▶ Dezvoltarea Taylor a lui  $x(t)$  este

$$x(t) = x^* + tx'(0) + O(t^2).$$

- ▶ Estimarea erorii absolute

$$\begin{aligned}\|\Delta x(t)\| &= \|x(t) - x^*\| \leq |t| \|x'(0)\| + O(t^2) \\ &\leq |t| \|A^{-1}\| (\|\Delta b\| + \|\Delta A\| \|x^*\|) + O(t^2)\end{aligned}$$

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Estimarea numărului de condiționare 2

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

- din  $\|b\| \leq \|A\| \|x^*\|$  obținem pentru eroarea relativă

$$\begin{aligned}\frac{\|\Delta x(t)\|}{\|x^*\|} &\leq |t| \|A^{-1}\| \left( \frac{\|\Delta b\|}{\|x^*\|} + \|\Delta A\| \right) + O(t^2) \\ &\leq \|A\| \|A^{-1}\| |t| \left( \frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right) + O(t^2)\end{aligned}$$

- Introducem notațiile

$$\rho_A(t) = |t| \frac{\|\Delta A\|}{\|A\|}, \quad \rho_b(t) = |t| \frac{\|\Delta b\|}{\|b\|}$$

și putem scrie pentru eroarea relativă

$$\frac{\|\Delta x(t)\|}{\|x^*\|} \leq \|A\| \|A^{-1}\| (\rho_A(t) + \rho_b(t)) + O(t^2) \quad (1)$$

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Estimarea numărului de condiționare 3

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

## Definiție

Dacă  $A$  este nesingulară, numărul

$$\text{cond}(A) = \|A\| \|A^{-1}\| \quad (2)$$

se numește **număr de condiționare** al matricei  $A$ . Dacă  $A$  este singulară,  $\text{cond}(A) = \infty$ .

Relația (1) se poate scrie sub forma

$$\frac{\|\Delta x(t)\|}{\|x^*\|} \leq \text{cond}(A) (\rho_A(t) + \rho_b(t)) + O(t^2)$$

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Exemple de matrice prost condiționate

Analiză matricială  
și condiționarea  
unui sistem liniar

- Matricea lui Hilbert  $H_m = (h_{ij})$  cu  $h_{ij} = \frac{1}{i+j-1}$ ,  $i, j = 1, \dots, m$ . Szegő a demonstrat

$$\text{cond}_2(H_m) = \frac{(\sqrt{2} + 1)^{4m+4}}{2^{14/4}\sqrt{\pi m}}.$$

$m$	10	20	40
$\text{cond}_2(H_m)$	$1.6 \cdot 10^{13}$	$2.45 \cdot 10^{28}$	$7.65 \cdot 10^{58}$

- Matricea Vandermonde  $V = (v_{ij})$ ,  $v_{ij} = t_j^{i-1}$ ,  $i, j = 1, \dots, m$

- elemente echidistante în  $[-1, 1]$

$$\text{cond}_\infty(V_m) \sim \frac{1}{\pi} e^{-\frac{\pi}{4}} e^{m\left(\frac{\pi}{4} + \frac{1}{2} \ln 2\right)}$$

- $t_j = 1/j$ ,  $j = 1, \dots, m$ :  $\text{cond}_\infty(V_m) > m^{m+1}$ .

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Conditionarea unui  
sistem liniar

Conditionarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării



David Hilbert  
(1862-1943)



Gábor Szegő (1895-1985)

# Un rezultat util

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

## Teoremă

- (1) Fie  $A$  o matrice pătratică oarecare și  $\|\cdot\|$  o normă matricială oarecare (indusă sau nu). Atunci

$$\rho(A) \leq \|A\|.$$

- (2) Fiind dată o matrice  $A$  și un număr  $\varepsilon > 0$ , există cel puțin o normă matricială subordonată astfel încât

$$\|A\| \leq \rho(A) + \varepsilon.$$

Analiză matricială  
Norme matriciale  
Exemple de norme  
matriciale

Conditionarea unui  
sistem liniar

Conditionarea unui sistem  
liniar  
Estimarea numărului de  
condiționare  
Exemple de matrice prost  
condiționate

## Metode iterative

Rezultate utile  
Introducere  
Convergența și delimitarea  
erorii

## Metode concrete

Rafinarea iterativă  
Metoda lui Jacobi  
Metoda Gauss-Seidel  
Metoda relaxării

**Demonstrație.** (1) Fie  $p$  un vector propriu dominant, adică  $p \neq 0$ ,  $Ap = \lambda p$  și  $|\lambda| = \rho(A)$  și  $q$  un vector astfel încât  $pq^* \neq 0$ . Dar

$$\rho(A) \|pq^*\| = \|\lambda pq^*\| = \|Apq^*\| \leq \|A\| \|pq^*\|,$$

de unde prima parte.

(2)  $\exists U$  unitară a.î.  $U^{-1}AU$  este triunghiulară superior, și are valorile proprii ale lui  $A$  pe diagonală

$$U^{-1}AU = \begin{bmatrix} \lambda_1 & t_{12} & t_{13} & \cdots & t_{1m} \\ & \lambda_2 & t_{23} & \cdots & t_{2m} \\ & & \ddots & & \vdots \\ & & & \lambda_{m-1} & t_{m-1,m} \\ & & & & \lambda_m \end{bmatrix}$$

Fiecărui scalar  $\delta \neq 0$  îi asociem matricea  
 $D_\delta = \text{diag}(1, \delta, \delta^2, \dots, \delta^{m-1})$ ,

astfel ca

$$(UD_\delta)^{-1} A (UD_\delta) = \begin{bmatrix} \lambda_1 & \delta t_{12} & \delta^2 t_{13} & \cdots & \delta^{m-1} t_{1m} \\ & \lambda_2 & \delta t_{23} & \cdots & \delta^{m-2} t_{2m} \\ & & \ddots & & \vdots \\ & & & \lambda_{m-1} & \delta t_{m-1,m} \\ & & & & \lambda_m \end{bmatrix}$$

Pentru  $\varepsilon$  dat, fixăm  $\delta$  a.î.  $\sum_{j=i+1}^m |\delta^{j-i} t_{ij}| \leq \varepsilon$ ,  
 $i = 1, \dots, m-1$ .

Atunci aplicația

$$\|\cdot\| : B \in \mathbb{C}^{m \times m} \mapsto \|B\| = \left\| (UD_\delta)^{-1} B (UD_\delta) \right\|_\infty$$

îndeplinește condițiile problemei. Într-adevăr, datorită  
alegerii lui  $\delta$  și definiției  $\|\cdot\|_\infty$

$$\|A\| < \rho(A) + \varepsilon$$

și este indusă de norma vectorială

$$v \in \mathbb{C}^m \mapsto \left\| (UD_d)^{-1} v \right\|_{\infty}.$$



## Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

## Condiționarea unui sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

## Metode iterative

### Rezultate utile

Introducere

Convergența și delimitarea  
erorii

## Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

## Teoremă

Fie  $B$  o matrice pătratică de ordin  $m$ . Următoarele afirmații sunt echivalente:

- (1)  $\lim_{k \rightarrow \infty} B^k = 0$
- (2)  $\lim_{k \rightarrow \infty} B^k v = 0, \forall v \in \mathbb{C}^m$
- (3)  $\rho(B) < 1$
- (4) Există o normă matricială subordonată  $\|\cdot\|$ , astfel încât  $\|B\| < 1$

## Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

## Condiționarea unui sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

## Metode iterative

### Rezultate utile

Introducere

Convergența și delimitarea  
erorii

## Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

## Demonstrație. (1) $\Rightarrow$ (2)

$$\|B^k v\| \leq \|B^k\| \|v\| \Rightarrow \lim_{k \rightarrow \infty} B^k v = 0$$

(2) $\Rightarrow$ (3) Dacă  $\rho(B) \geq 1$ , putem găsi un  $p$  astfel încât  $p \neq 0$ ,  $Bp = \lambda p$ ,  $|\lambda| \geq 1$ . Deoarece  $B^k p = \lambda^k p$ , sirul de vectori  $(B^k p)_{k \in \mathbb{N}}$  ar putea să nu conveargă către 0.

(3) $\Rightarrow$ (4) Din teorema 3 avem  $\rho(B) < 1 \Rightarrow \exists \|\cdot\|$  astfel încât  $\|B\| \leq \rho(B) + \varepsilon$ ,  $\forall \varepsilon > 0$ , deci  $\|B\| < 1$ .

(4) $\Rightarrow$ (1)  $\|B^k\| \leq \|B\|^k \rightarrow 0$ , dacă  $\|B\| < 1$ . ■

## Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

## Conditionarea unui sistem liniar

Conditionarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

## Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

## Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

- ▶ Pentru  $A$  nesingulară, presupunem că putem reduce rezolvarea lui

$$Ax = b \quad (3)$$

la rezolvarea problemei de punct fix

$$x = Tx + c, \quad (4)$$

unde  $T$  este o matrice,  $c$  este un vector,  $I - T$  este inversabilă și punctul fix al lui (4) concide cu soluția  $x^*$  a lui (3).

- ▶ Definim metoda iterativă prin: se ia un  $x^{(0)}$  arbitrar și se definește sirul  $(x^{(k)})$  prin

$$x^{(k+1)} = Tx^{(k)} + c. \quad (5)$$

## Lemă

Dacă  $\rho(X) < 1$ , există  $(I - X)^{-1}$  și

$$(I - X)^{-1} = I + X + X^2 + \cdots + X^k + \cdots.$$

**Demonstrație.** Fie  $S_k = I + X + X^2 + \cdots + X^k$ . Deoarece

$$(I - X)S_k = I - X^{k+1},$$

avem

$$\lim_{k \rightarrow \infty} (I - X)S_k = I \implies \lim_{k \rightarrow \infty} S_k = (I - X)^{-1},$$

căci  $X^{k+1} \rightarrow 0 \iff \rho(X) < 1$ . ■

# Convergență

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

## Teoremă

U.a.s.e.

- (1) metoda (5) este convergentă
- (2)  $\rho(T) < 1$
- (3)  $\|T\| < 1$  pentru cel puțin o normă matricială

## Demonstrație.

$$\begin{aligned}x^{(k)} &= Tx^{(k-1)} + c = T(Tx^{(k-2)} + c) + c \\&= T^{(k)}x^{(0)} + (I + T + \cdots + T^{k-1})c\end{aligned}$$

(5) convergentă  $\iff (I - T)$  inversabilă  $\iff$   
 $\rho(T) < 1 \iff \exists \|\cdot\| \text{ a.î. } \|T\| < 1$  (teorema 4). ■

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Conditionarea unui  
sistem liniar

Conditionarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

## Metode iterative

Rezultate utile

Introducere

Convergență și delimitarea  
erorii

## Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

Aplicând teorema de punct fix a lui Banach obținem

## Teoremă

Dacă există  $\|\cdot\|$  a.î.  $\|T\| < 1$ , sirul  $(x^{(k)})$  definit de (5)

este convergent pentru orice  $x^{(0)} \in \mathbb{R}^m$  și au loc delimitările

$$\|x^* - x^{(k)}\| \leq \|T\|^k \|x^{(0)} - x^*\|$$

$$\|x^* - x^{(k)}\| \leq \frac{\|T\|^k}{1 - \|T\|} \|x^{(1)} - x^{(0)}\|$$

$$\leq \frac{\|T\|}{1 - \|T\|} \|x^{(1)} - x^{(0)}\|.$$

## Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

## Condiționarea unui sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

## Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

## Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Criteriul de oprire

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

## Criteriul de oprire

$$\|x^{(k)} - x^{(k-1)}\| \leq \frac{1 - \|T\|}{\|T\|} \varepsilon. \quad (6)$$

## Propoziție

Dacă  $x^*$  este soluția sistemului (3) și  $\|T\| < 1$ , atunci

$$\|x^* - x^{(k)}\| \leq \frac{\|T\|}{1 - \|T\|} \|x^{(k)} - x^{(k-1)}\| \quad (7)$$

Analiză matricială  
Norme matriciale  
Exemple de norme  
matriciale

Conditionarea unui  
sistem liniar

Conditionarea unui sistem  
liniar  
Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile  
Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă  
Metoda lui Jacobi  
Metoda Gauss-Seidel  
Metoda relaxării

# Demonstrația criteriului I

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

**Demonstrație.**  $\forall p \in \mathbb{N}^*$  avem

$$\|x^{(k+p)} - x^{(k)}\| \leq \|x^{(k+1)} - x^{(k)}\| + \dots + \|x^{(k+p)} - x^{(k+p-1)}\| \quad (8)$$

Din (5) rezultă

$$\|x^{(m+1)} - x^{(m)}\| \leq \|T\| \|x^{(m)} - x^{(m-1)}\|$$

sau pentru  $k < m$

$$\|x^{(m+1)} - x^{(m)}\| \leq \|T\|^{m-k-1} \|x^{(k)} - x^{(k-1)}\|.$$

Aplicând aceste inegalități, pentru  $m = k, \dots, k+p-1$ , relația (8) devine

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Conditionarea unui  
sistem liniar

Conditionarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Demonstrația criteriului II

$$\begin{aligned}\|x^{(k+p)} - x^{(k)}\| &\leq (\|T\| + \dots + \|T\|^p) \|x^{(k)} - x^{(k-1)}\| \\ &\leq (\|T\| + \dots + \|T\|^p + \dots) \|x^{(k)} - x^{(k-1)}\|\end{aligned}$$

de unde, deoarece  $\|T\| < 1$

$$\|x^{(k+p)} - x^{(k)}\| \leq \frac{\|T\|}{1 - \|T\|} \|x^{(k)} - x^{(k-1)}\|,$$

din care prin trecere la limită în raport cu  $p$  se obține (7). ■

Dacă  $\|T\| \leq \frac{1}{2}$ , inegalitatea (7) devine

$$\|x^* - x^{(k)}\| \leq \|x^{(k)} - x^{(k-1)}\|,$$

iar criteriul de oprire

$$\|x^{(k)} - x^{(k-1)}\| < \varepsilon.$$

- Dacă metoda de rezolvare pentru  $Ax = b$  este nestabilă, atunci  $A\bar{x}_1 \neq b$ , unde  $\bar{x}_1$  este valoarea calculată. Vom calcula corecția  $\Delta x_1$  astfel încât

$$A(\bar{x} + \Delta x_1) = b \implies A\Delta x_1 = b - A\bar{x}$$

- Se rezolvă sistemul și se obține un nou  $\bar{x}$ ,  $\bar{x}_2 = \bar{x}_1 + \Delta x_1$ . Dacă din nou  $A\bar{x}_2 \neq b$ , se calculează o nouă corecție până când

$$\|\Delta x_i - \Delta x_{i-1}\| < \varepsilon \text{ sau } \|b - A\bar{x}_i\| < \varepsilon$$

- Calculul vectorului  $r_i = b - A\bar{x}_i$ , numit **reziduu**, se va efectua în dublă precizie.

# Metode concrete

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

- ▶ Fie sistemul  $Ax = b$ ,  $A$  inversabilă.
- ▶ Scriem  $A$  sub forma  $A = M - N$ , unde  $M$  este ușor de inversat (diagonală, triunghiulară, etc.)

$$Ax = b \iff Mx = Nx + b \iff x = M^{-1}Nx + M^{-1}b$$

- ▶ Ultima ecuație are forma  $x = Tx + c$ , unde  $T = M^{-1}N$  și  $c = M^{-1}b$ .
- ▶ Obținem iterațiile

$$x^{(0)} = \text{arbitrar}$$

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b.$$

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Conditionarea unui  
sistem liniar

Conditionarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

- ▶ Considerăm descompunerea  $A = D - L - U$ , unde  $D = \text{diag}(A)$ ,  $L = -\text{tril}(A, -1)$ ,  $U = -\text{triu}(A, 1)$ .
- ▶ Se ia  $M = D$ ,  $N = L + U$ .
- ▶ Se obține  $T = T_J = D^{-1}(L + U)$ ,  $c = c_J = D^{-1}b$ .
- ▶ Metoda se numește **metoda lui Jacobi**
- ▶ Pe componente

$$x_i^{(k)} = \frac{1}{a_{ii}} \left( b_i - \sum_{\substack{j=1 \\ j \neq i}}^m a_{ij} x_j^{(k-1)} \right), \quad i = 1, \dots, m, \quad k = 1, 2, \dots$$

- ▶ substituția simultană

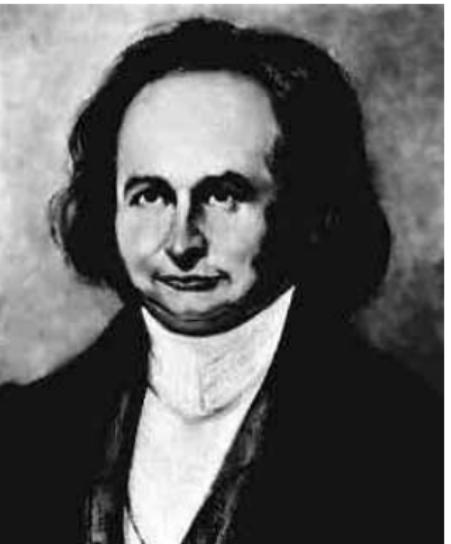


Figura: Carl Gustav Jacob Jacobi (1804-1851)

### Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

### Condiționarea unui sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

### Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

### Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Metoda Gauss-Seidel

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

- ▶ În descompunerea  $A = D - L - U$ , se ia  $M = D - L$ ,  $N = U$
- ▶ Se obține  $T = T_{GS} = (D - L)^{-1}U$ ,  $c_{GS} = (D - L)^{-1}b$ .
- ▶ **Metoda Gauss-Seidel**
- ▶ pe componente

$$x_i^{(k)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^m a_{ij} x_j^{(k-1)} \right),$$
$$i = 1, \dots, m, \quad k = 1, 2, \dots$$

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

**Metoda Gauss-Seidel**

Metoda relaxării

# Metoda Gauss-Seidel

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

- Pornim de la iterațiile Jacobi

$$x_i^{(k)} = \frac{1}{a_{ii}} \left( b_i - \sum_{\substack{j=1 \\ j \neq i}}^m a_{ij} x_j^{(k-1)} \right), \quad i = 1, \dots, m, \quad k = 1, 2, \dots$$

- deoarece  $x_j^{(k-1)}, j < i$  au fost deja actualizate le folosim  
în iterație

$$x_i^{(k)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^m a_{ij} x_j^{(k-1)} \right),$$
$$i = 1, \dots, m, \quad k = 1, 2, \dots$$

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Metoda relaxării

- ▶ Putem îmbunătăți metoda Gauss-Seidel introducând un parametru  $\omega$  și alegând

$$M = \frac{1}{\omega}D - L$$

- ▶ Avem

$$A = \left( \frac{1}{\omega}D - L \right) - \left( \frac{1-\omega}{\omega}D + U \right)$$

- ▶ Se obține

$$\begin{aligned} T &= T_\omega = \left( \frac{1}{\omega}D - L \right)^{-1} \left( \frac{1-\omega}{\omega}D + U \right) \\ &= (D - \omega L)^{-1} ((1 - \omega)D + \omega U) \\ c &= c_\omega = (D - \omega L)^{-1} \omega b \end{aligned}$$

- ▶ variante: subrelaxare  $\omega < 1$ , suprarelaxare  $\omega > 1$ ,  
Gauss-Seidel  $\omega = 1$

# Metoda relaxării II

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

- ▶ Justificarea: pentru a accelera convergența metodei Gauss-Seidel,  $x^{(k)}$  va fi media ponderată între  $x^{(k-1)}$  și  $x^{(k)}$  al metodei Gauss-Seidel

$$x^{(k)} = (1 - \omega)x^{(k-1)} + \omega x_{GS}^{(k)}$$

- ▶ Folosind acum formula pe componente pentru metoda Gauss-Seidel, se obține următoarea expresie pe componente pentru metoda relaxării

$$x_i^{(k)} = (1 - \omega)x_i^{(k-1)} + \frac{\omega}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^m a_{ij}x_j^{(k-1)} \right).$$

Analiză matricială  
Norme matriciale  
Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar  
Estimarea numărului de  
condiționare  
Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile  
Introducere  
Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă  
Metoda lui Jacobi  
Metoda Gauss-Seidel  
Metoda relaxării

# Convergența metodei relaxării

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

## Teoremă (Kahan)

Dacă  $a_{ii} \neq 0$ ,  $i = 1, \dots, n$ ,  $\rho(T_\omega) < |\omega - 1|$ . De aici rezultă că  $\rho(T_\omega) < 1 \implies 0 < \omega < 2$  (condiție necesară).

## Teoremă (Ostrowski-Reich)

Dacă  $A$  este o matrice pozitiv definită și  $0 < \omega < 2$ , atunci SOR converge pentru orice alegere a aproximăției inițiale  $x^{(0)}$ .

Valoarea optimă a parametrului relaxării este

$$\omega_O = \frac{2}{1 + \sqrt{1 - (\rho(T_J))^2}}.$$

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Conditionarea unui  
sistem liniar

Conditionarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Convergența metodelor Jacobi și Gauss-Seidel

- ▶ Condiția necesară și suficientă de convergență pentru o metodă iterativă staționară este

$$\rho(T) < 1$$

- ▶ O condiție suficientă este:  $\|T\| < 1$ , pentru o anumită normă
- ▶ Pentru metoda lui Jacobi (și Gauss-Seidel) avem următoarele două condiții suficiente de convergență

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^m |a_{ij}|$$

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^m |a_{ji}|$$

(diagonal dominantă pe linii și respectiv pe coloane)

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Conditionarea unui  
sistem liniar

Conditionarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Bibliografie I

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimare numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

-  Octavian Agratini, Ioana Chiorean, Gheorghe Coman, Trîmbițaș Radu, *Analiză numerică și teoria aproximării*, vol. III, Presa Universitară Clujeană, 2002, coordonatori D. D. Stancu și Gh. Coman.
-  R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd ed., SIAM, Philadelphia, PA, 1994, disponibila prin www, <http://www.netlib.org/templates>.
-  James Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
-  H. H. Goldstine, J. von Neumann, *Numerical inverting of matrices of high order*, Amer. Math. Soc. Bull. **53** (1947), 1021–1099.

# Bibliografie II

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

-  Gene H. Golub, Charles van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore and London, 1996.
-  C. G. J. Jacobi, *Über eine neue Auflösungsart der bei der Methode der kleinsten Quadrate vorkommenden linearen Gleichungen*, Astronomische Nachrichten **22** (1845), 9–12, Issue no. 523.
-  W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, New York, Port Chester, Melbourne, Sidney, 1996, disponibila prin www, <http://www.nr.com/>.

Analiză matricială

Norme matriciale

Exemple de norme matriciale

Conditionarea unui sistem liniar

Conditionarea unui sistem liniar

Estimarea numărului de condiționare

Exemple de matrice prost condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

# Bibliografie III

Analiză matricială  
și condiționarea  
unui sistem liniar

Radu Trîmbițaș

Analiză matricială

Norme matriciale

Exemple de norme  
matriciale

Condiționarea unui  
sistem liniar

Condiționarea unui sistem  
liniar

Estimarea numărului de  
condiționare

Exemple de matrice prost  
condiționate

Metode iterative

Rezultate utile

Introducere

Convergența și delimitarea  
erorii

Metode concrete

Rafinarea iterativă

Metoda lui Jacobi

Metoda Gauss-Seidel

Metoda relaxării

 Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996, disponibilă via www la adresa

<http://www-users.cs.umn.edu/~saad/books.html>.

 Lloyd N. Trefethen, David Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1996.

# Interpolare

## O metodă de aproximare

Radu T. Trîmbițaș

Universitatea „Babeș-Bolyai”

6 aprilie 2023

# Un spațiu util

- Pentru  $n \in \mathbb{N}^*$ , definim

$$H^n[a, b] = \{f : [a, b] \rightarrow \mathbb{R} : f \in C^{n-1}[a, b], \\ f^{(n-1)} \text{ absolut continuă pe } [a, b]\}. \quad (1)$$

- Orice funcție  $f \in H^n[a, b]$  admite o reprezentare de tip Taylor cu restul sub formă integrală

$$f(x) = \sum_{k=0}^{n-1} \frac{(x-a)^k}{k!} f^{(k)}(a) + \int_a^x \frac{(x-t)^{n-1}}{(n-1)!} f^{(n)}(t) dt. \quad (2)$$

- $H^n[a, b]$  este un spațiu liniar.
- Funcția  $f : I \rightarrow \mathbb{R}$ ,  $I$  interval, se numește **absolut continuă** pe  $I$  dacă  $\forall \varepsilon > 0 \exists \delta > 0$  astfel încât oricare ar fi un sistem finit de subintervale disjuncte ale lui  $I$   $\{(a_k, b_k)\}_{k=\overline{1,n}}$  cu proprietatea  $\sum_{k=1}^n (b_k - a_k) < \delta$  să avem

$$\sum_{k=1}^n |f(b_k) - f(a_k)| < \varepsilon.$$

# Teorema lui Peano I

- Teorema următoare, de o importanță deosebită în analiza numerică, este o teoremă de reprezentare a funcțiilor liniare reale, definite pe  $H^n[a, b]$ .
- Ea dă un procedeu general de obținere a erorii de aproximare.
- Funcția

$$z_+ = \begin{cases} z, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

se numește **parte pozitivă**, iar  $z_+^n$  se numește **putere trunchiată**.

# Teorema lui Peano II

## Teorema 1 (Peano)

Fie  $L$  o funcțională reală, continuă, definită pe  $H^n[a, b]$ . Dacă  $\text{Ker } L = \mathbb{P}_{n-1}$  atunci

$$Lf = \int_a^b K(t) f^{(n)}(t) dt, \quad (3)$$

unde

$$K(t) = \frac{1}{(n-1)!} L[(\cdot - t)_+^{n-1}] \quad (\text{nucleul lui Peano}). \quad (4)$$

## Teorema lui Peano - continuare I

**Demonstrație.**  $f$  admite o reprezentare de tip Taylor, cu restul în formă integrală

$$f(x) = (T_{n-1}f)(x) + (R_{n-1}f)(x)$$

unde

$$R_{n-1}(x) = \int_a^x \frac{(x-t)^{n-1}}{(n-1)!} f^{(n)}(t) dt = \frac{1}{(n-1)!} \int_a^b (x-t)_+^{n-1} f^{(n)}(t) dt$$

Aplicând  $L$  obținem

$$\begin{aligned} Lf &= \underbrace{LT_{n-1}}_0 + LR_{n-1} \Rightarrow Lf = \frac{1}{(n-1)!} L \left( \int_a^b (\cdot - t)_+^{n-1} f^{(n)}(t) dt \right) = \\ &\stackrel{\text{cont}}{=} \frac{1}{(n-1)!} \int_a^b L(\cdot - t)_+^{n-1} f^{(n)}(t) dt. \end{aligned}$$

# Teorema lui Peano - continuare II

## Observația 2

Concluzia teoremei rămâne valabilă și dacă  $f$  nu este continuă, ci are forma

$$Lf = \sum_{i=0}^{n-1} \int_a^b f^{(i)}(x) d\mu_i(x), \quad \mu_i \in BV[a, b].$$

## Corolarul 3

Dacă  $K$  păstrează semn constant pe  $[a, b]$  și  $f^{(n)}$  este continuă pe  $[a, b]$ , atunci există  $\xi \in [a, b]$  astfel încât

$$Lf = \frac{1}{n!} f^{(n)}(\xi) L e_n, \tag{5}$$

unde  $e_k(x) = x^k$ ,  $k \in \mathbb{N}$ .

## Teorema lui Peano - continuare III

**Demonstrație.** Deoarece  $K$  păstrează semn constant putem aplica în (3) teorema de medie

$$Lf = f^{(n)}(\xi) \int_a^b K_n(t) dt, \quad \xi \in [a, b].$$

Concluzia se obține luând  $f = e_n$ . ■

Acest corolar este folosit în aplicații.



Figura: Giuseppe Peano (1858-1932)

# Exemplu I

## Exemplul 4

Vom folosi teorema lui Peano pentru a da expresia erorii în formula

$$\int_a^b f(x) dx = (b-a)f\left(\frac{a+b}{2}\right) + R(f)$$

numită *formula dreptunghiului*.

**Soluție.** Funcționala de reprezentat este

$$R(f) = \int_a^b f(x) dx - (b-a)f\left(\frac{a+b}{2}\right).$$

## Exemplu II

Ea se anulează pentru  $f(x) = 1$  și  $f(x) = x$ , deci  $\text{Ker}(R) = \mathbb{P}_1$ . Din teorema lui Peano rezultă

$$R(f) = \int_a^b K(t)f''(t)dt$$

unde

$$\begin{aligned} K(t) &= R\left(\frac{(x-t)_+}{1!}\right) = \int_a^b (x-t)_+ dt - (b-a)\left(\frac{a+b}{2} - t\right)_+ \\ &= (b-t)^2/2 - (b-a)\left(\frac{a+b}{2} - t\right)_+ \end{aligned}$$

Nucleul păstrează semn constant

$$K(f) = \begin{cases} \frac{(b-t)^2}{2}, & \text{dacă } t > (a+b)/2; \\ \frac{(t-a)^2}{2}, & \text{dacă } t \leq (a+b)/2. \end{cases}$$

### Exemplu III

Putem aplica corolarul la teorema lui Peano

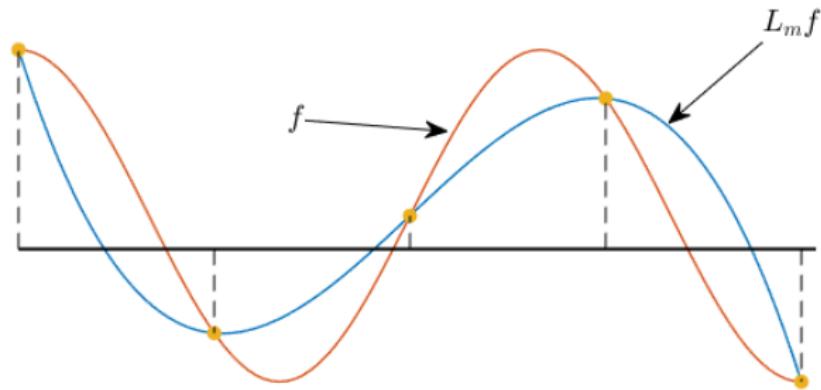
$$R(f) = \frac{f''(\xi)}{2!} R(e_2), \quad e_2(x) = x^2$$

și obținem

$$\begin{aligned} R(f) &= \frac{f''(\xi)}{2!} \left[ \int_a^b x^2 dx - (b-a) \left( \frac{a+b}{2} \right)^2 \right] \\ &= \frac{f''(\xi)}{2!} \left[ \frac{b^3 - a^3}{3} - (b-a) \left( \frac{a+b}{2} \right)^2 \right] \\ &= \frac{f''(\xi)}{2!} \left[ \frac{1}{12} b^3 - \frac{1}{12} a^3 + \frac{1}{4} b a^2 - \frac{1}{4} b^2 a \right] \\ &= \frac{(b-a)^3}{24} f''(\xi). \end{aligned}$$

# Interpolare Lagrange

Fie intervalul închis  $[a, b] \subset \mathbb{R}$ , o mulțime de  $m + 1$  puncte distincte  $\{x_0, x_1, \dots, x_m\} \subset [a, b]$  și o funcție  $f : [a, b] \mapsto \mathbb{R}$ . Dorim să determinăm un polinom  $P$ , **de grad minim** care să reproducă valorile funcției  $f$  în  $x_k$ , adică  $P(x_k) = f(x_k)$ ,  $k = \overline{0, m}$ .



# Interpolare Lagrange I

## Teorema 5

Există un polinom și numai unul  $L_m f \in \mathbb{P}_m$  astfel încât

$$\forall i = 0, 1, \dots, m, \quad (L_m f)(x_i) = f(x_i); \quad (6)$$

acest polinom se scrie sub forma

$$(L_m f)(x) = \sum_{i=0}^m f(x_i) \ell_i(x), \quad (7)$$

unde

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^m \frac{x - x_j}{x_i - x_j}. \quad (8)$$

# Interpolare Lagrange II

**Demonstrație.** Se verifică imediat că  $\ell_i \in \mathbb{P}_m$  și că  $\ell_i(x_j) = \delta_{ij}$  (simbolul lui Kronecker); rezultă că polinomul  $L_m f$  definit de (7) este de grad cel mult  $m$  și verifică (6).

Presupunem că există un alt polinom  $p_m^* \in \mathbb{P}_m$  care verifică (6) și punem  $q_m = L_m - p_m^*$ ; avem  $q_m \in \mathbb{P}_m$  și  $\forall i = 0, 1, \dots, m$ ,  $q_m(x_i) = 0$ ; deci  $q_m$  având  $(m+1)$  rădăcini distincte este identic nul, de unde unicitatea lui  $L_m$ . ■

## Definiția 6

*Polinomul  $L_m f$  definit astfel se numește **polinom de interpolare Lagrange** a lui  $f$  relativ la punctele  $x_0, x_1, \dots, x_m$ , iar funcțiile  $\ell_i(x)$ ,  $i = \overline{0, m}$ , se numesc **polinoame de bază (fundamentale) Lagrange** asociate acelor puncte.*

# Interpolare Lagrange III

## Observația 7

*Polinomul fundamental  $\ell_i$  este deci unicul polinom care verifică*

$$\ell_i \in \mathbb{P}_m \text{ și } \forall j = 0, 1, \dots, m, \quad \ell_i(x_j) = \delta_{ij}$$

*Punând*

$$u(x) = \prod_{j=0}^m (x - x_j)$$

*din (8) se deduce că  $\forall x \neq x_i, \quad \ell_i(x) = \frac{u(x)}{(x-x_i)u'(x_i)}$ .*

- Demonstrând teorema 5 am demonstrat de fapt existența și unicitatea soluției problemei generale de interpolare Lagrange:
- **[PGIL]** Fiind date  $b_0, b_1, \dots, b_m \in \mathbb{R}$ , să se determine

$$p_m \in \mathbb{P}_m \text{ astfel încât } \forall i = 0, 1, \dots, m, \quad p_m(x_i) = b_i. \quad (9)$$

# Interpolare Lagrange IV

- Problema (9) conduce la un sistem liniar de  $(m + 1)$  ecuații cu  $(m + 1)$  necunoscute (coeficienții lui  $p_m$ ).
  - Din teoria sistemelor liniare se știe că
- $\{\text{existența unei soluții } \forall b_0, b_1, \dots, b_m\} \Leftrightarrow \{\text{unicitatea soluției}\} \Leftrightarrow$

$$\{(b_0 = b_1 = \dots = b_m = 0) \Rightarrow p_m \equiv 0\}$$

- Punem  $p_m = a_0 + a_1x + \dots + a_mx^m$

$$a = (a_0, a_1, \dots, a_m)^T, \quad b = (b_0, b_1, \dots, b_m)^T$$

și notăm cu  $V = (v_{ij})$  matricea pătratică de ordin  $m + 1$  cu elementele  $v_{ij} = x_i^j$ . Ecuația (9) se scrie sub forma

$$Va = b$$

# Interpolare Lagrange V

- Matricea  $V$  este inversabilă (este Vandermonde); se arată ușor că  $V^{-1} = U^T$  unde  $U = (u_{ij})$  cu  $\ell_i(x) = \sum_{k=0}^m u_{ik}x^k$ ; se obține în acest mod un procedeu puțin costisitor de inversare a matricei Vandermonde și prin urmare și de rezolvare a sistemului (9).

# Exemple de PIL I

## Exemplul 8

*Polinomul de interpolare Lagrange corespunzător unei funcții  $f$  și nodurilor  $x_0$  și  $x_1$  este*

$$(L_1 f)(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1),$$

*adică dreapta care trece prin punctele  $(x_0, f(x_0))$  și  $(x_1, f(x_1))$ .*

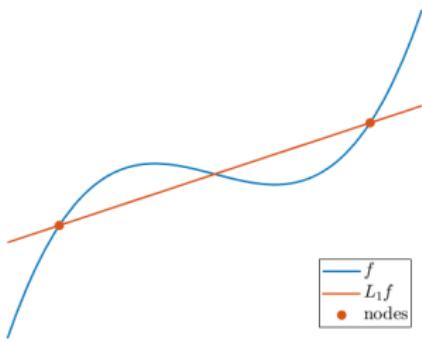
## Exemple de PIL II

### Exemplul 9

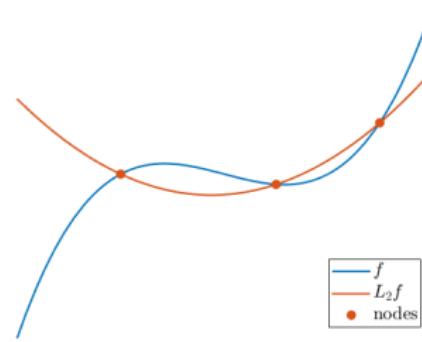
Analog, polinomul de interpolare Lagrange corespunzător unei funcții  $f$  și nodurilor  $x_0, x_1$  și  $x_2$  este

$$\begin{aligned}(L_2 f)(x) = & \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) \\ & + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2),\end{aligned}$$

adică parabola care trece prin punctele  $(x_0, f(x_0)), (x_1, f(x_1))$  și  $(x_2, f(x_2))$ .



$L_1 f$



$L_2 f$

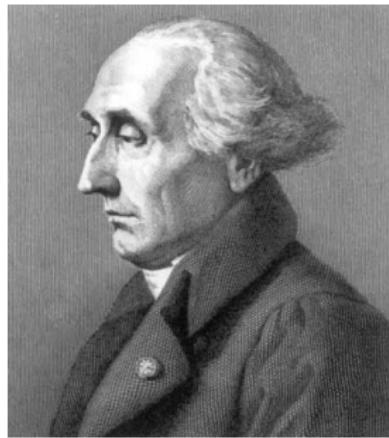


Figura: Joseph Louis Lagrange (1736-1813)

# Expresia erorii de interpolare

## Propoziția 10

*Operatorul  $L_m$  este projector, adică*

- este liniar ( $L_m(\alpha f + \beta g) = \alpha L_m f + \beta L_m g$ );
- este idempotent ( $L_m \circ L_m = L_m$ ).

**Demonstrație.** Liniaritatea rezultă imediat din formula (7). Datorită unicării polinomului de interpolare Lagrange  $L_m(L_m f) - L_m f$  este identic nul, deci  $L_m(L_m f) = L_m f$  și am arătat idempotența. ■

# Expresia erorii de interpolare

- Dacă dorim să utilizăm polinomul de interpolare Lagrange pentru a aproxima funcția  $f$  într-un punct  $x \in [a, b]$ , distinct de nodurile de interpolare  $(x_0, \dots, x_m)$ , trebuie să estimăm eroarea comisă  $(R_m f)(x) = f(x) - (L_m f)(x)$ .
- Dacă nu posedăm nici o informație referitoare la  $f$  în afara punctelor  $x_i$ , este clar că nu putem spune nimic despre  $(R_m f)(x)$ ; însă este posibil să schimbăm  $f$  în afara punctelor  $x_i$  fără a modifica  $(L_m f)(x)$ .
- Trebuie deci să facem ipoteze suplimentare, care vor fi ipoteze de regularitate asupra lui  $f$ . Să notăm cu  $C^m[a, b]$  spațiul funcțiilor reale de  $m$  ori continuu diferențiabile pe  $[a, b]$ .

Avem următoarea teoremă referitoare la estimarea erorii în interpolarea Lagrange.

# Expresia erorii de interpolare I

## Teorema 11

Presupunem că  $f \in C^m[\alpha, \beta]$  și există  $f^{(m+1)}$  pe  $(\alpha, \beta)$ , unde  $\alpha = \min\{x, x_0, \dots, x_m\}$  și  $\beta = \max\{x, x_0, \dots, x_m\}$ ; atunci, pentru orice  $x \in [\alpha, \beta]$ , există un  $\xi_x \in (\alpha, \beta)$  astfel încât

$$(R_m f)(x) = \frac{1}{(m+1)!} u_m(x) f^{(m+1)}(\xi_x), \quad (10)$$

unde

$$u_m(x) = \prod_{i=0}^m (x - x_i).$$

## Expresia erorii de interpolare II

**Demonstrație.** Dacă  $x = x_i$ ,  $(R_m f)(x) = 0$  și (10) se verifică trivial.

Presupunem că  $x$  este distinct de  $x_i$  și considerăm, pentru  $x$  fixat, funcția auxiliară

$$F(z) = \begin{vmatrix} u_m(z) & (R_m f)(z) \\ u_m(x) & (R_m f)(x) \end{vmatrix}.$$

Se observă că  $F \in C^m[\alpha, \beta]$ ,  $\exists F^{(m+1)}$  pe  $(\alpha, \beta)$ ,  $F(x) = 0$  și  $F(x_k) = 0$  pentru  $k = \overline{0, m}$ . Deci,  $F$  are  $(m+2)$  zerouri. Aplicând succesiv teorema lui Rolle, rezultă că există cel puțin un  $\xi \in (\alpha, \beta)$  astfel încât  $F^{(m+1)}(\xi) = 0$ , adică

$$F^{(m+1)}(\xi) = \begin{vmatrix} (m+1)! & f^{(m+1)}(\xi) \\ u_m(x) & (R_m f)(x) \end{vmatrix} = 0, \quad (11)$$

unde s-a ținut cont că  $(R_m f)^{(m+1)} = f^{(m+1)} - (L_m f)^{(m+1)} = f^{(m+1)}$ . Exprimând  $(R_m f)(x)$  din (11) se obține (10). ■

# Expresia erorii de interpolare III

## Corolarul 12

Punem  $M_{m+1} = \max_{x \in [a,b]} |f^{(m+1)}(x)| = \|f^{(m+1)}\|_\infty$ ; o margine superioară a erorii de interpolare  $(R_m f)(x) = f(x) - (L_m f)(x)$  este dată prin

$$|(R_m f)(x)| \leq \frac{M_{m+1}}{(m+1)!} |u_m(x)|.$$

Acest corolar ne permite să delimităm eroarea sau să obținem  $m$  dacă se impune o margine superioară a erorii.

## Expresia erorii de interpolare IV

Deoarece  $L_m$  este projector, rezultă că  $R_m$  este de asemenea projector; în plus  $\text{Ker } R_m = \mathbb{P}_m$ , deoarece  $R_m f = f - L_m f = f - f = 0, \forall f \in \mathbb{P}_m$ . Deci, putem aplica lui  $R_m$  teorema lui Peano.

### Teorema 13

Dacă  $f \in C^{m+1}[a, b]$ , atunci

$$(R_m f)(x) = \int_a^b K_m(x; t) f^{(m+1)}(t) dt \quad (12)$$

unde

$$K_m(x; t) = \frac{1}{m!} \left[ (x-t)_+^m - \sum_{k=0}^m \ell_k(x) (x_k - t)_+^m \right]. \quad (13)$$

# Expresia erorii de interpolare V

**Demonstrație.** Aplicând teorema lui Peano, avem

$$(R_m f)(x) = \int_a^b K_m(x; t) f^{(m+1)}(t) dt$$

și ținând cont că

$$K_m(x; t) = R_m \left[ \frac{(x-t)_+^m}{m!} \right] = \frac{(x-t)_+^m}{m!} - L_m \left[ \frac{(x-t)_+^m}{m!} \right],$$

teorema rezultă imediat. ■

# Exemplu

## Exemplul 14

Pentru polinoamele de interpolare din exemplul 8 resturile corespunzătoare sunt

$$(R_1 f)(x) = \frac{(x - x_0)(x - x_1)}{2} f''(\xi)$$

și respectiv

$$(R_2 f)(x) = \frac{(x - x_0)(x - x_1)(x - x_2)}{6} f'''(\xi).$$

# Eroarea pentru noduri Cebîșev I

- Fiind date funcția  $f$  și gradul  $m$  al polinomului de interpolare, cum trebuie alese nodurile astfel ca restul să fie cât mai mic posibil?

$$(R_m f)(x) = \frac{u_m(x)}{(m+1)!} f^{(m+1)}(\xi).$$

- Deoarece

$$|(R_m f)(x)| \leq \frac{\|u_m\|_\infty}{(m+1)!} \|f^{(m+1)}\|_\infty,$$

nodurile trebuie alese astfel ca  $\|u_m\|_\infty$  să fie minimă. Rezultă că  $u_m$  trebuie să fie polinomul monic Cebîșev de speță I de grad  $m+1$ ,

$$\mathring{T}_{m+1}$$

- În acest caz

$$\|R_m f\| \leq \frac{\|f^{(m+1)}\|_\infty}{2^m (m+1)!}.$$

# Metode de tip Aitken I

- În multe situații gradul necesar pentru a atinge precizia dorită în interpolarea polinomială este necunoscut.
- El se poate determina din expresia restului, dar pentru aceasta este necesar să cunoaștem  $\|f^{(m+1)}\|_\infty$ .
- $P_{m_1, m_2, \dots, m_k}$  — polinomul de interpolare Lagrange având nodurile  $x_{m_1}, \dots, x_{m_k}$ .

## Propoziția 15

Dacă  $f$  este definită în  $x_0, \dots, x_k$ ,  $x_j \neq x_i$ ,  $0 \leq i, j \leq k$ , atunci

$$\begin{aligned} P_{0,1,\dots,k}(x) &= \frac{(x - x_j)P_{0,1,\dots,j-1,j+1,\dots,k}(x) - (x - x_i)P_{0,1,\dots,i-1,i+1,\dots,k}(x)}{x_i - x_j} \\ &= \frac{1}{x_i - x_j} \left| \begin{array}{cc} x - x_j & P_{0,1,\dots,i-1,i+1,\dots,k}(x) \\ x - x_i & P_{0,1,\dots,j-1,j+1,\dots,k}(x) \end{array} \right| \end{aligned} \quad (14)$$

## Metode de tip Aitken II

**Demonstrație.**  $Q = P_{0,1,\dots,i-1,i+1,\dots,k}$ ,  $\widehat{Q} = P_{0,1,\dots,j-1,j+1,k}$

$$P(x) = \frac{(x - x_j)\widehat{Q}(x) - (x - x_i)Q(x)}{x_i - x_j}$$

$$P(x_r) = \frac{(x_r - x_j)\widehat{Q}(x_r) - (x_r - x_i)Q(x_r)}{x_i - x_j} = \frac{x_i - x_j}{x_i - x_j} f(x_r) = f(x_r)$$

pentru  $r \neq i \wedge r \neq j$ , căci  $Q(x_r) = \widehat{Q}(x_r) = f(x_r)$ . Dar

$$P(x_i) = \frac{(x_i - x_j)\widehat{Q}(x_i) - (x_i - x_i)Q(x_i)}{x_i - x_j} = f(x_i)$$

și

$$P(x_j) = \frac{(x_j - x_j)\widehat{Q}(x_j) - (x_j - x_i)Q(x_j)}{x_i - x_j} = f(x_j),$$

deci  $P = P_{0,1,\dots,k}$ . ■

## Metode de tip Aitken III

- am stabilit o relație de recurență între un polinom de interpolare Lagrange de gradul  $k$  și două polinoame de interpolare Lagrange de gradul  $k - 1$ .
- Calculele pot fi așezate în formă tabelară

$x_0$	$P_0$				
$x_1$	$P_1$	$P_{0,1}$			
$x_2$	$P_2$	$P_{1,2}$	$P_{0,1,2}$		
$x_3$	$P_3$	$P_{2,3}$	$P_{1,2,3}$	$P_{0,1,2,3}$	
$x_4$	$P_4$	$P_{3,4}$	$P_{2,3,4}$	$P_{1,2,3,4}$	$P_{0,1,2,3,4}$

- Dacă  $P_{0,1,2,3,4}$  nu ne asigură precizia dorită, se poate selecta un nou nod și adăuga o nouă linie tabelei

$x_5 \quad P_5 \quad P_{4,5} \quad P_{3,4,5} \quad P_{2,3,4,5} \quad P_{1,2,3,4,5} \quad P_{0,1,2,3,4,5}$

# Metode de tip Aitken IV

- **Criteriu de oprire:** elementele vecine de pe linie, coloană sau diagonală se pot compara pentru a vedea dacă s-a obținut precizia dorită.
- metoda lui Neville
- Notațiile pot fi simplificate

$$Q_{i,j} := P_{i-j, i-j+1, \dots, i-1, i},$$

$$Q_{i,j-1} := P_{i-j+1, \dots, i-1, i},$$

$$Q_{i-1,j-1} := P_{i-j, i-j+1, \dots, i-1}.$$

- Din (14) rezultă, pentru  $j = 1, 2, 3, \dots, i = j+1, j+2, \dots,$

$$Q_{i,j} = \frac{(x - x_{i-j}) Q_{i,j-1} - (x - x_i) Q_{i-1,j-1}}{x_i - x_{i-j}}.$$

## Metode de tip Aitken V

- În plus,  $Q_{i,0} = f(x_i)$ . Obținem tabelul

$x_0$	$Q_{0,0}$			
$x_1$	$Q_{1,0}$	$Q_{1,1}$		
$x_2$	$Q_{2,0}$	$Q_{2,1}$	$Q_{2,2}$	
$x_3$	$Q_{3,0}$	$Q_{3,1}$	$Q_{3,2}$	$Q_{3,3}$

- Dacă procedeul de interpolare converge, atunci sirul  $Q_{i,i}$  converge și el și s-ar putea lua drept criteriu de oprire

$$|Q_{i,i} - Q_{i-1,i-1}| < \varepsilon.$$

- Pentru a rapidiza algoritmul nodurile se vor ordona crescător după valorile  $|x_i - x|$ .
- **Metoda lui Aitken** este similară cu metoda lui Neville.

# Metode de tip Aitken VI

- Ea construiește tabelul

$x_0$	$P_0$				
$x_1$	$P_1$	$P_{0,1}$			
$x_2$	$P_2$	$P_{0,2}$	$P_{0,1,2}$		
$x_3$	$P_3$	$P_{0,3}$	$P_{0,1,3}$	$P_{0,1,2,3}$	
$x_4$	$P_4$	$P_{0,4}$	$P_{0,1,4}$	$P_{0,1,2,4}$	$P_{0,1,2,3,4}$

- Pentru a calcula o nouă valoare se utilizează valoarea din vârful coloanei precedente și valoarea din aceeași linie, coloana precedentă.

# Metoda diferențelor divizate I

- Vom nota cu  $L_k f$  PIL cu nodurile  $x_0, x_1, \dots, x_k$  pentru  $k = 0, 1, \dots, n$ . Vom construi  $L_m$  prin recurență.
- Avem

$$(L_0 f)(x) = f(x_0)$$

pentru  $k \geq 1$

- polinomul  $L_k - L_{k-1}$  este de grad  $k$ , se anulează în punctele  $x_0, x_1, \dots, x_{k-1}$  și deci este de forma:

$$(L_k f)(x) - (L_{k-1} f)(x) = f[x_0, x_1, \dots, x_k] (x - x_0)(x - x_1) \dots (x - x_{k-1}) \quad (15)$$

unde  $f[x_0, x_1, \dots, x_k]$  desemnează coeficientul lui  $x^k$  din  $(L_k f)(x)$ .

## Metoda diferențelor divizate II

- Se deduce expresia polinomului de interpolare  $L_m f$  cu nodurile  $x_0, x_1, \dots, x_n$

$$(L_m f)(x) = f(x_0) + \sum_{k=1}^m f[x_0, x_1, \dots, x_k] (x - x_0)(x - x_1) \dots (x - x_{k-1}), \quad (16)$$

- forma Newton a polinomului de interpolare Lagrange
- Formula (16) reduce calculul prin recurență al lui  $L_m f$  la cel al coeficienților  $f[x_0, x_1, \dots, x_k]$ ,  $k = \overline{0, m}$ .

# Recurență pentru diferențe divizate

## Lema 16

$$\forall k \geq 1 \quad f[x_0, x_1, \dots, x_k] = \frac{f[x_1, x_2, \dots, x_k] - f[x_0, x_1, \dots, x_{k-1}]}{x_k - x_0} \quad (17)$$

și

$$f[x_i] = f(x_i), \quad i = 0, 1, \dots, k.$$

**Demonstrație.** Notăm, pentru  $k \geq 1$  cu  $L_{k-1}^* f$  polinomul de interpolare pentru  $f$  de grad  $k-1$  și cu nodurile  $x_1, x_2, \dots, x_k$ ; coeficientul lui  $x^{k-1}$  este  $f[x_1, x_2, \dots, x_k]$ . Polinomul  $q_k$  de grad  $k$  definit prin

$$q_k(x) = \frac{(x - x_0)(L_{k-1}^* f)(x) - (x - x_k)(L_{k-1} f)(x)}{x_k - x_0}$$

coincide cu  $f$  în punctele  $x_0, x_1, \dots, x_k$  și deci  $q_k(x) \equiv (L_k f)(x)$ . Formula (17) se obține identificând coeficientul lui  $x^k$  din cei doi membri. ■

# Calculul PIL cu diferențe divizate I

## Definiția 17

Cantitatea  $f[x_0, x_1, \dots, x_k]$  se numește **diferență divizată de ordinul  $k$**  a lui  $f$  în punctele  $x_0, x_1, \dots, x_k$ .

Altă notație utilizată este  $[x_0, \dots, x_k; f]$ .

Din definiție rezultă că  $f[x_0, x_1, \dots, x_k]$  este independentă de ordinea punctelor  $x_i$  și ea poate fi calculată în funcție de  $f(x_0), \dots, f(x_m)$ . Într-adevăr PIL de grad  $\leq m$  relativ la punctele  $x_0, \dots, x_m$  se scrie

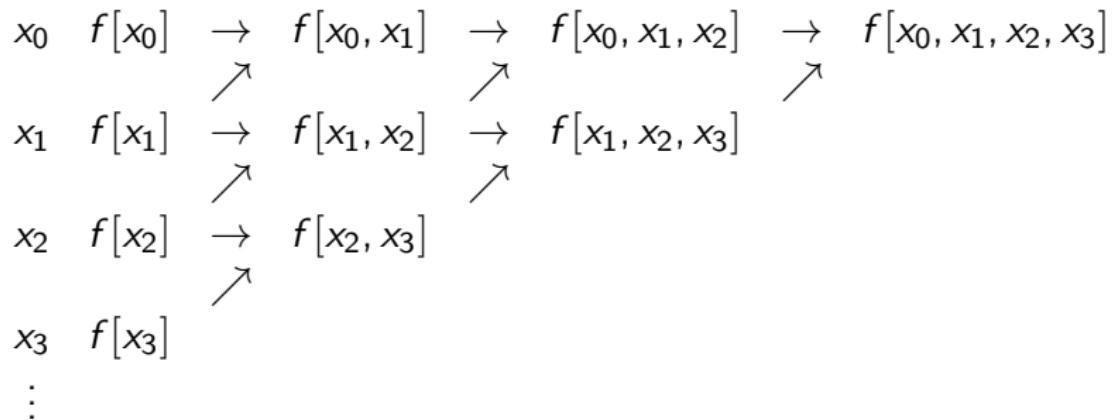
$$(L_m f)(x) = \sum_{i=0}^m \ell_i(x) f(x_i)$$

și coeficientul lui  $x^m$  este

$$f[x_0, \dots, x_m] = \sum_{i=0}^m \frac{f(x_i)}{\prod_{\substack{j=0 \\ j \neq i}}^m (x_i - x_j)}. \quad (18)$$

## Calculul PIL cu diferențe divizate II

- Diferențele divizate se pot obține prin algoritmul tabelar următor, bazat pe formula (17), care este mai flexibil și mai puțin costisitor decât aplicarea formulei (18)



$$Q_{i,j} = \frac{Q_{i+1,j-1} - Q_{i,j-1}}{x_{i+j} - x_i}, \quad j = 1, \dots, m, \quad i = 0, \dots, m-j.$$

# Calculul PIL cu diferențe divizate III

- Prima coloană este formată din valorile funcției  $f$ , a doua din diferențele divizate de ordinul I, etc.; se trece la coloana următoare folosind formula (17).
- Pentru a rapidiza algoritmul nodurile se vor ordona crescător după valorile  $|x_i - x|$ .

## Exemplul 18

Să calculăm forma Newton a polinomului de interpolare Lagrange pentru funcția  $f(x) = x^3$  și nodurile  $x_k = k$ ,  $k = 0, \dots, 3$ . Tabela diferențelor divizate este:

$x$	$f[x_i]$	$f[x_i, x_j]$	$f[x_i, x_j, x_k]$	$f[x_i, x_j, x_k, x_l]$
0	0	1	3	1
1	1	7	6	
2	8	19		
3	27			

La calculul polinomului de interpolare se folosesc diferențele divizate din prima linie a tabelei.

$$\begin{aligned}(L_3 f)(x) &= f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \\&\quad + (x - x_0)(x - x_1)(x - x_2)f[x_0, x_1, x_2, x_3] \\&= x + 3x(x - 1) + x(x - 1)(x - 2);\end{aligned}$$

# Proprietăți ale diferențelor divizate I

## Teorema 19

Eroarea de interpolare este dată de

$$f(x) - (L_m f)(x) = u_m(x) f[x_0, x_1, \dots, x_m, x]. \quad (19)$$

**Demonstrație.** Într-adevăr, este suficient să observăm că

$$(L_m f)(t) + u_m(t) f[x_0, \dots, x_m; x]$$

este conform lui (16) polinomul de interpolare (în  $t$ ) al lui  $f$  în punctele  $x_0, x_1, \dots, x_m, x$ . ■

## Proprietăți ale diferențelor divizate II

Teorema 20 (formula de medie pentru diferențe divizate)

Dacă  $f \in C^m[a, b]$ , există  $\xi \in (a, b)$  a.î.

$$f[x_0, x_1, \dots, x_m] = \frac{1}{m!} f^{(m)}(\xi) \quad (20)$$

**Demonstrație.** Rezultă din (10) și din (19) ■

## Proprietăți ale diferențelor divizate III

Teorema 21 (scrierea sub forma unui cât a doi determinanți)

Are loc

$$f[x_0, \dots, x_m] = \frac{(Wf)(x_0, \dots, x_m)}{V(x_0, \dots, x_m)} \quad (21)$$

unde

$$(Wf)(x_0, \dots, x_m) = \begin{vmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{m-1} & f(x_0) \\ 1 & x_1 & x_1^2 & \dots & x_1^{m-1} & f(x_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^{m-1} & f(x_m) \end{vmatrix}, \quad (22)$$

iar  $V(x_0, \dots, x_m)$  este determinantul Vandermonde.

## Proprietăți ale diferențelor divizate IV

**Demonstrație.** Se dezvoltă  $(Wf)(x_0, \dots, x_m)$  după elementele ultimei coloane; fiecare complement algebric este un determinant Vandermonde. Se obține

$$f[x_0, \dots, x_m] = \frac{1}{V(x_0, \dots, x_m)} \sum_{i=0}^m V(x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_m) f(x_i) =$$

$$= \sum_{i=0}^m (-1)^{m-i} \frac{f(x_i)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_n - x_i)},$$

din care după schimbarea semnelor ultimilor  $m - i$  termeni rezultă (18). ■



Figura: Sir Isaac Newton (1643 - 1727)

# Metoda baricentrică I

- Rescriem (7), (8) a.î. PIL să poată fi evaluat și actualizat cu  $O(m)$  operații. Avem

$$\ell_j(x) = \frac{u_m(x)}{\prod_{k \neq j} (x_j - x_k)} \cdot \frac{1}{x - x_j}, \quad (23)$$

unde

$$u_m(x) = (x - x_0)(x - x_1) \cdots (x - x_m) \quad (24)$$

- Definind ponderile baricentrice prin

$$w_j = \frac{1}{\prod_{k \neq j} (x_j - x_k)}, \quad j = 0, \dots, m, \quad (25)$$

adică,  $w_j = 1/u'_m(x_j)$  și putem scrie  $\ell_j$  sub forma

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

## Metoda baricentrică II

- Acum PIL se scrie ( $f_j := f(x_j)$ )

$$p(x) = u_m(x) \sum_{j=0}^m \frac{w_j}{x - x_j} f_j. \quad (26)$$

- Înlocuind funcția constantă 1 obținem

$$1 = \sum_{j=0}^m \ell_j(x) = u_m(x) \sum_{j=0}^m \frac{w_j}{x - x_j}. \quad (27)$$

- Împărțind (26) cu expresia de mai sus și simplificând cu  $u_m(x)$ , obținem

$$p(x) = \frac{\sum_{j=0}^m \frac{w_j}{x - x_j} f_j}{\sum_{j=0}^m \frac{w_j}{x - x_j}}, \quad (28)$$

numită **formula baricentrică**.

# Distribuții remarcabile I

- În cazul unor noduri particulare se pot da formule explicite pentru ponderile baricentrice  $w_j$ .
- Pentru noduri echidistante

$$w_j = (-1)^j \binom{m}{j}. \quad (29)$$

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

$$w_j = \frac{1}{u'_m(x_j)}, \quad (30)$$

se pot obține formule explicite pentru ponderile  $w_j$ .

## Distribuții remarcabile II

- Punctele Cebîșev de speță I sunt date de

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

Anulând factorii independenți de  $j$  se obține

$$w_j = (-1)^j \sin \frac{(2j+1)\pi}{2m+2}. \quad (31)$$

- Punctele Cebîșev de speță II sunt date de

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

iar ponderile corespunzătoare sunt

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

# Interpolarea în puncte Cebîșev I

- Dificultățile legate de interpolarea polinomială de grad mare pot fi depășite aglomerând punctele de interpolare la capătul intervalului în loc de a alege puncte echidistante
- Noduri: *puncte de interpolare Cebîșev* de speță a două sau puncte *Gauss-Lobatto* pe  $[-1, 1]$

$$x_j = \cos \frac{\pi j}{n}, \quad j = 0, \dots, n \quad (32)$$

- Pentru un interval  $[a, b]$  se face schimbarea de variabilă

$$t = \frac{2x - b - a}{b - a}$$

- Utilizate în pachetul MATLAB `chebfun` - Univ. Oxford, L. N. Trefethen

## Interpolarea în puncte Cebîșev II

- Dacă  $(x_j)_{j=0}^n$  sunt puncte Cebîșev, polinomul nodurilor satisface

$$\left| \prod_{j=0}^n (x - x_j) \right| \leq 2^{-n+1}$$

- Ponderile baricentrice au forma

$$w_j = \frac{2^{n-1}}{n} \begin{cases} (-1)^j / 2 & \text{dacă } j = 0 \text{ sau } j = n, \\ (-1)^j & \text{altfel,} \end{cases} \quad (33)$$

deci foarte convenabile la evaluare. Factorul  $2^{n-1}/n$  se poate elimina deoarece în formula baricentrică (28) apare și la numărător și la numitor

# Proprietăți ale interpolării în puncte Cebîșev

## Teorema 22

Fie  $f \in C[-1, 1]$ ,  $p_n$  polinomul său de interpolare în puncte Cebîșev (32) și  $p_n^*$  polinomul său de cea mai bună aproximare în norma  $\|\cdot\|_\infty$ . Atunci

- ①  $\|f - p_n\|_\infty \leq \left(2 + \frac{2}{\pi} \log n\right) \|f - p_n^*\|_\infty$
- ② Dacă  $\exists k \in \mathbb{N}^*$  a.î.  $f^{(k)}$  este cu variație mărginită pe  $[-1, 1]$ , atunci  $\|f - p_n\|_\infty = O(n^{-k})$ , când  $n \rightarrow \infty$ .
- ③ Dacă  $f$  este analitică într-o vecinătate din planul complex a lui  $[-1, 1]$ , atunci  $\exists C < 1$  a.î.  $\|f - p_n\|_\infty = O(C^n)$ ; în particular dacă  $f$  este analitică în elipsa închisă cu focarele  $\pm 1$  și semiaxele  $M \geq 1$  și  $m \geq 0$ , putem lua  $C = 1/(M+m)$ .

# Interpolare Hermite I

În loc să facem să coincidă  $f$  și polinomul de interpolare în punctele  $x_i$  din  $[a, b]$ , am putea face ca  $f$  și polinomul de interpolare să coincidă împreună cu derivatele lor până la ordinul  $r_i$  în punctele  $x_i$ . Se obține:

## Teorema 23

*Fie date  $(m + 1)$  puncte distincte  $x_0, x_1, \dots, x_m$  din  $[a, b]$  și  $(m + 1)$  numere naturale  $r_0, r_1, \dots, r_m$ , punem  $n = m + r_0 + r_1 + \dots + r_m$ . Atunci, fiind dată o funcție  $f$ , definită pe  $[a, b]$  și admitând derivate de ordin  $r_i$  în punctele  $x_i$  există un singur polinom și numai unul  $H_n f$  de grad  $\leq n$  astfel încât*

$$\forall (i, \ell), \quad 0 \leq i \leq m, \quad 0 \leq \ell \leq r_i \quad (H_n f)^{(\ell)}(x_i) = f^{(\ell)}(x_i), \quad (34)$$

*unde  $f^{(\ell)}(x_i)$  este derivata de ordinul  $\ell$  a lui  $f$  în  $x_i$ .*

## Interpolare Hermite II

### Definiția 24

*Polinomul definit în acest mod se numește **polinom de interpolare al lui Hermite** al funcției  $f$  relativ la punctele  $x_0, x_1, \dots, x_m$  și la întregii  $r_0, r_1, \dots, r_m$ .*

**Demonstrație.** Ecuația (34) conduce la un sistem liniar de  $(n + 1)$  ecuații cu  $(n + 1)$  necunoscute (coeficienții lui  $H_n f$ ), deci este suficient să arătăm că sistemul omogen corespunzător admite doar soluția nulă, adică relațiile

$$H_n f \in \mathbb{P}_n \text{ și } \forall (i, \ell), \quad 0 \leq i \leq k, \quad 0 \leq \ell \leq r_i, \quad (H_n f)^{(\ell)}(x_i) = 0$$

ne asigură că pentru orice  $i = 0, 1, \dots, m$ ,  $x_i$  este rădăcină de ordinul  $r_i + 1$  a lui  $H_n f$ ; prin urmare  $H_n f$  are forma

$$(H_n f)(x) = q(x) \prod_{i=0}^m (x - x_i)^{r_i+1},$$

# Interpolare Hermite III

unde  $q$  este un polinom. Cum  $\sum_{i=0}^m (r_i + 1) = n + 1$ , acest lucru nu este compatibil cu apartenența lui  $H_n$  la  $\mathbb{P}_n$ , decât dacă  $q \equiv 0$  și deci  $H_n \equiv 0$ .

■

# Diferențe divizate cu noduri multiple I

Formulele (20) și (21) servesc ca bază pentru introducerea diferenței divizate cu noduri multiple: dacă  $f \in C^m[a, b]$  și  $\alpha \in [a, b]$ , atunci

$$\lim_{x_0, \dots, x_m \rightarrow \alpha} [x_0, \dots, x_m; f] = \lim_{\xi \rightarrow \alpha} \frac{f^{(m)}(\xi)}{m!} = \frac{f^{(m)}(\alpha)}{m!}$$

Aceasta justifică relația

$$[\underbrace{\alpha, \dots, \alpha}_{m+1}; f] = \frac{1}{m!} f^{(m)}(\alpha).$$

Reprezentând aceasta ca pe un cât de doi determinanți se obține

$$(Wf) \left( \underbrace{\alpha, \dots, \alpha}_{m+1} \right) = \begin{vmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{m-1} & f(\alpha) \\ 0 & 1 & 2\alpha & \dots & (m-1)\alpha^{m-2} & f'(\alpha) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & (m-1)! & f^{(m-1)}(\alpha) \\ 0 & 0 & 0 & \dots & 0 & f^{(m)}(\alpha) \end{vmatrix}$$

## Diferențe divizate cu noduri multiple II

și

$$V \left( \underbrace{\alpha, \dots, \alpha}_{m+1} \right) = \begin{vmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^m \\ 0 & 1 & 2\alpha & \dots & m\alpha^{m-1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & m! \end{vmatrix},$$

adică cei doi determinanți sunt constituiți din linia relativă la nodul  $\alpha$  și derivatele succesive ale acesteia până la ordinul  $m$  în raport cu  $\alpha$ .

Generalizarea pentru mai multe noduri este următoarea: Fie  
 $r_k \in \mathbb{N}$ ,  $k = \overline{0, m}$ ,  $n = r_0 + \dots + r_m + m$ . Presupunem că există  
 $f^{(j)}(x_k)$ ,  $k = \overline{0, m}$ ,  $j = \overline{0, r_k}$ . Mărimea

$$[\underbrace{x_0, \dots, x_0}_{r_0+1}, \underbrace{x_1, \dots, x_1}_{r_1+1}, \dots, \underbrace{x_m, \dots, x_m}_{r_m+1}; f] = \frac{(Wf)(x_0, \dots, x_0, \dots, x_m, \dots, x_m)}{V(x_0, \dots, x_0, \dots, x_m, \dots, x_m)}$$

unde

$$(Wf)(x_0, \dots, x_0, \dots, x_m, \dots, x_m) =$$

$$= \begin{vmatrix} 1 & x_0 & \dots & x_0^{r_0} & \dots & x_0^{n-1} & f(x_0) \\ 0 & 1 & \dots & (r_0)x_0^{r_0-1} & \dots & (n-1)x_0^{n-2} & f'(x_0) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & (r_0)! & \dots & \prod_{p=1}^{r_0} (n-p)x_0^{n-r_0+1} & f^{(r_0)}(x_0) \\ 1 & x_m & \dots & x_m^{r_m} & \dots & x_m^{n-1} & f(x_m) \\ 0 & 1 & \dots & (r_m)x_m^{r_m-1} & \dots & (n-1)x_m^{n-2} & f'(x_m) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & (r_m)! & \dots & \prod_{p=1}^{r_m} (n-p)x_m^{n-r_m+1} & f^{(r_m)}(x_m) \end{vmatrix}$$

iar  $V(x_0, \dots, x_0, \dots, x_m, \dots, x_m)$  este ca mai sus, exceptând ultima coloană care este

$$(x_0^n, nx_0^{n-1}, \dots, \prod_{p=0}^{r_0} (n-p)x_0^{n-r_0+1}, \dots, x_m^n, nx_m^{n-1}, \dots, \prod_{p=0}^{r_m} (n-p)x_m^{n-r_m+1})^T$$

se numește diferență divizată cu nodurile multiple  $x_k$ ,  $k = \overline{0, m}$  și ordinele de multiplicitate  $r_k$ ,  $k = \overline{0, m}$ .

# Polinomul de interpolare Hermite I

Generalizând forma Newton a polinomului de interpolare Lagrange se obține o metodă bazată pe diferențele divizate cu noduri multiple pentru PIH.

Presupunem că se dau nodurile  $x_i$ ,  $i = \overline{0, m}$  și valorile  $f(x_i)$ ,  $f'(x_i)$ .

Definim secvența de noduri  $z_0, z_1, \dots, z_{2m+1}$  prin  $z_{2i} = z_{2i+1} = x_i$ ,  $i = \overline{0, m}$ . Construim acum tabela diferențelor divizate utilizând nodurile  $z_i$ ,  $i = \overline{0, 2m+1}$ . Deoarece  $z_{2i} = z_{2i+1} = x_i$  pentru orice  $i$ ,  $f[x_{2i}, x_{2i+1}]$  este o diferență divizată cu nod dublu și este egală cu  $f'(x_i)$ , deci vom utiliza  $f'(x_0), f'(x_1), \dots, f'(x_m)$  în locul diferențelor divizate de ordinul I

$$f[z_0, z_1], f[z_2, z_3], \dots, f[z_{2m}, z_{2m+1}].$$

Restul diferențelor se obțin în manieră obișnuită, aşa cum se arată în tabelul 1. Ideea poate fi extinsă și pentru alte interpolări Hermite. Se pare că metoda este datorată lui Powell.

## Polinomul de interpolare Hermite II

$$\begin{array}{llll} z_0 = x_0 & f[z_0] & f[z_0, z_1] = f'(x_0) & f[z_0, z_1, z_2] = \frac{f[z_1, z_2] - f[z_0, z_1]}{z_2 - z_0} \\ z_1 = x_0 & f[z_1] & f[z_1, z_2] = \frac{f(z_2) - f(z_1)}{z_2 - z_1} & f[z_1, z_2, z_3] = \frac{f[z_3, z_2] - f[z_2, z_1]}{z_3 - z_1} \\ z_2 = x_1 & f[z_2] & f[z_2, z_3] = f'(x_1) & f[z_2, z_3, z_4] = \frac{f[z_4, z_3] - f[z_3, z_2]}{z_4 - z_2} \\ z_3 = x_1 & f[z_3] & f[z_3, z_4] = \frac{f(z_4) - f(z_3)}{z_4 - z_3} \\ z_4 = x_2 & f[z_4] & f[z_4, z_5] = f'(x_2) \\ z_5 = x_2 & f[z_5] \end{array}$$

**Tabela:** Tabelă de diferențe divizate pentru noduri duble

# Expresia erorii

Folosind teorema de medie pentru diferențe divizate obținem următoarea expresie a erorii pentru interpolarea Hermite:

## Propoziția 25

Dacă  $f \in C^{n+1}[a, b]$  există  $\xi \in [a, b]$  astfel încât

$$(R_n f)(x) = \frac{u(x)}{(n+1)!} f^{(n+1)}(\xi), \quad (35)$$

unde

$$u(x) = (x - x_0)^{r_0+1} \dots (x - x_m)^{r_m+1} = \prod_{k=0}^m (x - x_k)^{r_k+1}.$$



Figura: Charles Hermite

Charles Hermite (1822-1901), matematician francez de frunte, membru al Academiei Franceze, cunoscut pentru lucrările sale în domeniul teoriei numerelor, algebră și analiză. A devenit faimos după ce a dat, în 1873, demonstrația transcendenței numărului  $e$ .

## Exemplul 26

Pentru  $f \in C^4[a, b]$ , să se calculeze polinomul de interpolare Hermite cu nodurile duble  $x_0 = a$  și  $x_1 = b$  și sa se dea expresia erorii de interpolare.

**Soluție.** Avem  $x_0 = a$ ,  $r_0 = 1$ ,  $x_1 = b$ ,  $r_1 = 1$  și  $m = 1$ . Gradul polinomului va fi  $n = 1 + r_0 + r_1 = 3$ .

Tabela diferențelor divizate este:

	$D^0$	$D^1$	$D^2$	$D^3$
$z_0 = a$	$f(a)$	$f'(a)$	$\frac{f(b) - f(a) - (b-a)f'(a)}{(b-a)^2}$	$\frac{(b-a)(f'(b) + f'(a)) - 2(f(b) - f(a))}{(b-a)^3}$
$z_1 = a$	$f(a)$	$\frac{f(b) - f(a)}{b-a}$	$\frac{(b-a)f'(b) - f(b) + f(a)}{(b-a)^2}$	
$z_2 = b$	$f(b)$	$f'(b)$		
$z_3 = b$	$f(b)$			

Polinomul de interpolare va fi

$$\begin{aligned}(H_3f)(x) &= f[z_0] + (x - z_0)f[z_0, z_1] + (x - z_0)(x - z_1)f[z_0, z_1, z_2] + \\&\quad (x - z_0)(x - z_1)(x - z_2)f[z_0, z_1, z_2, z_3] \\&= f(a) + (x - a)f'(a) + (x - a)^2 \frac{f(b) - f(a) - (b - a)f'(a)}{(b - a)^2} + \\&\quad (x - a)^2(x - b) \frac{(b - a)(f'(b) + f'(a)) - 2(f(b) - f(a))}{(b - a)^3}.\end{aligned}$$

Restul

$$(R_3f)(x) = \frac{(x - a)^2(x - b)^2}{4!} f^{(4)}(\xi).$$



# Bibliografie I

-  E. Blum, *Numerical Computing: Theory and Practice*, Addison-Wesley, 1972.
-  P. G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, Paris, Milan, Barcelone, Mexico, 1990.
-  Gheorghe Coman, *Analiză numerică*, Editura Libris, Cluj-Napoca, 1995.
-  W. Gautschi, *Numerical Analysis. An Introduction*, Birkhäuser, Basel, 1997.
-  W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, New York, Port Chester, Melbourne, Sidney, 1996, disponibila prin www, <http://www.nr.com/>.

## Bibliografie II

- D. D. Stancu, *Analiză numerică – Curs și culegere de probleme*, Lito UBB, Cluj-Napoca, 1977.
- J. Stoer, R. Burlisch, *Introduction to Numerical Analysis*, 2nd ed., Springer Verlag, 1992.
- R. Trîmbițaș, *Numerical Analysis in MATLAB*, Cluj University Press, 2010

# Interpolare spline

## Interpolare polinomială pe porțiuni

Radu Trîmbițaș

UBB

7 mai 2020

# Convergența interpolării polinomiale I

- Să definim ce înțelegem prin convergență.
- Presupunem că se dă un tablou triunghiular de noduri de interpolare  $x_i = x_i^{(m)}$ , având exact  $m + 1$  noduri distincte pentru orice  $m = 0, 1, 2, \dots$ .

$$\begin{matrix} & & x_0^{(0)} & & \\ & x_0^{(1)} & & x_1^{(1)} & \\ x_0^{(2)} & & x_1^{(2)} & & x_2^{(2)} \\ \vdots & \vdots & \vdots & & \ddots \\ x_0^{(m)} & x_1^{(m)} & x_2^{(m)} & \dots & x_m^{(m)} \\ \vdots & \vdots & \vdots & & \vdots \end{matrix} \quad (1)$$

- Presupunem că toate nodurile sunt conținute într-un interval finit  $[a, b]$ .

## Convergența interpolării polinomiale II

- Atunci pentru orice  $m$  definim

$$p_m(x) = L_m(f; x; x_0^{(m)}, x_1^{(m)}, \dots, x_m^{(m)}), \quad x \in [a, b]. \quad (2)$$

- Spunem că interpolarea Lagrange bazată pe tabelul de noduri (1) converge dacă

$$p_m(x) \rightrightarrows f(x), \text{ când } n \rightarrow \infty \text{ pe } [a, b]. \quad (3)$$

- În general, procedeul interpolării Lagrange diverge.

## Exemplul 1 (Contraexemplul lui Runge)

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5, 5],$$
$$x_k^{(m)} = -5 + 10 \frac{k}{m}, \quad k = \overline{0, m}. \quad (4)$$

Nodurile sunt echidistante pe  $[-5, 5]$ , deci asimptotic uniform distribuite. Observăm că  $f$  are doi poli în  $z = \pm i$ . Se poate demonstra că

$$\lim_{m \rightarrow \infty} |f(x) - p_m(f; x)| = \begin{cases} 0 & \text{dacă } |x| < 3.633\dots \\ \infty & \text{dacă } |x| > 3.633\dots \end{cases} \quad (5)$$

Graficul pentru  $m = 10, 13, 16$  apare în figura 1.

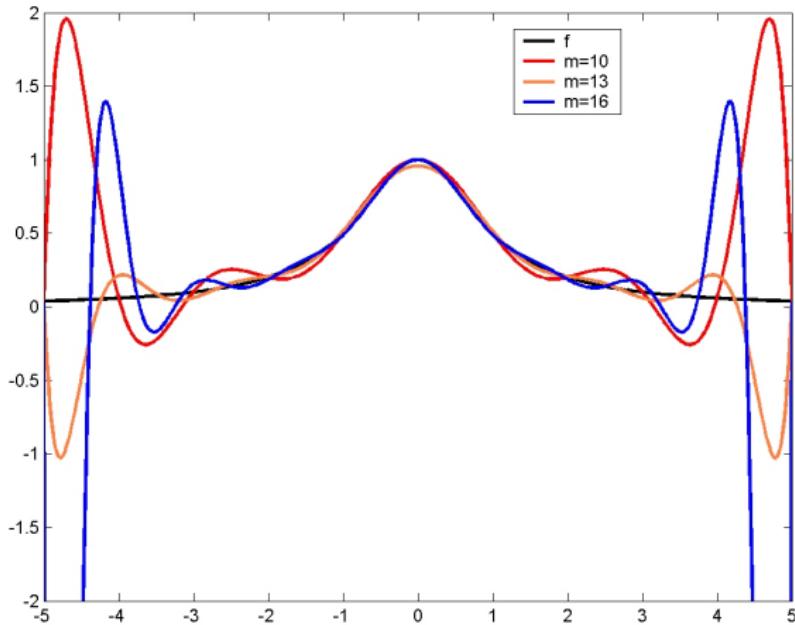


Figura: O ilustrare grafică a contraexemplului lui Runge

## Exemplul 2 (Contraexemplul lui Bernstein)

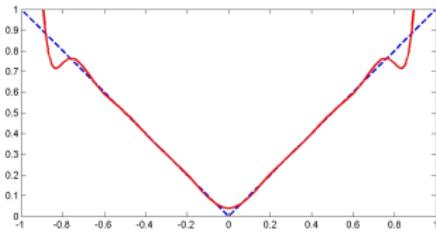
$$f(x) = |x|, \quad x \in [-1, 1]$$

$$x_k^{(m)} = -1 + \frac{2k}{m}, \quad k = 0, 1, 2, \dots, m \quad (6)$$

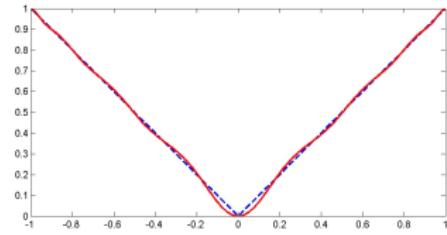
Problema analiticității nu se pune, deoarece  $f$  nu este derivabilă în  $x = 0$ . Se obține că

$$\lim_{m \rightarrow \infty} |f(x) - L_m(f; x)| = \infty \quad \forall x \in [-1, 1]$$

exceptând punctele  $x = -1$ ,  $x = 0$  și  $x = 1$ . Vezi figura 7, pentru  $m = 20$ . Convergența în  $x = \pm 1$  este trivială deoarece acestea sunt noduri de interpolare și deci eroarea în aceste puncte este 0. Același lucru este adevărat pentru  $x = 0$ , când  $n$  este impar, dar nu și când  $n$  este par. Eșecul convergenței pentru aceste noduri se explică doar parțial prin insuficiența regularității a lui  $f$ . Un alt motiv este distribuția uniformă a nodurilor. Există exemple mai bune de distribuții ale nodurilor (noduri Cebîșev). În figura 7 se dă graficul pentru  $m = 17$ .



Noduri echidistante



Noduri Cebîşev

# Introducere

Fie  $\Delta$  o diviziune a lui  $[a, b]$

$$\Delta : a = x_1 < x_2 < \cdots < x_{n-1} < x_n = b \quad (7)$$

Vom utiliza un polinom de grad mic pe subintervalul  $[x_i, x_{i+1}]$ ,  $i = \overline{1, n-1}$ . Motivul este acela că pe intervale suficient de mici funcțiile pot fi aproximate arbitrar de bine prin polinoame de grad mic, chiar 0 sau 1. Am introdus deja spațiul

$$S_m^k(\Delta) = \{s : s \in C^k[a, b], s|_{[x_i, x_{i+1}]} \in \mathbb{P}_m, i = 1, 2, \dots, n-1\} \quad (8)$$

$m \geq 0$ ,  $k \in \mathbb{N} \cup \{-1\}$ , numit **spațiul funcțiilor spline polinomiale de grad  $m$  și clasă de netezime  $k$** . Dacă  $k = m$ , atunci funcțiile  $s \in S_m^m(\Delta)$  sunt polinoame de grad  $m$ .

# Spline liniare I

Pentru  $m = 1$  și  $k = 0$  se obțin **spline liniare**.

Dorim să găsim  $s \in S_1^0(\Delta)$  astfel încât

$$s(x_i) = f_i, \text{ unde } f_i = f(x_i), \quad i = 1, 2, \dots, n.$$

Soluția este trivială, vezi figura 2. Pe intervalul  $[x_i, x_{i+1}]$

$$s(f; x) = f_i + (x - x_i)f[x_i, x_{i+1}], \quad (9)$$

iar

$$|f(x) - s(f(x))| \leq \frac{(\Delta x_i)^2}{8} \max_{x \in [x_i, x_{i+1}]} |f''(x)|. \quad (10)$$

$$\Delta x_i = x_{i+1} - x_i$$

Rezultă că

$$\|f(\cdot) - s(f, \cdot)\|_\infty \leq \frac{1}{8} |\Delta|^2 \|f''\|_\infty. \quad (11)$$

# Spline liniare II

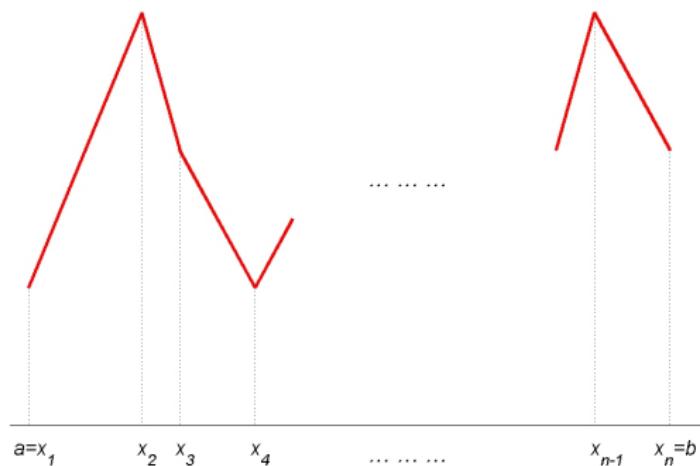


Figura: Spline liniare

## Spline liniare III

Dimensiunea lui  $S_1^0(\Delta)$  se calculează astfel: deoarece avem  $n - 1$  porțiuni și pe fiecare 2 coeficienți (2 grade de libertate) și fiecare condiție reduce numărul de grade de libertate cu 1, avem în final

$$\dim S_1^0(\Delta) = 2n - 2 - (n - 2) = n.$$

O bază a spațiului este dată de așa-numitele funcții *B-spline*:

Punem  $x_0 = x_1, x_{n+1} = x_n$ , pentru  $i = \overline{1, n}$

$$B_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & \text{pentru } x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & \text{pentru } x_i \leq x \leq x_{i+1} \\ 0, & \text{în rest} \end{cases} \quad (12)$$

Pentru  $i = 1$  prima și pentru  $i = n$  a doua ecuație se ignoră.

Funcția  $B_i$  se numește *pălărie chinezească*. Graficul funcțiilor  $B_i$  apare în figura 3.

# Spline liniare IV

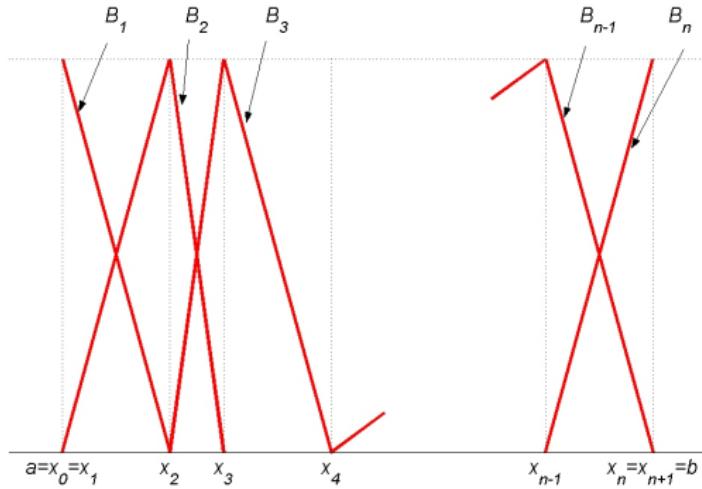


Figura: Funcții B-spline de grad 1

# Spline liniare V

Ele au proprietatea

$$B_i(x_j) = \delta_{ij},$$

sunt liniar independente, deoarece

$$s(x) = \sum_{i=1}^n c_i B_i(x) = 0 \wedge x \neq x_j \Rightarrow c_j = 0.$$

și

$$\langle B_i \rangle_{i=\overline{1,n}} = S_1^0(\Delta),$$

$B_i$  joacă același rol ca polinoamele fundamentale Lagrange  $\ell_i$ .

# Interpolare cu spline cubice I

Funcțiile spline cubice sunt cele mai utilizate.

Vom discuta întâi problema interpolării pentru  $s \in S_3^1(\Delta)$ . Continuitatea derivatei de ordinul I pentru  $s_3(f; \cdot)$  se poate realiza impunând valorile primei deriveate în fiecare punct  $x_i$ ,  $i = 1, 2, \dots, n$ . Astfel fie  $m_1, m_2, \dots, m_n$  numere arbitrarе date și notăm

$$s_3(f; \cdot)|_{[x_i, x_{i+1}]} = p_i(x), \quad i = 1, 2, \dots, n-1 \quad (13)$$

Realizăm  $s'_3(f; x_i) = m_i$ ,  $i = \overline{1, n}$ , luând fiecare bucată ca soluție unică a problemei de interpolare Hermite, și anume

$$\begin{aligned} p_i(x_i) &= f_i, & p_i(x_{i+1}) &= f_{i+1}, & i &= \overline{1, n-1}, \\ p'_i(x_i) &= m_i, & p'_i(x_{i+1}) &= m_{i+1} \end{aligned} \quad (14)$$

## Interpolare cu spline cubice II

Vom rezolva problema folosind interpolarea Newton. Diferențele divizate sunt

$$\begin{array}{lll} x_i & f_i & m_i \\ & & \frac{f[x_i, x_{i+1}] - m_i}{\Delta x_i} & \frac{m_{i+1} + m_i - 2f[x_i, x_{i+1}]}{(\Delta x_i)^2} \\ x_i & f_i & f[x_i, x_{i+1}] & \frac{m_{i+1} - f[x_i, x_{i+1}]}{\Delta x_i} \\ x_{i+1} & f_{i+1} & m_{i+1} \\ x_{i+1} & f_{i+1} \end{array}$$

și deci forma Newton a polinomului de interpolare Hermite este

$$p_i(x) = f_i + (x - x_i)m_i + (x - x_i)^2 \frac{f[x_i, x_{i+1}] - m_i}{\Delta x_i} + (x - x_i)^2(x - x_{i+1}) \frac{m_{i+1} + m_i - 2f[x_i, x_{i+1}]}{(\Delta x_i)^2}.$$

## Interpolare cu spline cubice III

Forma Taylor a lui  $p_i$  pentru  $x_i \leq x \leq x_{i+1}$  este

$$p_i(x) = c_{i,0} + c_{i,1}(x - x_i) + c_{i,2}(x - x_i)^2 + c_{i,3}(x - x_i)^3 \quad (15)$$

și deoarece  $x - x_{i+1} = x - x_i - \Delta x_i$ , prin identificare avem

$$c_{i,0} = f_i$$

$$c_{i,1} = m_i$$

$$c_{i,2} = \frac{f[x_i, x_{i+1}] - m_i}{\Delta x_i} - c_{i,3}\Delta x_i = \frac{3f[x_i, x_{i+1}] - 2m_i - m_{i+1}}{\Delta x_i} \quad (16)$$

$$c_{i,3} = \frac{m_{i+1} + m_i - 2f[x_i, x_{i+1}]}{(\Delta x_i)^2}$$

Deci, pentru a calcula  $s_3(f; x)$  într-un punct care nu este nod, trebuie în prealabil să localizăm intervalul  $[x_i, x_{i+1}] \ni x$ , să calculăm coeficienții cu (16) și să evaluăm spline-ul cu (15).

Vom discuta câteva alegeri posibile pentru  $m_1, m_2, \dots, m_n$ .

## Interpolare Hermite cubică pe porțiuni

Se alege  $m_i = f'(x_i)$  (presupunând că aceste derivate sunt cunoscute). Se ajunge la o schemă strict locală, în care fiecare bucată poate fi determinată independent de celalaltă. Mai mult, eroarea este

$$|f(x) - p_i(x)| \leq \left(\frac{1}{2}\Delta x_i\right)^4 \max_{x \in [x_i, x_{i+1}]} \frac{|f^{(4)}(x)|}{4!}, \quad x_i \leq x \leq x_{i+1}. \quad (17)$$

Deci

$$\|f(\cdot) - s_3(f; \cdot)\|_\infty \leq \frac{1}{384} |\Delta|^4 \|f^{(4)}\|_\infty. \quad (18)$$

Pentru puncte echidistante

$$|\Delta| = (b - a)/(n - 1)$$

și deci

$$\|f(\cdot) - s_3(f; \cdot)\|_\infty = O(n^{-4}), \quad n \rightarrow \infty. \quad (19)$$

## Spline cubice de clasă $C^2$ I

Cerem ca  $s_3(f; \cdot) \in S_3^2(\Delta)$ , adică continuitatea derivatelor de ordinul al II-lea. Aceasta înseamnă, cu notația (13)

$$p''_{i-1}(x_i) = p''_i(x_i), \quad i = \overline{2, n-1}, \quad (20)$$

care convertită în coeficienți Taylor (15) dă

$$2c_{i-1,2} + 6c_{i-1,3}\Delta x_{i-1} = 2c_{i,2}, \quad i = \overline{2, n-1}.$$

Înlocuind cu valorile explicite (16) pentru coeficienți se ajunge la sistemul liniar

$$\Delta x_i m_{i-1} + 2(\Delta x_{i-1} + \Delta x_i)m_i + (\Delta x_{i-1})m_{i+1} = b_i, \quad i = \overline{2, n-1} \quad (21)$$

unde

$$b_i = 3\{\Delta x_i f[x_{i-1}, x_i] + \Delta x_{i-1} f[x_i, x_{i+1}]\} \quad (22)$$

## Spline cubice de clasă $C^2$ II

Avem un sistem de  $n - 2$  ecuații liniare cu  $n$  necunoscute  $m_1, m_2, \dots, m_n$ . Odată alese  $m_1$  și  $m_n$ , sistemul devine tridiagonal și se poate rezolva eficient prin eliminare gaussiană, prin factorizare sau cu o metodă iterativă. Se dau în continuare câteva alegeri posibile pentru  $m_1$  și  $m_n$ .

## Spline complete(racordate, limitate)

Luăm  $m_1 = f'(a)$ ,  $m_n = f'(b)$ . Se știe că pentru acest tip de spline, dacă  $f \in C^4[a, b]$

$$\|f^{(r)}(\cdot) - s^{(r)}(f; \cdot)\|_{\infty} \leq c_r |\Delta|^{4-r} \|f^{(n)}\|_{\infty}, \quad r = 0, 1, 2, 3 \quad (23)$$

unde  $c_0 = \frac{5}{384}$ ,  $c_1 = \frac{1}{24}$ ,  $c_2 = \frac{3}{8}$ , iar  $c_3$  depinde de raportul  $\frac{|\Delta|}{\min_i \Delta x_i}$ .

## Spline care utilizează derivatele secunde

Impunem condițiile  $s_3''(f; a) = f''(a)$ ;  $s_3''(f; b) = f''(b)$ . Aceste condiții conduc la două ecuații suplimentare

$$\begin{aligned} 2m_1 + m_2 &= 3f[x_1, x_2] - \frac{1}{2}f''(a)\Delta x_1 \\ m_{n-1} + 2m_n &= 3f[x_{n-1}, x_n] + \frac{1}{2}f''(b)\Delta x_{n-1} \end{aligned} \tag{24}$$

Prima ecuație se pune la începutul sistemului (21), iar a doua la sfârșitul lui, păstrându-se astfel structura tridiagonală a sistemului.

# Spline cubice naturale

Impunând  $s''(f; a) = s''(f; b) = 0$ , se obțin două ecuații noi din (24) luând  $f''(a) = f''(b) = 0$ .

**Avantajul** – este nevoie numai de valori ale lui  $f$ , nu și ale derivatelor, dar prețul plătit este degradarea preciziei la  $O(|\Delta|^2)$  în vecinătatea capetelor (în afară de cazul când  $f''(a) = f''(b) = 0$ ).

## "Not-a-knot spline" (C. deBoor) I

Cerem ca  $p_1(x) \equiv p_2(x)$  și  $p_{n-2}(x) \equiv p_{n-1}(x)$ ; adică primele două părți și respectiv ultimele două trebuie să coincidă. Într-adevăr, asta înseamnă că primul punct interior  $x_2$  și ultimul  $x_{n-1}$  sunt ambele inactive. Se obțin încă două ecuații suplimentare exprimând continuitatea lui  $s_3'''(f; x)$  în  $x = x_2$  și  $x = x_{n-1}$ . Condiția de continuitate a lui  $s_3(f, .)$  în  $x_2$  și  $x_{n-1}$  revine la egalitatea coeficienților dominanți  $c_{1,3} = c_{2,3}$  și  $c_{n-2,3} = c_{n-1,3}$ . De aici se obțin ecuațiile

$$\begin{aligned} (\Delta x_2)^2 m_1 + [(\Delta x_2)^2 - (\Delta x_1)^2] m_2 - (\Delta x_1)^2 m_3 &= \beta_1 \\ (\Delta x_2)^2 m_{n-2} + [(\Delta x_2)^2 - (\Delta x_1)^2] m_{n-1} - (\Delta x_1)^2 m_n &= \beta_2, \end{aligned}$$

unde

$$\begin{aligned} \beta_1 &= 2\{(\Delta x_2)^2 f[x_1, x_2] - (\Delta x_1)^2 f[x_2, x_3]\} \\ \beta_2 &= 2\{(\Delta x_{n-1})^2 f[x_{n-2}, x_{n-1}] - (\Delta x_{n-2})^2 f[x_{n-1}, x_n]\}. \end{aligned}$$

## "Not-a-knot spline" (C. deBoor) II

Prima ecuație se adaugă pe prima poziție iar a doua pe ultima poziție a sistemului format din cele  $n - 2$  ecuații date de (21) și (22).

Sistemul obținut nu mai este tridiagonal, dar el se poate transforma în unul tridiagonal combinând ecuațiile 1 cu 2 și  $n - 1$  cu  $n$ . După aceste transformări prima și ultima ecuație devin

$$\Delta x_2 m_1 + (\Delta x_2 + \Delta x_1) m_2 = \gamma_1 \quad (25)$$

$$(\Delta x_{n-1} + \Delta x_{n-2}) m_{n-1} + \Delta x_{n-2} m_n = \gamma_2, \quad (26)$$

unde

$$\gamma_1 = \frac{1}{\Delta x_2 + \Delta x_1} \left\{ f[x_1, x_2] \Delta x_2 [\Delta x_1 + 2(\Delta x_1 + \Delta x_2)] + (\Delta x_1)^2 f[x_2, x_3] \right\}$$

$$\begin{aligned} \gamma_2 &= \frac{1}{\Delta x_{n-1} + \Delta x_{n-2}} \left\{ (\Delta x_{n-1})^2 f[x_{n-2}, x_{n-1}] + \right. \\ &\quad \left. [2(\Delta x_{n-1} + \Delta x_{n-2}) + \Delta x_{n-1}] \Delta x_{n-2} f[x_{n-1}, x_n] \right\}. \end{aligned}$$



Figura: Carl-Wilhelm Reinhold de Boor

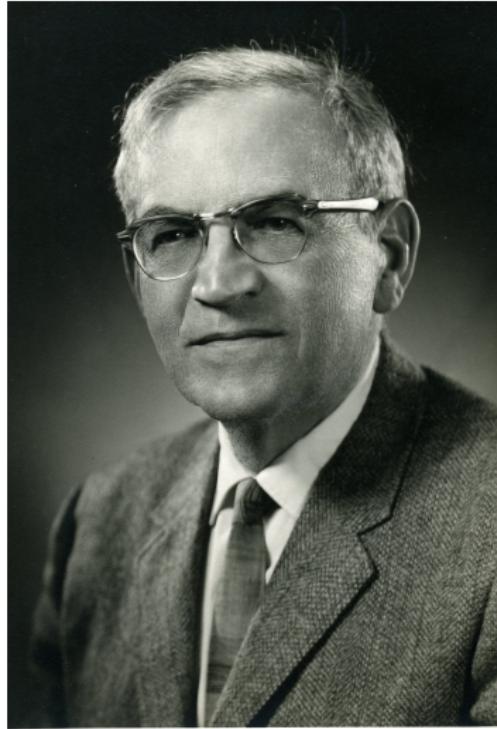


Figura: Isaac Schoenberg (1903-1990)

# Proprietatea de minimalitate a funcțiilor spline cubice I

Funcțiile spline cubice complete și naturale au proprietăți interesante de optimalitate. Pentru a le formula, este convenabil să considerăm nu numai subdiviziunea  $\Delta$  ci și

$$\Delta': a = x_0 = x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n = x_{n+1} = b, \quad (27)$$

în care capetele sunt noduri duble. Aceasta înseamnă că ori de câte ori interpolăm pe  $\Delta'$ , interpolăm valorile funcției pe punctele interioare, iar la capete valorile funcției și ale derivatei. Prima teoremă se referă la funcții spline cubice complete  $s_{compl}(f; \cdot)$ .

# Proprietatea de minimalitate a funcțiilor spline cubice II

## Teorema 3

Pentru orice funcție  $g \in C^2[a, b]$  care interpolează  $f$  pe  $\Delta'$ , are loc

$$\int_a^b [g''(x)]^2 dx \geq \int_a^b [s_{compl}''(f; x)]^2 dx, \quad (28)$$

cu egalitate dacă și numai dacă  $g(\cdot) = s_{compl}(f; \cdot)$ .

## Observația 4

$s_{compl}(f; \cdot)$  din teorema 3 interpolează  $f$  pe  $\Delta'$  și dintre toți interpolanții de acest tip, derivata sa de ordinul II are norma minimă.

**Demonstrație.** Folosim notația prescurtată  $s_{compl} = s$ . Teorema rezultă imediat, dacă arătăm că

$$\int_a^b [g''(x)]^2 dx = \int_a^b [g''(x) - s''(x)]^2 dx + \int_a^b [s''(x)]^2 dx. \quad (29)$$



## Proprietatea de minimalitate a funcțiilor spline cubice III

Aceasta implică imediat (28) și faptul că egalitatea în (28) are loc dacă și numai dacă  $g''(x) - s''(x) \equiv 0$ , din care integrând de două ori de la  $a$  la  $x$  și utilizând proprietățile de interpolare ale lui  $s$  și  $g$  în  $x = a$  se obține  $g(x) = s(x)$ . Relația (29) este echivalentă cu

$$\int_a^b s''(x)[g''(x) - s''(x)]dx = 0. \quad (30)$$

Integrând prin părți și ținând cont că  $s'(b) = g'(b) = f'(b)$  și  $s'(a) = g'(a) = f'(a)$  se obține

$$\begin{aligned} & \int_a^b s''(x)[g''(x) - s''(x)]dx = \\ &= s''(x)[g'(x) - s'(x)] \Big|_a^b - \int_a^b s'''(x)[g'(x) - s'(x)]dx = \\ &= - \int_a^b s'''(x)[g'(x) - s'(x)]dx. \end{aligned} \quad (31)$$

## Proprietatea de minimalitate a funcțiilor spline cubice IV

Deoarece  $s'''$  este constantă pe porțiuni

$$\begin{aligned} \int_a^b s'''(x)[g'(x) - s'(x)]dx &= \sum_{\nu=1}^{n-1} s'''(x_\nu + 0) \int_{x_\nu}^{x_{\nu+1}} [g'(x) - s'(x)]dx = \\ &= \sum_{\nu=1}^{n-1} s'''(x_\nu + 0) [g(x_{\nu+1}) - s(x_{\nu+1}) - (g(x_\nu) - s(x_\nu))] = 0 \end{aligned}$$

căci atât  $s$  cât și  $g$  interpolează  $f$  pe  $\Delta$ . Aceasta demonstrează (30) și deci și teorema. ■

Pentru interpolarea pe  $\Delta$ , calitatea de a fi optimal revine funcțiilor spline naturale de interpolare  $s_{nat}(f; \cdot)$ .

## Teorema 5

Pentru orice funcție  $g \in C^2[a, b]$  ce interpolează  $f$  pe  $\Delta$ , are loc

$$\int_a^b [g''(x)]^2 dx \geq \int_a^b [s''_{nat}(f; x)]^2 dx \quad (32)$$

cu egalitate dacă și numai dacă  $g(\cdot) = s_{nat}(f; \cdot)$ .

Demonstrația este analoagă cu a teoremei 3, deoarece (29) are loc din nou căci  $s''(b) = s''(a) = 0$ .

Punând  $g(\cdot) = s_{compl}(f; \cdot)$  în teorema 5 se obține

$$\int_a^b [s''_{compl}(f; x)]^2 dx \geq \int_a^b [s''_{nat}(f; x)]^2 dx. \quad (33)$$

Deci, într-un anumit sens, spline-ul cubic natural este cel mai neted interpolant.

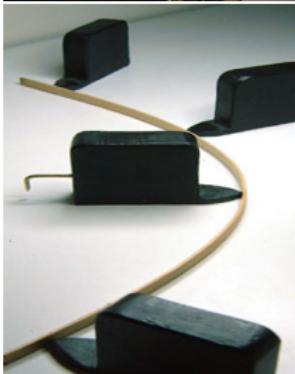
## Proprietatea de minimalitate a funcțiilor spline cubice VI

Proprietatea exprimată în teorema 5 stă la originea numelui de spline. Un spline este o vergea flexibilă folosită pentru a desena curbe. Dacă forma sa este dată de ecuația  $y = g(x)$ ,  $x \in [a, b]$  și dacă spline-ul trebuie să treacă prin punctele  $(x_i, g_i)$ , atunci se presupune că spline-ul are o formă ce minimizează energia potențială

$$\int_a^b \frac{[g''(x)]^2 dx}{(1 + [g'(x)]^2)^3},$$

pentru toate funcțiile  $g$  supuse acelorași restricții. Pentru variații lente ale lui  $g$  ( $\|g'\|_\infty \ll 1$ ) aceasta aproximează bine proprietatea de minim din teorema 5.

# Instrumentul spline



spline



spline



florar

# Metoda celor mai mici pătrate

## Aproximări în medie pătratică

Radu T. Trîmbițaș

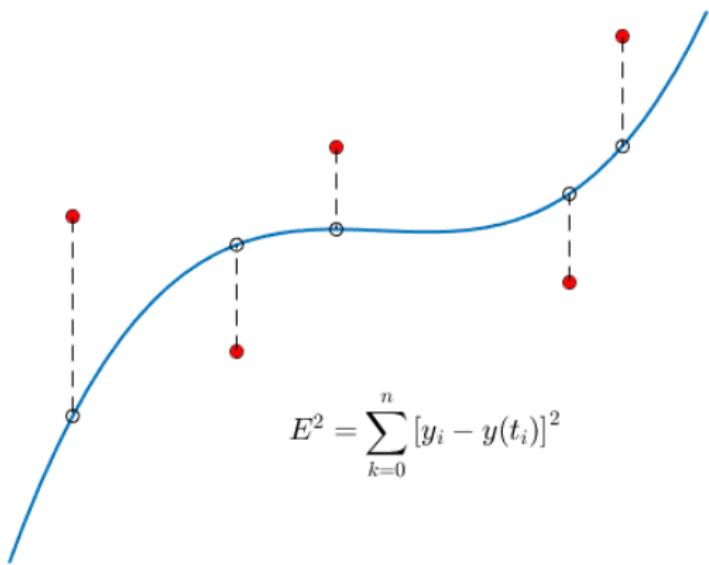
UBB

May 28, 2022

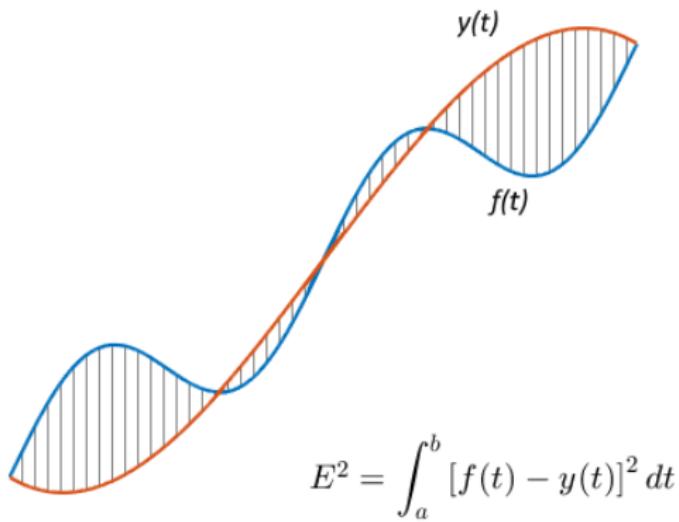
# Introducere

- Termenul *metoda celor mai mici pătrate* (MCMMMP) (least squares method) sau *aproximare în medie pătratică* descrie o abordare utilizată frecvent la rezolvarea unor sisteme supradeterminate sau specificate inexact (în sens aproximativ). În loc să rezolvăm sistemul exact, vom încerca să minimizăm suma pătratelor reziduurilor.
- Interpretare statistică: dacă se fac ipoteze adecvate probabilistice asupra distribuției erorilor, MCMMMP produce estimații de verosimilitate maximă ale parametrilor. Chiar dacă aceste ipoteze nu sunt satisfăcute, experiența a arătat că metoda produce rezultate utile.
- Algoritmii de rezolvare se bazează pe factorizări ortogonale ale matricelor.

# Aproximare MCMMP discretă



# Aproximare MCMMP continuă



# Modele și potrivirea curbelor I

- O sursă comună de probleme în sensul celor mai mici pătrate este *potrivirea curbelor (curve fitting)*. Fie  $t$  variabila independentă și  $y(t)$  o funcție necunoscută de  $t$  pe care dorim să o aproximăm.
- Să presupunem că avem  $m$  observații ( $y_i$ ) măsurate în valorile specificate ( $t_i$ ):

$$y_i = y(t_i), \quad i = \overline{1, m}.$$

- *Modelul* nostru este o combinație de  $n$  funcții de bază ( $\pi_i$ ),  $m \gg n$

$$y(t) \approx c_1\pi_1(t, \alpha) + \cdots + c_n\pi_n(t, \alpha).$$

- *Matricea de proiecțare (design matrix)*  $X(\alpha)$  va fi matricea cu elementele

$$x_{i,j} = \pi_j(t_i, \alpha),$$

ale cărei elemente pot depinde de  $\alpha$ .

## Modele și potrivirea curbelor II

- În notație matricială, modelul se poate exprima ca:

$$y \approx X(\alpha)c.$$

- Reziduurile sunt diferențele dintre valorile observate și cele date de model

$$r_i = y_i - \sum_{j=1}^n c_j \pi_j(t_i, \alpha) \quad (1)$$

sau în notație matricială

$$r = y - X(\alpha)c. \quad (2)$$

## Modele și potrivirea curbelor III

- Ne propunem să minimizăm o anumită normă a reziduurilor. Cele mai frecvente alegeri sunt

$$\|r\|_2^2 = \sum_{i=1}^m r_i^2$$

sau

$$\|r\|_{2,w}^2 = \sum_{i=1}^m w_i r_i^2.$$

- O explicație intuitivă, fizică, a celei de-a doua alegeri ar fi aceea că anumite observații sunt mai importante decât altele și le vom asocia ponderi,  $w_i$ . De exemplu, dacă la observația  $i$  eroarea este aproximativ  $e_i$ , atunci putem alege  $w_i = 1/e_i$ . Deci, avem de a face cu o problemă discretă de aproximare în sensul celor mai mici pătrate. Problema este *liniară* dacă nu depinde de  $\alpha$  și *nelinieră* în caz contrar.

## Modele și potrivirea curbelor IV

- Orice algoritm de rezolvare a unei probleme de aproximare în sensul celor mai mici pătrate fără ponderi poate fi utilizat la rezolvarea unei probleme cu ponderi prin scalarea observațiilor și a matricei de proiecție. În MATLAB aceasta se poate realiza prin  
 $X = \text{diag}(w) * X;$   
 $y = \text{diag}(w) * y$
- Dacă problema este liniară și avem mai multe observații decât funcții de bază, suntem conduși la rezolvarea sistemului supradeterminat

$$Xc \approx y,$$

pe care îl vom rezolva în sensul celor mai mici pătrate

$$c = X \backslash y.$$

# Ecuațiile normale I

- Dorim să rezolvăm

$$Xc \approx y \quad (3)$$

- Sistemul este supradeterminat (și în general incompatibil) — nu ne putem aștepta să îl rezolvăm exact. Îl vom rezolva în sensul celor mai mici pătrate:

$$\min_c \|y - Xc\|.$$

- Abordare teoretică: înmulțim ambii membri cu  $X^T$ . Aceasta reduce sistemul (3) la un sistem  $n \times n$ , pătratic, cunoscut sub numele de (sistem de) *ecuații normale*:

$$X^T Xc = X^T y. \quad (4)$$

## Ecuațiile normale II

- Matricea  $B = X^T X$  are elementele de forma

$$b_{ij} = (\pi_i(t), \pi_j(t)), \quad (5)$$

unde  $(\cdot, \cdot)$  este produsul scalar din  $\mathbb{R}^m$ .

- Sistemul (4) se scrie

$$\sum_{j=1}^n (\pi_i, \pi_j) c_j = (\pi_i, y), \quad i = 1, 2, \dots, n. \quad (6)$$

- Dacă există mii de observații și numai câțiva parametri, matricea de proiecțare  $X$  este mare, dar  $X^T X$  este mică. **Am proiectat  $y$  pe subspațiul generat de coloanele lui  $X$ .** Dacă funcțiile de bază sunt linear independente, atunci  $X^T X$  este nesingulară și

$$c = (X^T X)^{-1} X^T y.$$

## Ecuațiile normale III

- Această formulă apare în majoritatea textelor de statistică și metode numerice.
- Caracteristici nedorite: ineficiența și mai ales proasta condiționare: ecuațiile normale sunt întotdeauna mai prost condiționate decât sistemul inițial

$$\text{cond}(X^T X) = \text{cond}(X)^2.$$

- În aritmetică cu precizie finită, ecuațiile normale pot deveni singulare și  $(X^T X)^{-1}$  ar putea să nu existe, chiar dacă coloanele lui  $X$  sunt liniar independente.
- Exemplu

$$X = \begin{bmatrix} 1 & 1 \\ \delta & 0 \\ 0 & \delta \end{bmatrix}$$

## Ecuațiile normale IV

- Dacă  $\delta$  este mic, dar nenul, coloanele lui  $X$  sunt aproape paralele, dar liniar independente. Ecuațiile normale fac situația și mai proastă:

$$X^T X = \begin{bmatrix} 1 & 1 \\ \delta & 0 \\ 0 & \delta \end{bmatrix}^T \begin{bmatrix} 1 & 1 \\ \delta & 0 \\ 0 & \delta \end{bmatrix} = \begin{bmatrix} \delta^2 + 1 & 1 \\ 1 & \delta^2 + 1 \end{bmatrix}$$

- Dacă  $|\delta| < 10^{-8}$ , matricea  $X^T X$  calculată în dublă precizie este chiar singulară și inversa nu există.

# Factorizare QR I

- O metodă de a evita proasta condiționare este factorizarea QR:  
 $X = QR$ , unde  $Q$  este ortogonală, iar  $R$  este triunghiulară superior.

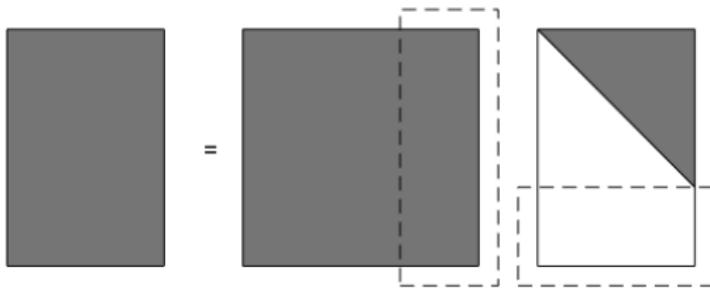
$$c = R^{-1} Q^T y.$$

- Abordarea numerică este să aplicăm ortogonalizarea atât matricei  $X$ , cât și membrului drept  $y$ . Vom obține astfel un sistem triunghiular superior, care se rezolvă prin substituție inversă.
- Operatorul \ din MATLAB știe să rezolve sisteme în sensul celor mai mici pătrate prin factorizare QR. Factorizarea QR se poate obține prin funcția MATLAB qr.
- Există două versiuni de factorizare QR.
  - În versiunea completă,  $R$  are aceeași dimensiune ca  $X$  iar  $Q$  este pătratică cu același număr de linii ca  $X$ .
  - În versiunea redusă,  $Q$  are aceeași mărime ca  $X$ , iar  $R$  este pătratică cu același număr de coloane ca  $X$ .

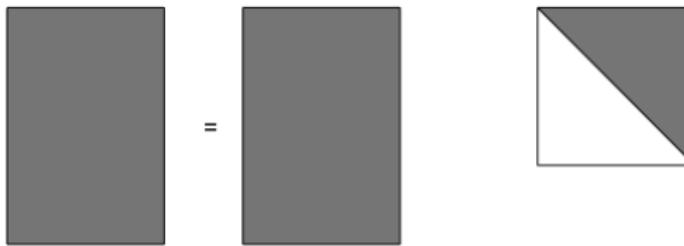
## Factorizare QR II

- Procesul Gram-Schmidt din algebra liniară generează aceeași factorizare, dar este mai puțin stabil numeric.

# Factorizare QR III



$X = Q R$



$X = \hat{Q} \hat{R}$

Figure: Factorizarea QR completă și redusă

# Derivate parțiale I

- Scriem aproximanta sub forma

$$\varphi = \sum_{j=1}^n c_j \pi_j(t).$$

- Eroarea (reziduul) va fi  $y - \varphi$ , iar

$$E(\varphi)^2 = \|r\|^2 = \sum_{k=1}^m \left( y_k - \sum_{j=1}^n c_j \pi_j(t_k) \right)^2$$

- Pătratul erorii este o funcție cuadratică de coeficienții  $c_1, \dots, c_n$  ai lui  $\varphi$ . Problema revine la a minimiza această funcție pătratică; ea se rezolvă anulând derivatele parțiale.

## Derivate parțiale II

- Se obține

$$\frac{\partial}{\partial c_i} E(\varphi)^2 = 2 \sum_{k=1}^m \left( y_k - \sum_{j=1}^n c_j \pi_j(t_k) \right) \pi_i(t_k) = 0,$$

adică

$$\sum_{j=1}^n c_j \left( \sum_{k=1}^m \pi_i(t_k) \pi_j(t_k) \right) = \sum_{k=1}^m \pi_i(t_k) y_k.$$

- Cu ajutorul produsului scalar

$$\sum_{j=1}^n (\pi_i, \pi_j) c_j = (\pi_i, y), \quad i = 1, 2, \dots, n, \tag{7}$$

adică chiar ecuațiile normale (6).

# Ortogonalitate I

- Fie  $\hat{\varphi}_n$  aproximanta cu coeficienții  $(\hat{c}_k)$  soluției ale ecuațiilor normale. Observăm întâi că eroarea  $y - \hat{\varphi}_n$  este ortogonală pe  $\Phi_n = \langle \pi_1, \dots, \pi_n \rangle$ , adică

$$(y - \hat{\varphi}_n, \varphi) = 0, \quad \forall \varphi \in \Phi_n \quad (8)$$

- Deoarece  $\varphi$  este o combinație liniară de  $\pi_i$ , este suficient ca (8) să aibă loc pentru fiecare  $\varphi = \pi_i, i = 1, 2, \dots, n$ .
- Se obține

$$(y - \hat{\varphi}_n, \pi_i) = \left( y - \sum_{j=1}^n \hat{c}_j \pi_j, \pi_i \right) = (y, \pi_i) - \sum_{j=1}^n \hat{c}_j (\pi_j, \pi_i) = 0,$$

sau

$$\sum_{j=1}^n (\pi_i, \pi_j) c_j = (\pi_i, y), \quad i = 1, 2, \dots, n, \quad (9)$$

adică chiar ecuațiile normale (6).

## Ortogonalitate II

- Rezultatul din (8) are o interpretare geometrică simplă. Aproximanta în sensul celor mai mici pătrate  $\hat{q}_n$  este proiecția ortogonală a lui  $y$  pe  $\Phi_n$ , vezi figura 2.

# Ortogonalitate III

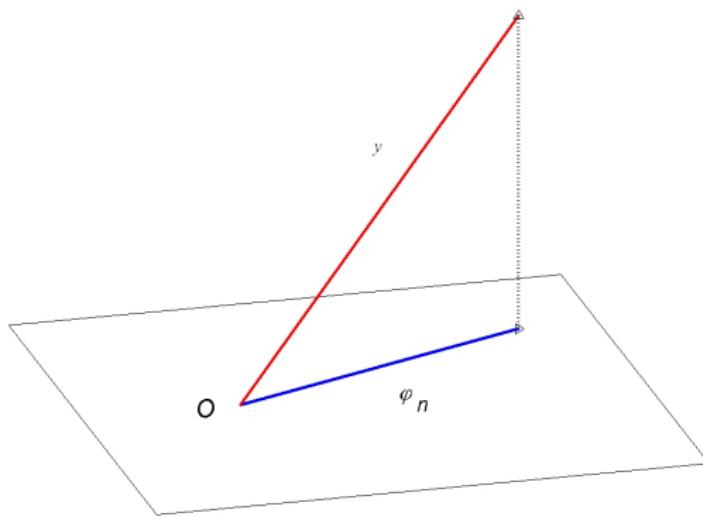


Figure: Interpretarea geometrică a aproximării prin MCMMP

# Exemplul 1 |

## Exemplu

Dându-se punctele

$$(0, -4), (1, 0), (2, 4), (3, -2),$$

determinați polinomul de gradul I corespunzător acestor date prin metoda celor mai mici pătrate.

**Soluție.** Aproximanta căutată are forma

$$\varphi(x) = c_0 + c_1 x.$$

Sistemul de ecuații normale se determină din condițiile  $f - \varphi \perp 1$  și  $f - \varphi \perp x$ . Se obține

$$\begin{cases} c_0(1, 1) + c_1(x, 1) = (f, 1) \\ c_0(1, x) + c_1(x, x) = (f, x) \end{cases}$$

## Exemplul 1 II

Dar,  $(1, 1) = \sum_{i=1}^4 1 \cdot 1 = 4$ ,

$(1, x) = (x, 1) = \sum_{i=1}^4 1 \cdot x_i = 1 \cdot 0 + 1 \cdot 1 + 1 \cdot 2 + 1 \cdot 3 = 6$ ,

$(x, x) = \sum_{i=1}^4 x_i^2 = 14$ . Pentru membrul drept avem  $(f, 1) = (y, 1) = -2$  și  $(f, x) = (y, x) = 2$ . Am obținut sistemul

$$\begin{bmatrix} 4 & 6 \\ 6 & 14 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

cu soluția  $c_0 = -2$ ,  $c_1 = 1$ . Deci  $\varphi(x) = x - 2$ . ■

## Exemplul 2 I

### Exemplu

Datele următoare dă populația SUA (în milioane) determinată la recensăminte de US Census, între anii 1900 și 2010. Dorim să modelăm populația și să o estimăm pentru anii 1975 și 2010.

An	Populația	An	Populația
1900	75.995	1960	179.320
1910	91.972	1970	203.210
1920	105.710	1980	226.510
1930	123.200	1990	249.630
1940	131.670	2000	281.420
1950	150.700	2010	308.790

**Soluție.** Vom modela populația printr-un model polinomial de gradul 3

$$y = c_0 + c_1 t + c_2 t^2 + c_3 t^3,$$

## Exemplul 2 II

și printr-un model exponențial

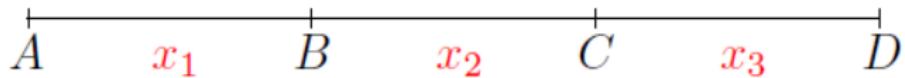
$$y = Ke^{\lambda t}.$$



## Exemplul 3 I

Exemplu (Măsurarea unui segment de drum - Stiefel)

*La măsurarea unui segment de drum, presupunem că am efectuat 5 măsurători*



$AD = 89m$ ,  $AC = 67m$ ,  $BD = 53m$ ,  $AB = 35m$  și  $CD = 20m$ , și că dorim să determinăm lungimea segmentelor  $x_1 = AB$ ,  $x_2 = BC$  și  $x_3 = CD$ .

## Exemplul 3 II

**Soluție.** Conform observațiilor obținem un sistem cu mai multe ecuații decât necunoscute (sistem supradeterminat):

$$\begin{array}{lcl} x_1 + x_2 + x_3 & = & 89 \\ x_1 + x_2 & = & 67 \\ x_2 + x_3 & = & 53 \\ x_1 & = & 35 \\ x_3 & = & 20 \end{array} \Leftrightarrow Ax = b, \quad A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 89 \\ 67 \\ 53 \\ 35 \\ 20 \end{bmatrix}.$$

Din ultimele trei ecuații se obține soluția  $x_1 = 35$ ,  $x_2 = 33$  și  $x_3 = 20$ . Totuși, dacă înlocuim în primele două ecuații se obține

$$\begin{aligned} x_1 + x_2 + x_3 - 89 &= -1, \\ x_1 + x_2 - 67 &= 1. \end{aligned}$$

### Exemplul 3 III

Ecuațiile sunt contradictorii și se ajunge la un sistem incompatibil. Un remediu este să găsim o soluție aproximativă care satisfacă sistemul cât mai bine posibil. Considerăm vectorul reziduu

$$r = b - Ax.$$

Căutăm un vector  $x$  care să minimizeze într-un anumit sens vectorul reziduu.

$$x = [ \begin{array}{ccc} 35.1250 & 32.5000 & 20.6250 \end{array} ]^T$$



# Aproximări continue I

- Dorim să approximăm o funcție pe un interval  $[a, b]$ , mărginit sau nemărginit

$$f \approx \varphi = c_1\pi_1 + \cdots + c_n\pi_n$$

- Produsul scalar va fi

$$(u, v) = \int_a^b w(t)u(t)v(t)dt,$$

iar norma

$$\|u\| = \sqrt{\int_a^b w(t)u^2(t)dt}.$$

- Funcția  $w$  este o funcție nenegativă pe  $[a, b]$ , adică  $w(t) \geq 0$ ,  $\forall t \in [a, b]$  și neidentic nulă pe orice subinterval al lui  $[a, b]$ .

## Aproximări continue II

- Reziduul  $r = f - \varphi$  va avea normă minimă

$$\|r\|^2 = \int_a^b w(t) [f(t) - \varphi(t)]^2 dt \quad (10)$$

$$= \int_a^b w(t) \left[ f(t) - \sum_{j=0}^n c_j \pi_j(t) \right]^2 dt =: E^2[\varphi] \quad (11)$$

- Ecuațiile normale se pot obține minimizând pătratul normei cu ajutorul derivatelor parțiale, sau din condiția de ortogonalitate

$$\sum_{j=1}^n (\pi_i, \pi_j) c_j = (\pi_i, f), \quad i = 1, 2, \dots, n. \quad (12)$$

## Aproximări continue III

- Ele formează un sistem de forma

$$Ac = b \quad (13)$$

unde matricea  $A$  și vectorul  $b$  au elementele

$$A = [a_{ij}], \quad a_{ij} = (\pi_i, \pi_j), \quad b = [b_i], \quad b_i = (\pi_i, f). \quad (14)$$

# Existența și unicitatea I

- Datorită simetriei produsului scalar,  $A$  este o matrice simetrică. Mai mult,  $A$  este pozitiv definită, adică

$$x^T A x = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j > 0, \text{ dacă } x \neq [0, 0, \dots, 0]^T. \quad (15)$$

- Funcția (15) se numește **formă pătratică** (deoarece este omogenă de grad 2). Pozitiv definitarea lui  $A$  ne spune că forma pătratică ai cărei coeficienți sunt elementele lui  $A$  este întotdeauna nenegativă și zero numai dacă variabilele  $x_i$  se anulează.
- Pentru a demonstra (15) să inserăm definiția lui  $a_{ij}$  și să utilizăm proprietățile produsului scalar

$$x^T A x = \sum_{i=1}^n \sum_{j=1}^n x_i x_j (\pi_i, \pi_j) = \sum_{i=1}^n \sum_{j=1}^n (x_i \pi_i, x_j \pi_j) = \left\| \sum_{i=1}^n x_i \pi_i \right\|^2.$$

## Existența și unicitatea II

- Aceasta este evident nenegativă. Ea este zero numai dacă  $\sum_{i=1}^n x_i \pi_i \equiv 0$  pe  $\text{supp } d\lambda$ , care pe baza liniar independenței lui  $\pi_i$  implică  $x_1 = x_2 = \dots = x_n = 0$ .
- Este un rezultat cunoscut din algebra liniară că o matrice  $A$  simetrică pozitiv definită este nesingulară. Într-adevăr, determinantul său, precum și minorii principali sunt strict pozitivi. Rezultă că sistemul de ecuații normale (7) are soluție unică.
- Corespunde această soluție minimului lui  $E^2[\varphi]$  în (10)? Matricea hessiană  $H = [\partial^2 E^2 / \partial c_i \partial c_j]$  trebuie să fie pozitiv definită. Dar  $H = 2A$ , deoarece  $E^2$  este o funcție cuadratică. De aceea,  $H$ , ca și  $A$ , este într-adevăr pozitiv definită și soluția ecuațiilor normale ne dă minimul dorit.

## Existența și unicitatea III

- Problema de aproximare în sensul celor mai mici pătrate are o soluție unică, dată de

$$\hat{\varphi}(t) = \sum_{j=1}^n \hat{c}_j \pi_j(t) \quad (16)$$

unde  $\hat{c} = [\hat{c}_1, \hat{c}_2, \dots, \hat{c}_n]^T$  este vectorul soluție al ecuațiilor normale (7).

# Neajunsurile MCMMP I

- Ecuațiile normale rezolvă problema de aproximare în sensul celor mai mici pătrate complet în teorie. Dar în practică?
- Referitor la o mulțime generală de funcții de bază liniar independente, pot apărea următoarele dificultăți:
  - ➊ Sistemul de ecuații normale (7) poate fi **prost condiționat**. Un exemplu simplu este următorul:  $\text{supp } d\lambda = [0, 1]$ ,  $d\lambda(t) = dt$  pe  $[0, 1]$  și  $\pi_j(t) = t^{j-1}$ ,  $j = 1, 2, \dots, n$ . Atunci

$$(\pi_i, \pi_j) = \int_0^1 t^{i+j-2} dt = \frac{1}{i+j-1}, \quad i, j = 1, 2, \dots, n,$$

adică matricea  $A$  este matricea Hilbert. Proasta condiționare a ecuațiilor normale se datorează alegerii neinspirate a funcțiilor de bază. Acestea devin aproape liniar dependente când exponentul crește. O altă sursă de degradare provine din elementele membrului drept

$b_j = \int_0^1 \pi_j(t) f(t) dt$ . Când  $j$  este mare  $\pi_j(t) = t^{j-1}$  se comportă pe

# Neajunsurile MCMMP II

$[0, 1]$  ca o funcție discontinuă. Un polinom care oscilează mai rapid pe  $[0, 1]$  ar fi de preferat, căci ar angaja mai viguros funcția  $f$ .

- ② Al doilea dezavantaj este faptul că toți coeficienții  $\hat{c}_j$  din (16) depind de  $n$ , adică  $\hat{c}_j = \hat{c}_j^{(n)}$ ,  $j = 1, 2, \dots, n$ . Mărirea lui  $n$  ne dă un nou sistem de ecuații mai mare și cu o soluție complet diferită. Acest fenomen se numește **nepermanența coeficienților**  $\hat{c}_j$ .
- Amândouă neajunsurile (1) și (2) pot fi eliminate (sau măcar atenuate) alegând ca funcții de bază un sistem ortogonal,

$$(\pi_i, \pi_j) = 0 \text{ dacă } i \neq j \quad (\pi_j, \pi_j) = \|\pi_j\|^2 > 0 \quad (17)$$

## Neajunsurile MCMMP III

- Atunci sistemul de ecuații normale devine diagonal și poate fi rezolvat imediat cu formula

$$\hat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}, \quad j = 1, 2, \dots, n. \quad (18)$$

Evident, acești coeficienți  $\hat{c}_j$  sunt independenți de  $n$  și odată calculați rămân la fel pentru orice  $n$  mai mare. Avem acum proprietatea de **permanență a coeficienților**. De asemenea nu trebuie să rezolvăm sistemul de ecuații normale, ci putem aplica direct (18).

# Neajunsurile MCMMP IV

- Orice sistem  $\{\hat{\pi}_j\}$  care este liniar independent pe  $\text{supp } d\lambda$  poate fi ortogonalizat (în raport cu măsura  $d\lambda$ ) prin **procedeul Gram-Schmidt**. Se ia

$$\pi_1 = \hat{\pi}_1$$

și apoi, pentru  $j = 2, 3, \dots$  se calculează recursiv

$$\pi_j = \hat{\pi}_j - \sum_{k=1}^{j-1} c_k \pi_k, \quad c_k = \frac{(\hat{\pi}_j, \pi_k)}{(\pi_k, \pi_k)}, \quad k = \overline{1, j-1}.$$

Atunci fiecare  $\pi_j$  astfel determinat este ortogonal pe toate funcțiile precedente.

# Exemple de sisteme ortogonale

- ① Sistemul trigonometric – cunoscut din analiza Fourier.
- ② Polinoame ortogonale

# Sistemul trigonometric I

- *Sistemul trigonometric* este format din funcțiile:

$$1, \cos t, \cos 2t, \cos 3t, \dots, \sin t, \sin 2t, \sin 3t, \dots$$

- El este ortogonal pe  $[0, 2\pi]$  în raport ponderea  $w(t) = 1$ .

$$\int_0^{2\pi} \sin kt \sin \ell t dt = \begin{cases} 0, & \text{pentru } k \neq \ell \\ \pi, & \text{pentru } k = \ell \end{cases} \quad k, \ell = 1, 2, 3, \dots$$

$$\int_0^{2\pi} \cos kt \cos \ell t dt = \begin{cases} 0, & k \neq \ell \\ 2\pi, & k = \ell = 0 \\ \pi, & k = \ell > 0 \end{cases} \quad k, \ell = 0, 1, 2$$

$$\int_0^{2\pi} \sin kt \cos \ell t dt = 0, \quad k = 1, 2, 3, \dots, \quad \ell = 0, 1, 2, \dots$$

## Sistemul trigonometric II

- Aproximarea are forma

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt). \quad (19)$$

- Utilizând (18) obținem

$$\begin{aligned} a_k &= \frac{1}{\pi} \int_0^{2\pi} f(t) \cos kt dt, \quad k = 1, 2, \dots \\ b_k &= \frac{1}{\pi} \int_0^{2\pi} f(t) \sin kt dt, \quad k = 1, 2, \dots \end{aligned} \quad (20)$$

numiți **coeficienți Fourier** ai lui  $f$ . Ei sunt coeficienții (18) pentru sistemul trigonometric.

- Prin extensie coeficienții (18) pentru orice sistem ortogonal  $(\pi_j)$  se vor numi **coeficienții Fourier** ai lui  $f$  relativ la acest sistem.

# Sistemul trigonometric III

- În particular, recunoaștem în seria Fourier trunchiată pentru  $k = n$  aproximarea lui  $f$  în clasa polinoamelor trigonometrice de grad  $\leq n$  relativ la norma

$$\|u\|_2 = \left( \int_0^{2\pi} |u(t)|^2 dt \right)^{1/2}$$

# Polinoame ortogonale I

- Dându-se o măsură  $d\lambda$ , stim că orice număr finit de puteri  $1, t, t^2, \dots$  sunt liniar independente pe  $[a, b]$ , dacă  $\text{supp } d\lambda = [a, b]$ , iar  $1, t, \dots, t^{N-1}$  liniar independente pe  $\text{supp } d\lambda = \{t_1, t_2, \dots, t_N\}$ .
- Deoarece o mulțime de vectori liniar independenți a unui spațiu liniar poate fi ortogonalizată prin procedeul Gram-Schmidt, orice măsură  $d\lambda$  de tipul considerat generează o mulțime unică de polinoame ortogonale monice  $\pi_j(t, d\lambda)$ ,  $j = 0, 1, 2, \dots$  ce satisfac

$$\text{grad } \pi_j = j, \quad j = 0, 1, 2, \dots$$

$$\int_{\mathbb{R}} \pi_k(t) \pi_\ell(t) d\lambda(t) = 0, \text{ dacă } k \neq \ell \quad (21)$$

- Aceste polinoame se numesc **polinoame ortogonale** relativ la măsura  $d\lambda$ .

## Polinoame ortogonale II

- Vom permite indicilor să meargă de la 0. Multimea  $\pi_j$  este infinită dacă  $\text{supp}d\lambda = [a, b]$  și constă din exact  $N$  polinoame  $\pi_0, \pi_1, \dots, \pi_{N-1}$  dacă  $\text{supp}d\lambda = \{t_1, \dots, t_N\}$ . În ultimul caz polinoamele se numesc **polinoame ortogonale discrete**.
- Între trei polinoame ortogonale monice (un polinom se numește **monic** dacă coeficientul său dominant este 1) consecutive există o relație liniară. Mai exact, există constantele reale  $\alpha_k = \alpha_k(d\lambda)$  și  $\beta_k = \beta_k(d\lambda) > 0$  (depinzând de măsura  $d\lambda$ ) astfel încât

$$\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0, 1, 2, \dots \quad (22)$$

$$\pi_{-1}(t) = 0, \quad \pi_0(t) = 1.$$

(Se subînțelege că (22) are loc pentru orice  $k \in \mathbb{N}$  dacă  $\text{supp}d\lambda = [a, b]$  și numai pentru  $k = \overline{0, N-2}$  dacă  $\text{supp}d\lambda = \{t_1, t_2, \dots, t_N\}$ ).

## Polinoame ortogonale III

- Pentru a demonstra (22) și a obține expresiile coeficienților să observăm că  $\pi_{k+1}(t) - t\pi_k(t)$  este un polinom de grad  $\leq k$ , și deci poate fi exprimat ca o combinație liniară a lui  $\pi_0, \pi_1, \dots, \pi_k$ . Scriem această combinație sub forma

$$\pi_{k+1} - t\pi_k(t) = -\alpha_k \pi_k(t) - \beta_k \pi_{k-1}(t) + \sum_{j=0}^{k-2} \gamma_{k,j} \pi_j(t) \quad (23)$$

(sumele vide se consideră nule).

- Înmulțim scalar ambii membri ai relației anterioare cu  $\pi_k$  și obținem

$$(-t\pi_k, \pi_k) = -\alpha_k (\pi_k, \pi_k)$$

adică

$$\alpha_k = \frac{(t\pi_k, \pi_k)}{(\pi_k, \pi_k)}, \quad k = 0, 1, 2, \dots \quad (24)$$

## Polinoame ortogonale IV

- La fel, înmulțind scalar cu  $\pi_{k-1}$  obținem

$$(-t\pi_k, \pi_{k-1}) = -\beta_k(\pi_{k-1}, \pi_{k-1}).$$

Deoarece  $(t\pi_k, \pi_{k-1}) = (\pi_k, t\pi_{k-1})$  și  $t\pi_{k-1}$  diferă de  $\pi_k$  printr-un polinom de grad  $< k$  se obține prin ortogonalitate

$(t\pi_k, \pi_{k-1}) = (\pi_k, \pi_k)$ , deci

$$\beta_k = \frac{(\pi_k, \pi_k)}{(\pi_{k-1}, \pi_{k-1})}, \quad k = 1, 2, \dots \quad (25)$$

- Înmulțind (23) cu  $\pi_\ell$ ,  $\ell < k - 1$ , se obține

$$\gamma_{k,\ell} = 0, \quad \ell = 0, 1, \dots, k - 1 \quad (26)$$

## Polinoame ortogonale V

- Formula de recurență (22) ne dă o modalitate practică de determinare a polinoamelor ortogonale. Deoarece  $\pi_0 = 1$ , putem calcula  $\alpha_0$  cu (24) pentru  $k = 0$  și apoi  $\pi_1$ , etc. Procedeul – numit **procedura lui Stieltjes** – este foarte potrivit pentru polinoame ortogonale discrete, căci în acest caz produsul scalar se exprimă prin sume finite.
- În cazul continuu, calculul produsului scalar necesită calcul de integrale, ceea ce complica lucrurile. Din fericire, pentru multe măsuri speciale importante, coeficienții se cunosc explicit.
- Cazul special când măsura este simetrică (adică  $d\lambda(t) = w(t)$  cu  $w(-t) = w(t)$  și  $\text{supp } d\lambda$  simetrică față de origine) merită o atenție specială, deoarece în acest caz  $\alpha_k = 0$ ,  $\forall k \in \mathbb{N}$ , conform lui (19) căci

$$(t\pi_k, \pi_k) = \int_{\mathbb{R}} w(t)t\pi_k^2(t)dt = \int_a^b w(t)t\pi_k^2(t)dt = 0,$$

## Polinoame ortogonale VI

deoarece avem o integrală dintr-o funcție impară pe un domeniu simetric.

- Rădăcinile polinoamelor ortogonale sunt reale, distincte și situate în interiorul intervalului  $[a, b]$ .

Deoarece  $\int_{\mathbb{R}} \pi_n(t) d\lambda(t) = 0$ , trebuie să existe cel puțin un punct interior lui  $[a, b]$  pe care  $\pi_n(t)$  își schimbă semnul. Fie  $t_1, t_2, \dots, t_k$ ,  $k \leq n$  toate aceste puncte. Dacă  $k < n$ , din ortogonalitate

$$\int_{\mathbb{R}} \pi_n(t) \prod_{j=1}^k (t - t_j) d\lambda(t) = 0,$$

imposibil, deoarece integrandul are semn constant. Deci,  $k = n$ .



Figure: Thomas Ioannes Stieltjes (1856-1894)

# Polinoamele lui Legendre I

- Se definesc prin aşa-numita formulă a lui Rodrigues

$$\pi_k(t) = \frac{k!}{(2k)!} \frac{d^k}{dt^k} (t^2 - 1)^k. \quad (27)$$

- Exemple:

$$\pi_0(t) = 1,$$

$$\pi_1(t) = t,$$

$$\pi_2(t) = t^2 - \frac{1}{3},$$

$$\pi_3(t) = t^3 - \frac{3}{5}t.$$

- Verificăm întâi ortogonalitatea pe  $[-1, 1]$  în raport cu ponderea  $w(t) = 1$ .

## Polinoamele lui Legendre II

- Pentru orice  $0 \leq \ell < k$ , prin integrare repetată prin părți se obține:

$$\begin{aligned} & \int_{-1}^1 \frac{d^k}{dt^k} (t^2 - 1)^k t^\ell dt \\ &= \sum_{m=0}^{\ell} \ell(\ell - 1) \dots (\ell - m + 1) t^{\ell-m} \frac{d^{k-m-1}}{dt^{k-m-1}} (t^2 - 1)^k \Big|_{-1}^1 = 0, \end{aligned}$$

ultima relație având loc deoarece  $0 \leq k - m - 1 < k$ .

- Deci,

$$(\pi_k, p) = 0, \quad \forall p \in \mathbb{P}_{k-1},$$

demonstrându-se astfel ortogonalitatea.

# Polinoamele lui Legendre III

- **Relația de recurență**
- Datorită simetriei, putem scrie

$$\pi_k(t) = t^k + \mu_k t^{k-2} + \dots, \quad k \geq 2$$

și observând (din nou datorită simetriei) că relația de recurență are forma

$$\pi_{k+1}(t) = t\pi_k(t) - \beta_k \pi_{k-1}(t),$$

obținem

$$\beta_k = \frac{t\pi_k(t) - \pi_{k+1}(t)}{\pi_{k-1}(t)},$$

care este valabilă pentru orice  $t$ .

## Polinoamele lui Legendre IV

- Făcând  $t \rightarrow \infty$ ,

$$\beta_k = \lim_{t \rightarrow \infty} \frac{t\pi_k(t) - \pi_{k+1}(t)}{\pi_{k-1}(t)} = \lim_{t \rightarrow \infty} \frac{(\mu_k - \mu_{k+1})t^{k-1} + \dots}{t^{k-1} + \dots} = \mu_k - \mu_{k+1}.$$

(Dacă  $k = 1$ , punem  $\mu_1 = 0$ .)

- Din formula lui Rodrigues rezultă

$$\begin{aligned}\pi_k(t) &= \frac{k!}{(2k)!} \frac{d^k}{dt^k} \left( t^{2k} - kt^{2k-2} + \dots \right) \\ &= \frac{k!}{(2k)!} (2k(2k-1)\dots(k+1)t^k - \\ &\quad k(2k-2)(2k-3)\dots(k-1)t^{k-1} + \dots) \\ &= t^k - \frac{k(k-1)}{2(2k-1)} t^{k-2} + \dots,\end{aligned}$$

# Polinoamele lui Legendre V

aşa că

$$\mu_k = \frac{k(k-1)}{2(2k-1)}, \quad k \geq 2.$$

Deci,

$$\beta_k = \mu_k - \mu_{k+1} = \frac{k^2}{(2k-1)(2k+1)}$$

şi deoarece  $\mu_1 = 0$ ,

$$\beta_k = \frac{1}{4 - k^{-2}}, \quad k \geq 1. \tag{28}$$

## Polinoamele Cebîșev de speță I I

- Ele se pot defini prin relația

$$T_n(x) = \cos(n \arccos x), \quad n \in \mathbb{N}. \quad (29)$$

- Din identitatea trigonometrică

$$\cos((k+1)\theta) + \cos((k-1)\theta) = 2 \cos \theta \cos k\theta$$

și din (29), punând  $\theta = \arccos x$  se obține

$$\begin{aligned} T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x) \quad k = 1, 2, 3, \dots \\ T_0(x) &= 1, \quad T_1(x) = x. \end{aligned} \quad (30)$$

# Polinoamele Cebîșev de speță I II

- De exemplu,

$$T_2(x) = 2x^2 - 1,$$

$$T_3(x) = 4x^3 - 3x,$$

$$T_4(x) = 8x^4 - 8x^2 + 1$$

s.a.m.d.

- Din relația (30) se obține pentru coeficientul dominant al lui  $T_n$  valoarea  $2^{n-1}$  (dacă  $n \geq 1$ ), deci polinomul Cebîșev de speță I monic este

$$\overset{\circ}{T}_n(x) = \frac{1}{2^{n-1}} T_n(x), \quad n \geq 0, \quad \overset{\circ}{T}_0 = T_0. \quad (31)$$

- Din (29) se pot obține rădăcinile lui  $T_n$

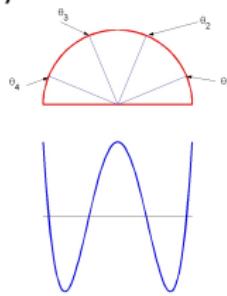
$$x_k^{(n)} = \cos \theta_k^{(n)}, \quad \theta_k^{(n)} = \frac{2k-1}{2n}\pi, \quad k = \overline{1, n}. \quad (32)$$

# Polinoamele Cebîșev de speță I III

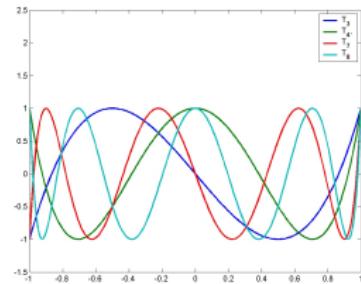
- Ele sunt proiecțiile pe axa reală ale punctelor de pe cercul unitate de argument  $\theta_k^{(n)}$ .
- Pe intervalul  $[-1, 1]$   $T_n$  oscilează de la +1 la -1, atingând aceste valori extreme în punctele

$$y_k^{(n)} = \cos \eta_k^{(n)}, \quad \eta_k^{(n)} = \frac{k\pi}{n}, \quad k = \overline{0, n}.$$

$T_4$  și rădăcinile sale



$T_3, T_4, T_7, T_8$  pe  $[-1, 1]$



# Polinoamele Cebîșev de speță I IV

- Polinoamele Cebîșev de speță I sunt ortogonale pe  $[-1, 1]$  în raport cu ponderea

$$w(x) = \frac{1}{\sqrt{1 - x^2}}.$$

- Se verifică ușor din (29) că

$$\begin{aligned} \int_{-1}^1 T_k(x) T_\ell(x) \frac{dx}{\sqrt{1 - x^2}} &= \int_0^\pi T_k(\cos \theta) T_\ell(\cos \theta) d\theta \\ &= \int_0^\pi \cos k\theta \cos \ell\theta d\theta = \begin{cases} 0 & \text{dacă } k \neq \ell \\ \pi & \text{dacă } k = \ell = 0 \\ \pi/2 & \text{dacă } k = \ell \neq 0 \end{cases} \quad (33) \end{aligned}$$

# Polinoamele Cebîșev de speță I V

- Dezvoltarea în serie Fourier de polinoame Cebîșev este dată de

$$f(x) = \sum_{j=0}^{\infty}' c_j T_j(x) = \frac{1}{2}c_0 + \sum_{j=1}^{\infty} c_j T_j(x), \quad (34)$$

unde

$$c_j = \frac{2}{\pi} \int_{-1}^1 f(x) T_j(x) \frac{dx}{\sqrt{1-x^2}}, \quad j \in \mathbb{N}.$$

- Păstrând în (34) numai termenii de grad cel mult  $n$  se obține o aproximare polinomială utilă de grad  $n$

$$\tau_n(x) = \sum_{j=0}^n' c_j T_j(x), \quad (35)$$

având eroarea

$$f(x) - \tau_n(x) = \sum_{j=n+1}^{\infty} c_j T_j(x) \approx c_{n+1} T_{n+1}(x). \quad (36)$$

# Polinoamele Cebîșev de speță I VI

- Aproximanta din (35) este cu atât mai bună cu cât coeficienții din extremitatea dreaptă tind mai repede către zero. Eroarea (36) oscilează în esență între  $+c_{n+1}$  și  $-c_{n+1}$  și este deci de mărime „uniformă”. Acest lucru contrastează puternic cu dezvoltarea Taylor în jurul lui  $x = 0$ , unde polinomul de grad  $n$  are eroarea proporțională cu  $x^{n+1}$  pe  $[-1, 1]$ .

# Minimalitatea normei I

## Teoremă

Pentru orice polinom monic  $\overset{\circ}{p}_n$  de grad  $n$  are loc

$$\max_{-1 \leq x \leq 1} \left| \overset{\circ}{p}_n(x) \right| \geq \max_{-1 \leq x \leq 1} \left| \overset{\circ}{T}_n(x) \right| = \frac{1}{2^{n-1}}, \quad n \geq 1, \quad (37)$$

unde  $\overset{\circ}{T}_n(x)$  este dat de (31).

**Demonstrație.** Se face prin reducere la absurd. Presupunem că

$$\max_{-1 \leq x \leq 1} \left| \overset{\circ}{p}_n(x) \right| < \frac{1}{2^{n-1}}. \quad (38)$$

## Minimalitatea normei II

Atunci polinomul  $d_n(x) = \overset{\circ}{T}_n(x) - \overset{\circ}{p}_n(x)$  (de grad  $\leq n - 1$ ) satisface

$$d_n(y_0^{(n)}) > 0, \quad d_n(y_1^{(n)}) < 0, \quad d_n(y_2^{(n)}) > 0, \dots, (-1)^n d_n(y_n^{(n)}) > 0. \quad (39)$$

Deoarece  $d_n$  are  $n$  schimbări de semn, el este identic nul; aceasta contrazice (39) și astfel (38) nu poate fi adevărată. ■

Rezultatul (37) se poate interpreta în modul următor: cea mai bună aproximare uniformă din  $\mathbb{P}_{n-1}$  pe  $[-1, 1]$  a lui  $f(x) = x^n$  este dată de  $x^n - \overset{\circ}{T}_n(x)$ , adică, de agregarea termenilor până la gradul  $n - 1$  din  $\overset{\circ}{T}_n$  luate cu semnul minus. Din teoria aproximățiilor uniforme se știe că cea mai bună aproximare polinomială uniformă este unică. Deci, egalitatea în (37) poate avea loc numai dacă  $\overset{\circ}{p}_n(x) = \overset{\circ}{T}_n(x)$ .

# Polinoamele Cebîșev de speță a II-a

- Se definesc prin

$$Q_n(t) = \frac{\sin[(n+1)\arccos t]}{\sqrt{1-t^2}}, \quad t \in [-1, 1]$$

- Ele sunt ortogonale pe  $[-1, 1]$  în raport cu măsura  $d\lambda(t) = w(t)dt$ ,  
 $w(t) = \sqrt{1-t^2}$ .
- Relația de recurență este

$$Q_{n+1}(t) = 2tQ_n(t) - Q_{n-1}(t), \quad Q_0(t) = 1, \quad Q_1(t) = 2t.$$



Figure: Pafnuti Lvovici Cebîşev (1821-1894)

# Polinoamele lui Laguerre I

- Sunt ortogonale pe  $[0, \infty)$  în raport cu ponderea  $w(t) = t^\alpha e^{-t}$ .
- Se definesc prin

$$\ell_n^\alpha(t) = \frac{e^t t^{-\alpha}}{n!} \frac{d^n}{dt^n}(t^{n+\alpha} e^{-t}) \text{ pentru } \alpha > 1$$

- Relația de recurență pentru polinoamele monice este

$$\ell_{k+1}^\alpha(t) = (t - 2k - \alpha - 1)\ell_k^\alpha(t) - \beta_k \ell_{k-1}^\alpha(t),$$

unde

$$\beta_k = \begin{cases} \Gamma(1 + \alpha), & \text{pentru } k = 0; \\ k(k + \alpha), & \text{pentru } k > 0. \end{cases}$$

# Polinoamele lui Laguerre II

- Exemple pentru  $\alpha = 0$ :

$$\ell_0^{(0)}(t) = 1,$$

$$\ell_1^{(0)}(t) = t - 1,$$

$$\ell_2^{(0)}(t) = t^2 - 4t + 2,$$

$$\ell_3^{(0)}(t) = t^3 - 9t^2 + 18t - 6$$



Figure: Edmond Laguerre (1834-1886)

# Polinoamele lui Hermite I

- Se definesc prin

$$H_n(t) = (-1)^n e^{t^2} \frac{d^n}{dt^n}(e^{-t^2}).$$

- Ele sunt ortogonale pe  $(-\infty, \infty)$  în raport cu ponderea  $w(t) = e^{-t^2}$  și verifică relația de recurență

$$H_{k+1}(t) = tH_k(t) - \beta_k H_{n-1}(t)$$

unde

$$\beta_k = \begin{cases} \sqrt{\pi}, & \text{pentru } k = 0; \\ \frac{k}{2}, & \text{pentru } k > 0. \end{cases}$$

# Polinoamele lui Hermite II

- Exemple:

$$H_0(t) = 1,$$

$$H_1(t) = t,$$

$$H_2(t) = t^2 - \frac{1}{2},$$

$$H_3(t) = t^3 - \frac{3}{2}t.$$



Figure: Charles Hermite (1822-1901)

# Polinoamele lui Jacobi I

- Sunt ortogonale pe  $[-1, 1]$  în raport cu ponderea

$$w(t) = (1-t)^\alpha(1+t)^\beta.$$

- Formula lui Rodrigues:

$$\begin{aligned} P_n^{(\alpha, \beta)}(t) &= \frac{(-1)^n}{2^n n!} (1-t)^{-\alpha} (1+t)^{-\beta} \cdot \\ &\quad \frac{d^n}{dz^n} \left\{ (1-t)^\alpha (1+t)^\beta (1-t^2)^n \right\}. \end{aligned}$$

## Polinoamele lui Jacobi II

- Coeficienții din relația de recurență sunt

$$\alpha_k = \frac{\beta^2 - \alpha^2}{(2k + \alpha + \beta)(2k + \alpha + \beta + 2)}$$

și

$$\beta_0 = 2^{\alpha+\beta+1} \frac{\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+1)},$$

$$\beta_1 = \frac{4(1+\alpha)(1+\beta)}{(2+\alpha+\beta)^2(3+\alpha+\beta)},$$

$$\beta_k = \frac{4k(k+\alpha)(k+\alpha+\beta)(k+\beta)}{(2k+\alpha+\beta-1)(2k+\alpha+\beta)^2(2k+\alpha+\beta+1)}, \quad k > 1.$$

# Polinoamele lui Jacobi III

- Exemple pentru  $\alpha = 1/2$  și  $\beta = -1/2$

$$\pi_0^{(\alpha, \beta)}(t) = 1,$$

$$\pi_1^{(\alpha, \beta)}(t) = t,$$

$$\pi_2^{(\alpha, \beta)}(t) = t^2 + \frac{1}{2}t - \frac{1}{4},$$

$$\pi_3^{(\alpha, \beta)}(t) = t^3 + \frac{1}{2}t^2 - \frac{1}{2}t - \frac{1}{8}.$$



Figure: Carl Gustav Jacob Jacobi (1804-1851)

## Exemplu

Pentru funcția  $f(t) = \arccos t$ ,  $t \in [-1, 1]$ , obțineți aproximanta în sensul celor mai mici pătrate,  $\hat{\varphi} \in P_n$  a lui  $f$  relativ la funcția pondere

$w(t) = (1 - t^2)^{-\frac{1}{2}} = \frac{1}{\sqrt{1-t^2}}$  adică, găsiți soluția  $\varphi = \hat{\varphi}$  a problemei

$$\min \left\{ \int_{-1}^1 [f(t) - \varphi(t)]^2 \frac{dt}{\sqrt{1-t^2}} : \varphi \in P_n \right\}.$$

Exprimați  $\varphi$  cu ajutorul polinoamelor Cebîșev  $\pi_j(t) = T_j(t)$ .

**Soluție.**  $\hat{\varphi}(t) = \frac{c_0}{2} + c_1 T_1(x) + \cdots + c_n T_n(x)$

$$c_k = \frac{(f, T_k)}{(T_k, T_k)} = \frac{2}{\pi} (f, T_k) = \frac{2}{\pi} \int_{-1}^1 \frac{\arccos t}{\sqrt{1-t^2}} \cos(k \arccos t) dt$$

$$= \frac{2}{\pi} \int_0^\pi u \cos k u du = \frac{2}{\pi} \left[ \frac{u \sin k u}{k} \Big|_0^\pi - \frac{1}{k} \int_0^\pi \sin k u du \right]$$

$$= \frac{2}{\pi} \left[ \frac{1}{k} \frac{\cos ku}{k} \Big|_0^\pi \right] = -\frac{2}{\pi k^2} [(-1)^k - 1]$$

$k$  par  $c_k = 0$

$k$  impar  $c_k = -\frac{2}{\pi k^2}(-2) = \frac{4}{\pi k^2}$  ■

# Bibliografie I

- Å. Björk, *Numerical Methods for Least Squares Problem*, SIAM, Philadelphia, 1996.
- E. Blum, *Numerical Computing: Theory and Practice*, Addison-Wesley, 1972.
- P. G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, Paris, Milan, Barcelone, Mexico, 1990.
- Gheorghe Coman, *Analiză numerică*, Editura Libris, Cluj-Napoca, 1995.
- W. Gander, M. Gander, F. Kwok, *Scientific Computing. An Introduction using Maple and MATLAB*, Springer, 2014
- W. Gautschi, *Numerical Analysis. An Introduction*, Birkhäuser, Basel, 1997.

## Bibliografie II

- W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, New York, Port Chester, Melbourne, Sidney, 1996, disponibila prin www, <http://www.nr.com/>.
- D. D. Stancu, *Analiză numerică – Curs și culegere de probleme*, Lito UBB, Cluj-Napoca, 1977.
- J. Stoer, R. Burlisch, *Introduction to Numerical Analysis*, 2nd ed., Springer Verlag, 1992.

# Aproximarea funcțiilor - metoda celor mai mici pătrate

## Aproximare în $L^2$

Radu Trîmbițaș

Universitatea “Babeș-Bolyai”

11 mai 2020

- Funcțiile de aproximat pot să fie definite pe:
  - un continuu (de regulă un interval) – funcții speciale pe care dorim să le evaluăm ca parte a unei subrute
  - pe o mulțime finită de puncte – situație întâlnită în științele fizice sau inginerie, când măsurările fizice se fac în funcție de alte cantități (cum ar fi timpul)
- Deoarece o astfel de evaluare trebuie să se reducă la un număr finit de operații aritmetice, trebuie în ultimă instanță să aproximăm funcțiile prin intermediul polinoamelor sau funcțiilor raționale.
- Dorim să aproximăm o funcție dată, cât mai bine posibil în termeni de funcții mai simple.

# Scheme de aproximare

- În general o **schemă de aproximare** poate fi descrisă prin:
  - ① o funcție  $f \in X$  ce urmează a fi aproximată;
  - ② o clasă  $\Phi$  de aproximante;
  - ③ o normă  $\|\cdot\|$  ce măsoară mărimea funcțiilor.

## Problema celei mai bune aproximări

- Căutăm o aproximare  $\hat{\varphi} \in \Phi$  a lui  $f$  astfel încât

$$\|f - \hat{\varphi}\| \leq \|f - \varphi\| \text{ pentru orice } \varphi \in \Phi. \quad (1)$$

- Această problemă se numește **problemă de cea mai bună aproximare** a lui  $f$  cu elemente din  $\Phi$ , iar funcția  $\hat{\varphi}$  se numește **cea mai bună aproximare** a lui  $f$  relativ la norma  $\|\cdot\|$ .
- $\Phi$  este un spațiu liniar finit dimensional sau o submulțime a acestuia.
- Cunoscându-se o bază  $\{\pi_j\}_{j=1}^n$  a lui  $\Phi$  putem scrie

$$\Phi = \Phi_n = \left\{ \varphi : \varphi(t) = \sum_{j=1}^n c_j \pi_j(t), c_j \in \mathbb{R} \right\}. \quad (2)$$

# Exemple de clase de aproximante I

## Exemplu

$\Phi = \mathbb{P}_m$  - mulțimea polinoamelor de grad cel mult  $m$ . O bază a sa este  $e_j(t) = t^j$ ,  $j = 0, 1, \dots, m$ . Deci  $\dim \mathbb{P}_m = m + 1$ . Polinoamele sunt cele mai utilizate aproximante pentru funcții pe domenii mărginite (intervale sau mulțimi finite de funcții). Motivul – teorema lui Weierstrass – orice funcție din  $C[a, b]$  poate fi aproximată oricât de bine printr-un polinom de grad suficient de mare.

## Exemple de clase de aproximante II

### Exemplu

$\Phi = \mathbb{S}_m^k(\Delta)$  spațiul funcțiilor spline polinomiale și cu clasa de netezime  $k$  pe subdiviziunea

$$\Delta : a = t_1 < t_2 < t_3 < \cdots < t_{N-1} < t_N = b$$

a intervalului  $[a, b]$ . Acestea sunt funcții polinomiale pe porțiuni de grad  $\leq m$ , legate în  $t_1, \dots, t_{N-1}$ , astfel încât toate derivatele până la ordinul  $k$  să fie continue. Presupunem  $0 \leq k < m$ . Pentru  $k = m$  se obține  $\mathbb{P}_m$ . Dacă  $k = -1$  permitem discontinuități în punctele de joncțiune.

# Exemple de clase de aproximante III

## Exemplu

$\Phi = \mathbb{T}_m[0, 2\pi]$  spațiul polinoamelor trigonometrice de grad cel mult  $m$  pe  $[0, 2\pi]$ . Acestea sunt combinații liniare ale funcțiilor

$$\begin{aligned}\pi_k(t) &= \cos((k-1)t), \quad k = \overline{1, m+1}, \\ \pi_{m+1-k}(t) &= \sin kt, \quad k = \overline{1, m}.\end{aligned}$$

Dimensiunea spațiului este  $n = 2m + 1$ . Astfel de aproximante sunt alegeri naturale dacă funcția de aproximat este periodică de perioadă  $2\pi$ . (Dacă  $f$  are perioada  $p$  se face schimbarea de variabilă  $t \rightarrow tp/2\pi$ .)

# Exemple de norme și tipuri de aproximare I

Câteva alegeri posibile ale normei, atât pentru aproximări continue, cât și pentru cele discrete apar în tabelul de mai jos

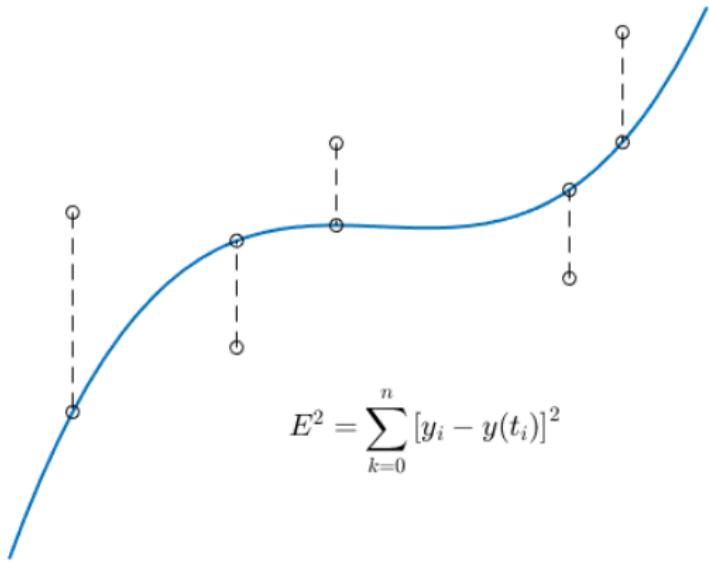
normă continuă	tip	normă discretă
$\ u\ _\infty = \max_{a \leq t \leq b}  u(t) $	$L_\infty$	$\ u\ _\infty = \max_{1 \leq i \leq N}  u(t_i) $
$\ u\ _{1,w} = \int_a^b  u(t) w(t)dt$	$L_w^1$	$\ u\ _{1,w} = \sum_{i=1}^n w_i  u(t_i) $
$\ u\ _{2,w} = \left( \int_a^b  u(t) ^2 w(t) dt \right)^{1/2}$	$L_w^2$	$\ u\ _{2,w} = \left( \sum_{i=1}^N w_i  u(t_i) ^2 \right)^{1/2}$

- Cazul continuu presupune un interval  $[a, b]$  și o **funcție pondere**  $w(t)$  (posibil și  $w(t) \equiv 1$ ) definită pe intervalul  $[a, b]$  și pozitivă, exceptând zerourile izolate. Intervalul  $[a, b]$  poate fi nemărginit, dacă funcția pondere  $w$  este astfel încât integrala pe  $[a, b]$  care definește norma să aibă sens. Funcția dată  $f$  și funcția  $\varphi$  din clasa  $\Phi$  trebuie definite pe  $[a, b]$  și norma  $\|f - \varphi\|$  să aibă sens.

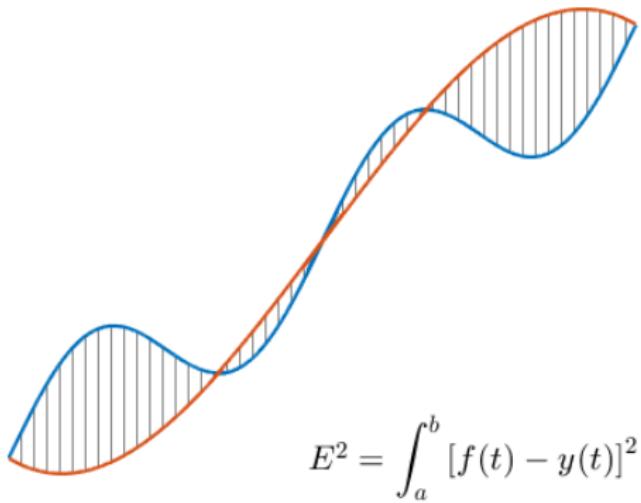
## Exemple de norme și tipuri de aproximare II

- Cazul discret presupune o mulțime de  $N$  puncte distințe  $t_1, t_2, \dots, t_N$  împreună cu ponderile  $w_1, w_2, \dots, w_N$  (posibil  $w_i = 1, i = \overline{1, N}$ ).  $f$  și  $\hat{\varphi}$  trebuie definite în punctele  $t_i$  în cazul discret.
- Deci combinând normele din tabelă cu spațiile liniare din exemple se obține o problemă de cea mai bună aproximare (1) cu sens.
- Dacă cea mai bună aproximantă  $\hat{\varphi}$  în cazul discret este astfel încât  $\|f - \hat{\varphi}\| = 0$ , atunci  $\hat{\varphi}(t_i) = f(t_i)$ , pentru  $i = 1, 2, \dots, N$ , spunem că  $\hat{\varphi}$  interpolează  $f$  în punctele  $t_i$  și numim această problemă de aproximare **problemă de interpolare**.
- Cele mai simple probleme de aproximare sunt problema celor mai mici pătrate și problema de interpolare.

# Aproximare MCMMMP discretă



# Aproximare MCMMP continuă



# Instrument notațional I

- Definim în cazul continuu

$$\lambda(t) = \begin{cases} 0, & \text{dacă } t < a \text{ (când } -\infty < a), \\ \int_a^t w(\tau) d\tau, & \text{dacă } a \leq t \leq b, \\ \int_a^b w(\tau) d\tau, & \text{dacă } t > b \text{ (când } b < \infty). \end{cases} \quad (3)$$

- putem scrie

$$\int_{\mathbb{R}} u(t) d\lambda(t) = \int_a^b u(t) w(t) dt, \quad \forall u \in C[a, b]$$

deoarece

$$d\lambda(t) = \begin{cases} w(t) dt, & t \in (a, b); \\ 0, & t \notin (a, b). \end{cases}$$

## Instrument notațional II

- $d\lambda$  se numește **măsură (pozitivă) continuă**
- **măsura discretă** (numită și „măsura Dirac“) asociată mulțimii de puncte  $\{t_1, t_2, \dots, t_N\}$  este o măsură  $d\lambda$  care este nenulă numai în punctele  $t_i$  și are aici valoarea  $w_i$ . Astfel în acest caz

$$\int_{\mathbb{R}} u(t) d\lambda(t) = \sum_{i=1}^N w_i u(t_i). \quad (4)$$

(definim  $\lambda(t)$  ca fiind o funcție în scară cu saltul în  $t_i$  egal cu  $w_i$ )

- unificare cu integrală Stieltjes: definim norma lui  $L_2$  prin

$$\|u\|_{2,d\lambda} = \left( \int_{\mathbb{R}} |u(t)|^2 d\lambda(t) \right)^{\frac{1}{2}} \quad (5)$$

și obținem norma continuă sau discretă după cum  $\lambda$  este ca în (3) sau o funcție în scară ca în (4).

## Instrument notațional III

- Vom numi **suportul** lui  $d\lambda$  – notat cu  $\text{supp}d\lambda$  – intervalul  $[a, b]$  în cazul continuu (presupunem că  $w$  este pozitivă pe  $[a, b]$  exceptând zerourile izolate) și mulțimea  $\{t_1, t_2, \dots, t_N\}$  în cazul discret.
- Spunem că mulțimea de funcții  $\pi_j$  din (2) **este liniar independentă** pe  $\text{supp}d\lambda$  dacă

$$\forall t \in \text{supp}d\lambda \quad \sum_{j=1}^n c_j \pi_j(t) \equiv 0 \Rightarrow c_1 = c_2 = \dots = c_k = 0 \quad (6)$$

## Aproximație prin metoda celor mai mici pătrate

Vom specializa problema (1) luând ca normă norma din  $L_2$

$$\|u\|_{2,d_\lambda} = \left( \int_{\mathbb{R}} |u(t)|^2 d\lambda(t) \right)^{\frac{1}{2}}, \quad (7)$$

unde  $d\lambda$  este fie o măsură continuă (conform (3)) sau discretă (conform (4)) și utilizând aproximanta  $\varphi$  dintr-un spațiu liniar  $n$ -dimensional

$$\Phi = \Phi_n = \left\{ \varphi : \varphi(t) = \sum_{j=1}^n c_j \pi_j(t), c_j \in \mathbb{R} \right\}. \quad (8)$$

$\pi_j$  liniar independente pe  $\text{supp } d\lambda$ ; integrala din (7) are sens pentru  $u = \pi_j$ ,  $j = 1, \dots, n$  și  $u = f$ . Problema astfel obținută se numește **problemă de aproximare în sensul celor mai mici pătrate** sau **problemă de aproximare în medie pătratică**.

# Produse scalare

- Definim produsul scalar prin

$$(u, v) = \int_{\mathbb{R}} u(t)v(t)d\lambda(t). \quad (9)$$

- Definiție corectă, conform inegalității Cauchy-Buniakovski-Schwarz

$$\|(u, v)\| \leq \|u\|_{2,d_\lambda} \|v\|_{2,d_\lambda}$$

- Proprietăți

- simetria  $(u, v) = (v, u)$ ;
- liniaritatea  $(\alpha_1 u_1 + \alpha_2 u_2, v) = \alpha_1(u_1, v) + \alpha_2(u_2, v)$ ;
- pozitiv definirea  $(u, u) \geq 0$  și  $(u, u) = 0 \Leftrightarrow u \equiv 0$  pe  $\text{supp } d\lambda$ .

- De asemenea

$$\|u\|_{2,d_\lambda}^2 = (u, u). \quad (10)$$

# Ortogonalitate

- Spunem că  $u$  și  $v$  sunt **ortogonale** dacă

$$(u, v) = 0. \quad (11)$$

- Mai general, putem considera sisteme ortogonale  $\{u_k\}_{k=1}^n$ :

$$(u_i, u_j) = 0 \text{ dacă } i \neq j, \quad u_k \neq 0 \text{ pe } \text{supp} d\lambda; \quad i, j = \overline{1, n}, \quad k = \overline{1, n}. \quad (12)$$

- Pentru un astfel de sistem are loc **teorema generalizată a lui Pitagora**

$$\left\| \sum_{k=1}^n \alpha_k u_k \right\|^2 = \sum_{k=1}^n |\alpha_k|^2 \|u_k\|^2. \quad (13)$$

- (13)  $\Rightarrow$  orice sistem ortogonal este liniar independent pe  $\text{supp} d\lambda$ .  
Într-adevăr, dacă membrul stâng al lui (13) se anulează, atunci și membrul drept se anulează și deoarece  $\|u_k\|^2 > 0$ , din ipoteză rezultă  $\alpha_1 = \alpha_2 = \dots = \alpha_n = 0$ .

# Ecuațiile normale I

- Putem scrie pătratul erorii din  $L_2$  sub forma

$$E^2[\varphi] := \|\varphi - f\|^2 = (\varphi - f, \varphi - f) = (\varphi, \varphi) - 2(\varphi, f) + (f, f).$$

- Înlocuind pe  $\varphi$  cu expresia sa se obține

$$\begin{aligned} E^2[\varphi] &= \int_{\mathbb{R}} \left( \sum_{j=1}^n c_j \pi_j(t) \right)^2 d\lambda(t) - 2 \int_{\mathbb{R}} \left( \sum_{j=1}^n c_j \pi_j(t) \right) f(t) d\lambda(t) + \\ &\quad + \int_{\mathbb{R}} f^2(t) d\lambda(t). \end{aligned} \tag{14}$$

- Pătratul erorii din  $L_2$  este o funcție quadratică de coeficienții  $c_1, \dots, c_n$  ai lui  $\varphi$ . Problema celei mai bune aproximări în  $L_2$  revine la a minimiza această funcție pătratică; ea se rezolvă anulând derivatele parțiale.

## Ecuațiile normale II

- Se obține

$$\frac{\partial}{\partial c_i} E^2[\varphi] = 2 \int_{\mathbb{R}} \left( \sum_{j=1}^n c_j \pi_j(t) \right) \pi_i(t) d\lambda(t) - 2 \int_{\mathbb{R}} \pi_i(t) f(t) d\lambda(t) = 0$$

adică

$$\sum_{j=1}^n (\pi_i, \pi_j) c_j = (\pi_i, f), \quad i = 1, 2, \dots, n. \quad (15)$$

- Aceste ecuații se numesc **ecuații normale** pentru problema celor mai mici pătrate.
- Ele formează un sistem de forma

$$Ac = b \quad (16)$$

unde matricea  $A$  și vectorul  $b$  au elementele

$$A = [a_{ij}], \quad a_{ij} = (\pi_i, \pi_j), \quad b = [b_i], \quad b_i = (\pi_i, f). \quad (17)$$

## Existența și unicitatea soluției I

- Datorită simetriei produsului scalar,  $A$  este o matrice simetrică. Mai mult,  $A$  este pozitiv definită, adică

$$x^T A x = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j > 0 \text{ dacă } x \neq [0, 0, \dots, 0]^T. \quad (18)$$

- Funcția (18) se numește **formă pătratică** (deoarece este omogenă de grad 2). Pozitiv definitarea lui  $A$  ne spune că forma pătratică ai cărei coeficienți sunt elementele lui  $A$  este întotdeauna nenegativă și zero numai dacă variabilele  $x_i$  se anulează.
- Pentru a demonstra (18) să inserăm definiția lui  $a_{ij}$  și să utilizăm proprietățile (i)-(iii) ale produsului scalar

$$x^T A x = \sum_{i=1}^n \sum_{j=1}^n x_i x_j (\pi_i, \pi_j) = \sum_{i=1}^n \sum_{j=1}^n (x_i \pi_i, x_j \pi_j) = \left\| \sum_{i=1}^n x_i \pi_i \right\|^2.$$

## Existența și unicitatea soluției II

- Aceasta este evident nenegativă. Ea este zero numai dacă  $\sum_{i=1}^n x_i \pi_i \equiv 0$  pe  $\text{supp } d\lambda$ , care pe baza liniar independenței lui  $\pi_i$  implică  $x_1 = x_2 = \dots = x_n = 0$ .
- Este un rezultat cunoscut din algebra liniară că o matrice  $A$  simetrică pozitiv definită este nesingulară. Într-adevăr, determinantul său, precum și minorii principali sunt strict pozitivi. Rezultă că sistemul de ecuații normale (15) are soluție unică.
- Corespunde această soluție minimului lui  $E[\varphi]$  în (14)? Matricea hessiană  $H = [\partial^2 E^2 / \partial c_i \partial c_j]$  trebuie să fie pozitiv definită. Dar  $H = 2A$ , deoarece  $E^2$  este o funcție cuadratică. De aceea,  $H$ , ca și  $A$ , este într-adevăr pozitiv definită și soluția ecuațiilor normale ne dă minimul dorit.

## Existența și unicitatea soluției III

- Problema de aproximare în sensul celor mai mici pătrate are o soluție unică, dată de

$$\hat{\varphi}(t) = \sum_{j=1}^n \hat{c}_j \pi_j(t) \quad (19)$$

unde  $\hat{c} = [\hat{c}_1, \hat{c}_2, \dots, \hat{c}_n]^T$  este vectorul soluție al ecuațiilor normale (15).

# Exemplu I

## Exemplu

Dându-se punctele

$$(0, -4), (1, 0), (2, 4), (3, -2),$$

determinați polinomul de gradul I corespunzător acestor date prin metoda celor mai mici pătrate.

**Soluție.** Aproximanta căutată are forma

$$\varphi(x) = c_0 + c_1 x.$$

Sistemul de ecuații normale se determină din condițiile  $f - \varphi \perp 1$  și  $f - \varphi \perp x$ . Se obține

$$\begin{cases} c_0(1, 1) + c_1(x, 1) = (f, 1) \\ c_0(1, x) + c_1(x, x) = (f, x) \end{cases}$$

## Exemplu II

Dar,  $(1, 1) = \sum_{i=1}^4 1 \cdot 1 = 3$ ,

$(1, x) = (x, 1) = \sum_{i=1}^4 1 \cdot x_i = 1 \cdot 0 + 1 \cdot 1 + 1 \cdot 2 + 1 \cdot 3 = 6$ ,

$(x, x) = \sum_{i=1}^4 x_i^2 = 14$ . Pentru membrul drept avem  $(f, 1) = (y, 1) = -2$  și  $(f, x) = (y, x) = 2$ . Am obținut sistemul

$$\begin{bmatrix} 4 & 6 \\ 6 & 14 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

cu soluția  $c_0 = -2$ ,  $c_1 = 1$ . Deci  $\varphi(x) = x - 2$ . ■

# Neajunsuri ale MCMMP I

- Ecuațiile normale rezolvă problema de aproximare în sensul celor mai mici pătrate complet în teorie. Dar în practică?
- Referitor la o mulțime generală de funcții de bază liniar independente, pot apărea următoarele dificultăți:
  - ➊ Sistemul de ecuații normale (15) poate fi **prost condiționat**. Un exemplu simplu este următorul:  $\text{supp}d\lambda = [0, 1]$ ,  $d\lambda(t) = dt$  pe  $[0, 1]$  și  $\pi_j(t) = t^{j-1}$ ,  $j = 1, 2, \dots, n$ . Atunci

$$(\pi_i, \pi_j) = \int_0^1 t^{i+j-2} dt = \frac{1}{i+j-1}, \quad i, j = 1, 2, \dots, n,$$

adică matricea  $A$  este matricea Hilbert. Proasta condiționare a ecuațiilor normale se datorează alegerii neinspirate a funcțiilor de bază. Acestea devin aproape liniar dependente când exponentul crește. O altă sursă de degradare provine din elementele membrului drept

$b_j = \int_0^1 \pi_j(t)f(t)dt$ . Când  $j$  este mare  $\pi_j(t) = t^{j-1}$  se comportă pe

$[0, 1]$  ca o funcție discontinuă. Un polinom care oscilează mai rapid pe  $[0, 1]$  ar fi de preferat, căci ar angaja mai viguros funcția  $f$ .

- ② Al doilea dezavantaj este faptul că toți coeficienții  $\hat{c}_j$  din (19) depind de  $n$ , adică  $\hat{c}_j = \hat{c}_j^{(n)}$ ,  $j = 1, 2, \dots, n$ . Mărirea lui  $n$  ne dă un nou sistem de ecuații mai mare și cu o soluție complet diferită. Acest fenomen se numește **nepermanența coeficienților**  $\hat{c}_j$ .
- Amândouă neajunsurile (1) și (2) pot fi eliminate (sau măcar atenuate) alegând ca funcții de bază un sistem ortogonal,

$$(\pi_i, \pi_j) = 0 \text{ dacă } i \neq j \quad (\pi_j, \pi_j) = \|\pi_j\|^2 > 0 \quad (20)$$

- Atunci sistemul de ecuații normale devine diagonal și poate fi rezolvat imediat cu formula

$$\hat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}, \quad j = 1, 2, \dots, n. \quad (21)$$

Evident, acești coeficienți  $\hat{c}_j$  sunt independenți de  $n$  și odată calculați rămân la fel pentru orice  $n$  mai mare. Avem acum proprietatea de **permanență a coeficienților**. De asemenea nu trebuie să rezolvăm sistemul de ecuații normale, ci putem aplica direct (21).

- Orice sistem  $\{\hat{\pi}_j\}$  care este liniar independent pe  $\text{supp } d\lambda$  poate fi ortogonalizat (în raport cu măsura  $d\lambda$ ) prin **procedeul Gram-Schmidt**. Se ia

$$\pi_1 = \hat{\pi}_1$$

și apoi, pentru  $j = 2, 3, \dots$  se calculează recursiv

$$\pi_j = \hat{\pi}_j - \sum_{k=1}^{j-1} c_k \pi_k, \quad c_k = \frac{(\hat{\pi}_j, \pi_k)}{(\pi_k, \pi_k)}, \quad k = \overline{1, j-1}.$$

Atunci fiecare  $\pi_j$  astfel determinat este ortogonal pe toate funcțiile precedente.

# Eroarea în MCMMP I

- Am văzut că dacă  $\Phi = \Phi_n$  constă din  $n$  funcții  $\pi_j$ ,  $j = 1, 2, \dots, n$  care sunt liniar independente pe  $\text{supp} d\lambda$ , atunci problema de aproximare în sensul celor mai mici pătrate pentru această măsură

$$\min_{\varphi \in \phi_n} \|f - \varphi\|_{2,d_\lambda} = \|f - \hat{\varphi}\|_{2,d_\lambda} \quad (22)$$

are o soluție unică  $\hat{\varphi} = \hat{\varphi}_n$ , dată de (19).

- Există multe moduri de a selecta baza  $\{\pi_j\}$  a lui  $\Phi_n$  și de aceea mai multe moduri de a reprezenta soluția, care conduc totuși la aceeași funcție. Eroarea în sensul celor mai mici pătrate – cantitatea din dreapta relației (22) – este independentă de alegerea funcțiilor de bază (deși calculul soluției, aşa cum s-a menționat anterior, nu este).
- În studiul acestor erori, putem presupune fără a restrânge generalitatea că baza  $\pi_j$  este un sistem ortogonal (fiecare sistem liniar independent poate fi ortogonalizat prin procedeul Gram-Schmidt).

- Avem conform (21)

$$\widehat{\varphi}_n(t) = \sum_{j=1}^n \widehat{c}_j \pi_j(t), \quad \widehat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}. \quad (23)$$

- Observăm întâi că eroarea  $f - \varphi_n$  este ortogonală pe  $\Phi_n$ , adică

$$(f - \widehat{\varphi}_n, \varphi) = 0, \quad \forall \varphi \in \Phi_n \quad (24)$$

unde produsul scalar este cel din (9).

- Deoarece  $\varphi$  este o combinație liniară de  $\pi_k$ , este suficient să arătăm (24) pentru fiecare  $\varphi = \pi_k$ ,  $k = 1, 2, \dots, n$ .

- Înlocuind  $\varphi_n$  cu expresia sa din (23) în (24), găsim

$$(f - \hat{\varphi}_n, \pi_k) = \left( f - \sum_{j=1}^n \hat{c}_j \pi_k, \pi_k \right) = (f, \pi_k) - \hat{c}_k (\pi_k, \pi_k) = 0,$$

ultima ecuație rezultând din formula pentru  $\hat{c}_k$  din (23).

- Rezultatul din (24) are o interpretare geometrică simplă. Dacă reprezentăm funcțiile ca vectori și spațiul  $\Phi_n$  ca un plan, atunci pentru orice funcție  $f$  care înceapă planul  $\Phi_n$ , aproximanta în sensul celor mai mici pătrate  $\hat{\varphi}_n$  este proiecția ortogonală a lui  $f$  pe  $\Phi_n$ , vezi figura 1.

## Eroarea în MCMMP IV

- În particular, alegând  $\varphi = \widehat{\varphi}_n$  în (24) obținem

$$(f - \widehat{\varphi}_n, \widehat{\varphi}_n) = 0$$

și de aceea, deoarece  $f = (f - \widehat{\varphi}) + \widehat{\varphi}$ , conform teoremei lui Pitagora și generalizării sale (13)

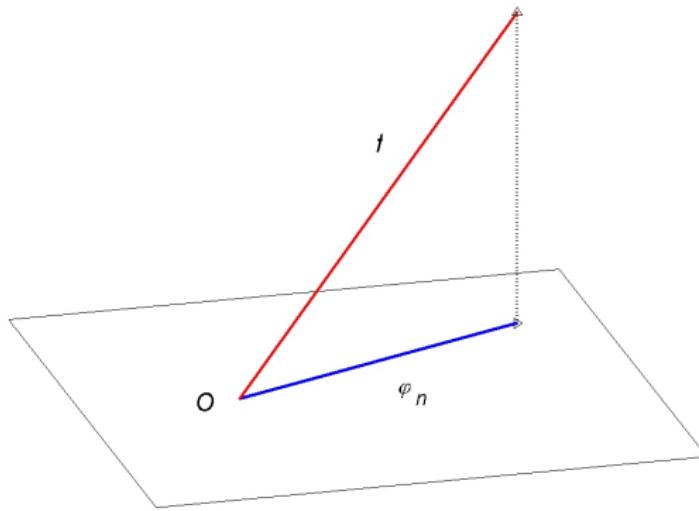
$$\begin{aligned}\|f\|^2 &= \|f - \widehat{\varphi}\|^2 + \|\widehat{\varphi}\|^2 \\ &= \|f - \widehat{\varphi}_n\|^2 + \left\| \sum_{j=1}^n \widehat{c}_j \pi_j \right\|^2 \\ &= \|f - \widehat{\varphi}_n\|^2 + \sum_{j=1}^n |\widehat{c}_j|^2 \|\pi_j\|^2.\end{aligned}$$

- Exprimând primul termen din dreapta obținem

$$\|f - \hat{\varphi}_n\| = \left\{ \|f\|^2 - \sum_{j=1}^n |\hat{c}_j| \|\pi_j\|^2 \right\}^{1/2}, \quad \hat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}. \quad (25)$$

- De notat că expresia dintre acolade trebuie să fie nenegativă.

# Eroarea în MCMMP VI



**Figura:** Aproximația în sensul celor mai mici pătrate ca proiecție ortogonală

# Convergența I

- Dacă se dă acum o secvență de spații liniare  $\Phi_n, n = 1, 2, 3, \dots$ , avem evident

$$\|f - \hat{\varphi}_1\| \geq \|f - \hat{\varphi}_2\| \geq \|f - \hat{\varphi}_3\| \geq \dots,$$

care rezultă nu numai din (25), dar mai direct din faptul că

$$\Phi_1 \subset \Phi_2 \subset \Phi_3 \subset \dots$$

- Deoarece există o infinitate de astfel de spații, atunci secvența de erori din  $L_2$ , fiind monoton descrescătoare, trebuie să conveargă la o limită. Este limita 0?
- Dacă este aşa, spunem că aproximarea prin metoda celor mai mici pătrate converge în medie pătratică când  $n \rightarrow \infty$ .

## Convergența II

- Este evident din (25) că o condiție necesară și suficientă pentru aceasta este

$$\sum_{j=1}^{\infty} |\hat{c}_j|^2 \|\pi_j\|^2 = \|f\|^2. \quad (26)$$

- Un mod echivalent de a formula convergența este următorul: dându-se  $f$  cu  $\|f\| < \infty$ , adică  $\forall f \in L_{2,d\lambda}$  și dându-se un  $\varepsilon > 0$  arbitrar de mic, există un întreg  $n = n_\varepsilon$  și o funcție  $\varphi^* \in \Phi_n$  astfel încât  $\|f - \varphi^*\| \leq \varepsilon$ .
- O clasă de funcții având această proprietate se numește completă în raport cu norma  $\|\cdot\| = \|\cdot\|_{2,d\lambda}$ . Vom numi relația (26) **relația de completitudine sau relația Parseval-Liapunov**.



Carl Friedrich Gauss 1777-1855  
Matematică, astronomie,  
geodezie, magnetism  
(Ca adolescent în  
Braunschweig a descoperit  
teorema binomială,  
reciprocitatea pătratică, media  
aritmetică-geometrică... )  
1807-1855: Universitatea din  
Göttingen

Adrien Marie Legendre  
(1752-1833) matematician  
francez, analiză (integrale  
eliptice), teoria numerelor,  
geometrie. Descoperitor,  
alături de Gauss, (în 1805) al  
metodei celor mai mici  
pătrate, deși Gauss a utilizat  
metoda încă din 1794, dar a  
publicat-o doar în 1809.

# Exemple de sisteme ortogonale

- ① Sistemul trigonometric – cunoscut din analiza Fourier.
- ② Polinoame ortogonale

# Sistemul trigonometric I

- *Sistemul trigonometric* este format din funcțiile:

$$1, \cos t, \cos 2t, \cos 3t, \dots, \sin t, \sin 2t, \sin 3t, \dots$$

- El este ortogonal pe  $[0, 2\pi]$  în raport cu măsura

$$d\lambda(t) = \begin{cases} dt & \text{pe } [0, 2\pi] \\ 0 & \text{în rest} \end{cases}$$

$$\int_0^{2\pi} \sin kt \sin \ell t dt = \begin{cases} 0, & \text{pentru } k \neq \ell \\ \pi, & \text{pentru } k = \ell \end{cases} \quad k, \ell = 1, 2, 3, \dots$$

$$\int_0^{2\pi} \cos kt \cos \ell t dt = \begin{cases} 0, & k \neq \ell \\ 2\pi, & k = \ell = 0 \\ \pi, & k = \ell > 0 \end{cases} \quad k, \ell = 0, 1, 2$$

$$\int_0^{2\pi} \sin kt \cos \ell t dt = 0, \quad k = 1, 2, 3, \dots, \quad \ell = 0, 1, 2, \dots$$

## Sistemul trigonometric II

- Aproximarea are forma

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt). \quad (27)$$

- Utilizând (21) obținem

$$\begin{aligned} a_k &= \frac{1}{\pi} \int_0^{2\pi} f(t) \cos kt dt, \quad k = 1, 2, \dots \\ b_k &= \frac{1}{\pi} \int_0^{2\pi} f(t) \sin kt dt, \quad k = 1, 2, \dots \end{aligned} \quad (28)$$

numiți **coeficienți Fourier** ai lui  $f$ . Ei sunt coeficienții (21) pentru sistemul trigonometric.

- Prin extensie coeficienții (21) pentru orice sistem ortogonal  $(\pi_j)$  se vor numi coeficienții Fourier ai lui  $f$  relativ la acest sistem.

# Sistemul trigonometric III

- În particular, recunoaștem în seria Fourier trunchiată pentru  $k = n$  aproximarea lui  $f$  în clasa polinoamelor trigonometrice de grad  $\leq n$  relativ la norma

$$\|u\|_2 = \left( \int_0^{2\pi} |u(t)|^2 dt \right)^{1/2}$$

# Polinoame ortogonale I

- Dându-se o măsură  $d\lambda$ , stim că orice număr finit de puteri  $1, t, t^2, \dots$  sunt liniar independente pe  $[a, b]$ , dacă  $\text{supp } d\lambda = [a, b]$ , iar  $1, t, \dots, t^{N-1}$  liniar independente pe  $\text{supp } d\lambda = \{t_1, t_2, \dots, t_N\}$ .
- Deoarece o mulțime de vectori liniar independenți a unui spațiu liniar poate fi ortogonalizată prin procedeul Gram-Schmidt, orice măsură  $d\lambda$  de tipul considerat generează o mulțime unică de polinoame ortogonale monice  $\pi_j(t, d\lambda)$ ,  $j = 0, 1, 2, \dots$  ce satisfac

$$\text{grad } \pi_j = j, \quad j = 0, 1, 2, \dots$$

$$\int_{\mathbb{R}} \pi_k(t) \pi_\ell(t) d\lambda(t) = 0, \text{ dacă } k \neq \ell \quad (29)$$

- Aceste polinoame se numesc **polinoame ortogonale** relativ la măsura  $d\lambda$ .

## Polinoame ortogonale II

- Vom permite indicilor să meargă de la 0. Multimea  $\pi_j$  este infinită dacă  $\text{supp}d\lambda = [a, b]$  și constă din exact  $N$  polinoame  $\pi_0, \pi_1, \dots, \pi_{N-1}$  dacă  $\text{supp}d\lambda = \{t_1, \dots, t_N\}$ . În ultimul caz polinoamele se numesc **polinoame ortogonale discrete**.
- Între trei polinoame ortogonale monice (un polinom se numește **monic** dacă coeficientul său dominant este 1) consecutive există o relație liniară. Mai exact, există constantele reale  $\alpha_k = \alpha_k(d\lambda)$  și  $\beta_k = \beta_k(d\lambda) > 0$  (depinzând de măsura  $d\lambda$ ) astfel încât

$$\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0, 1, 2, \dots \quad (30)$$

$$\pi_{-1}(t) = 0, \quad \pi_0(t) = 1.$$

(Se subînțelege că (30) are loc pentru orice  $k \in \mathbb{N}$  dacă  $\text{supp}d\lambda = [a, b]$  și numai pentru  $k = \overline{0, N-2}$  dacă  $\text{supp}d\lambda = \{t_1, t_2, \dots, t_N\}$ ).

## Polinoame ortogonale III

- Pentru a demonstra (30) și a obține expresiile coeficienților să observăm că  $\pi_{k+1}(t) - t\pi_k(t)$  este un polinom de grad  $\leq k$ , și deci poate fi exprimat ca o combinație liniară a lui  $\pi_0, \pi_1, \dots, \pi_k$ . Scriem această combinație sub forma

$$\pi_{k+1} - t\pi_k(t) = -\alpha_k \pi_k(t) - \beta_k \pi_{k-1}(t) + \sum_{j=0}^{k-2} \gamma_{k,j} \pi_j(t) \quad (31)$$

(sumele vide se consideră nule).

- Înmulțim scalar ambii membri ai relației anterioare cu  $\pi_k$  și obținem

$$(-t\pi_k, \pi_k) = -\alpha_k (\pi_k, \pi_k)$$

adică

$$\alpha_k = \frac{(t\pi_k, \pi_k)}{(\pi_k, \pi_k)}, \quad k = 0, 1, 2, \dots \quad (32)$$

## Polinoame ortogonale IV

- La fel, înmulțind scalar cu  $\pi_{k-1}$  obținem

$$(-t\pi_k, \pi_{k-1}) = -\beta_k(\pi_{k-1}, \pi_{k-1}).$$

Deoarece  $(t\pi_k, \pi_{k-1}) = (\pi_k, t\pi_{k-1})$  și  $t\pi_{k-1}$  diferă de  $\pi_k$  printr-un polinom de grad  $< k$  se obține prin ortogonalitate

$$(\pi_k, \pi_{k-1}) = (\pi_k, \pi_k), \text{ deci}$$

$$\beta_k = \frac{(\pi_k, \pi_k)}{(\pi_{k-1}, \pi_{k-1})}, \quad k = 1, 2, \dots \quad (33)$$

- Înmulțind (31) cu  $\pi_\ell$ ,  $\ell < k - 1$ , se obține

$$\gamma_{k,\ell} = 0, \quad \ell = 0, 1, \dots, k - 1 \quad (34)$$

## Polinoame ortogonale V

- Formula de recurență (30) ne dă o modalitate practică de determinare a polinoamelor ortogonale. Deoarece  $\pi_0 = 1$ , putem calcula  $\alpha_0$  cu (32) pentru  $k = 0$  și apoi  $\pi_1$ , etc. Procedeul – numit **procedura lui Stieltjes** – este foarte potrivit pentru polinoame ortogonale discrete, căci în acest caz produsul scalar se exprimă prin sume finite.
- În cazul continuu, calculul produsului scalar necesită calcul de integrale, ceea ce complica lucrurile. Din fericire, pentru multe măsuri speciale importante, coeficienții se cunosc explicit.
- Cazul special când măsura este simetrică (adică  $d\lambda(t) = w(t)$  cu  $w(-t) = w(t)$  și  $\text{supp } d\lambda$  simetrică față de origine) merită o atenție specială, deoarece în acest caz  $\alpha_k = 0$ ,  $\forall k \in \mathbb{N}$ , conform lui (27) căci

$$(t\pi_k, \pi_k) = \int_{\mathbb{R}} w(t)t\pi_k^2(t)dt = \int_a^b w(t)t\pi_k^2(t)dt = 0,$$

## Polinoame ortogonale VI

deoarece avem o integrală dintr-o funcție impară pe un domeniu simetric.



Figura: Thomas Ioannes Stieltjes (1856-1894)

# Polinoamele lui Legendre I

- Se definesc prin aşa-numita formulă a lui Rodrigues

$$\pi_k(t) = \frac{k!}{(2k)!} \frac{d^k}{dt^k} (t^2 - 1)^k. \quad (35)$$

- Exemple:

$$\pi_0(t) = 1,$$

$$\pi_1(t) = t,$$

$$\pi_2(t) = t^2 - \frac{1}{3},$$

$$\pi_3(t) = t^3 - \frac{3}{5}t.$$

- Verificăm întâi ortogonalitatea pe  $[-1, 1]$  în raport cu măsura  $d\lambda(t) = dt$ .

## Polinoamele lui Legendre II

- Pentru orice  $0 \leq \ell < k$ , prin integrare repetată prin părți se obține:

$$\int_{-1}^1 \frac{d^k}{dt^k} (t^2 - 1)^k t^\ell dt$$
$$= \sum_{m=0}^{\ell} \ell(\ell - 1) \dots (\ell - m + 1) t^{\ell-m} \frac{d^{k-m-1}}{dt^{k-m-1}} (t^2 - 1)^k \Big|_{-1}^1 = 0,$$

ultima relație având loc deoarece  $0 \leq k - m - 1 < k$ .

- Deci,

$$(\pi_k, p) = 0, \quad \forall p \in \mathbb{P}_{k-1},$$

demonstrându-se astfel ortogonalitatea.

# Polinoamele lui Legendre III

- **Relația de recurență**
- Datorită simetriei, putem scrie

$$\pi_k(t) = t^k + \mu_k t^{k-2} + \dots, \quad k \geq 2$$

și observând (din nou datorită simetriei) că relația de recurență are forma

$$\pi_{k+1}(t) = t\pi_k(t) - \beta_k \pi_{k-1}(t),$$

obținem

$$\beta_k = \frac{t\pi_k(t) - \pi_{k+1}(t)}{\pi_{k-1}(t)},$$

care este valabilă pentru orice  $t$ .

## Polinoamele lui Legendre IV

- Făcând  $t \rightarrow \infty$ ,

$$\beta_k = \lim_{t \rightarrow \infty} \frac{t\pi_k(t) - \pi_{k+1}(t)}{\pi_{k-1}(t)} = \lim_{t \rightarrow \infty} \frac{(\mu_k - \mu_{k+1})t^{k-1} + \dots}{t^{k-1} + \dots} = \mu_k - \mu_{k+1}.$$

(Dacă  $k = 1$ , punem  $\mu_1 = 0$ .)

- Din formula lui Rodrigues rezultă

$$\begin{aligned}\pi_k(t) &= \frac{k!}{(2k)!} \frac{d^k}{dt^k} \left( t^{2k} - kt^{2k-2} + \dots \right) \\ &= \frac{k!}{(2k)!} (2k(2k-1)\dots(k+1)t^k - \\ &\quad k(2k-2)(2k-3)\dots(k-1)t^{k-1} + \dots) \\ &= t^k - \frac{k(k-1)}{2(2k-1)} t^{k-2} + \dots,\end{aligned}$$

## Polinoamele lui Legendre V

aşa că

$$\mu_k = \frac{k(k-1)}{2(2k-1)}, \quad k \geq 2.$$

Deci,

$$\beta_k = \mu_k - \mu_{k+1} = \frac{k^2}{(2k-1)(2k+1)}$$

şi deoarece  $\mu_1 = 0$ ,

$$\beta_k = \frac{1}{4 - k^{-2}}, \quad k \geq 1. \tag{36}$$

## Polinoamele Cebîșev de speță I I

- Ele se pot defini prin relația

$$T_n(x) = \cos(n \arccos x), \quad n \in \mathbb{N}. \quad (37)$$

- Din identitatea trigonometrică

$$\cos((k+1)\theta) + \cos((k-1)\theta) = 2 \cos \theta \cos k\theta$$

și din (37), punând  $\theta = \arccos x$  se obține

$$\begin{aligned} T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x) \quad k = 1, 2, 3, \dots \\ T_0(x) &= 1, \quad T_1(x) = x. \end{aligned} \quad (38)$$

# Polinoamele Cebîșev de speță I II

- De exemplu,

$$T_2(x) = 2x^2 - 1,$$

$$T_3(x) = 4x^3 - 3x,$$

$$T_4(x) = 8x^4 - 8x^2 + 1$$

s.a.m.d.

- Din relația (38) se obține pentru coeficientul dominant al lui  $T_n$  valoarea  $2^{n-1}$  (dacă  $n \geq 1$ ), deci polinomul Cebîșev de speță I monic este

$$\overset{\circ}{T}_n(x) = \frac{1}{2^{n-1}} T_n(x), \quad n \geq 0, \quad \overset{\circ}{T}_0 = T_0. \quad (39)$$

- Din (37) se pot obține rădăcinile lui  $T_n$

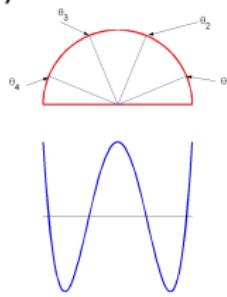
$$x_k^{(n)} = \cos \theta_k^{(n)}, \quad \theta_k^{(n)} = \frac{2k-1}{2n}\pi, \quad k = \overline{1, n}. \quad (40)$$

# Polinoamele Cebîșev de speță I III

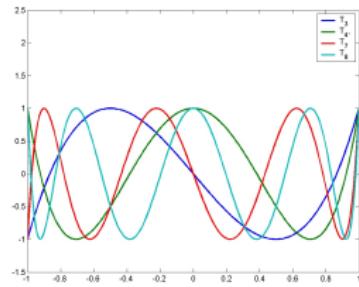
- Ele sunt proiecțiile pe axa reală ale punctelor de pe cercul unitate de argument  $\theta_k^{(n)}$ .
- Pe intervalul  $[-1, 1]$   $T_n$  oscilează de la +1 la -1, atingând aceste valori extreme în punctele

$$y_k^{(n)} = \cos \eta_k^{(n)}, \quad \eta_k^{(n)} = \frac{k\pi}{n}, \quad k = \overline{0, n}.$$

$T_4$  și rădăcinile sale



$T_3, T_4, T_7, T_8$  pe  $[-1, 1]$



# Polinoamele Cebîșev de speță I IV

- Polinoamele Cebîșev de speță I sunt ortogonale în raport cu măsura

$$d\lambda(x) = \frac{dx}{\sqrt{1-x^2}}, \quad \text{pe } [-1, 1].$$

- Se verifică ușor din (37) că

$$\begin{aligned} & \int_{-1}^1 T_k(x) T_\ell(x) \frac{dx}{\sqrt{1-x^2}} = \int_0^\pi T_k(\cos \theta) T_\ell(\cos \theta) d\theta \\ &= \int_0^\pi \cos k\theta \cos \ell\theta d\theta = \begin{cases} 0 & \text{dacă } k \neq \ell \\ \pi & \text{dacă } k = \ell = 0 \\ \pi/2 & \text{dacă } k = \ell \neq 0 \end{cases} \quad (41) \end{aligned}$$

# Polinoamele Cebîșev de speță I V

- Dezvoltarea în serie Fourier de polinoame Cebîșev este dată de

$$f(x) = \sum_{j=0}^{\infty}' c_j T_j(x) = \frac{1}{2}c_0 + \sum_{j=1}^{\infty} c_j T_j(x), \quad (42)$$

unde

$$c_j = \frac{2}{\pi} \int_{-1}^1 f(x) T_j(x) \frac{dx}{\sqrt{1-x^2}}, \quad j \in \mathbb{N}.$$

- Păstrând în (42) numai termenii de grad cel mult  $n$  se obține o aproximare polinomială utilă de grad  $n$

$$\tau_n(x) = \sum_{j=0}^n' c_j T_j(x), \quad (43)$$

având eroarea

$$f(x) - \tau_n(x) = \sum_{j=n+1}^{\infty} c_j T_j(x) \approx c_{n+1} T_{n+1}(x). \quad (44)$$

# Polinoamele Cebîșev de speță I VI

- Aproximanta din (43) este cu atât mai bună cu cât coeficienții din extremitatea dreaptă tind mai repede către zero. Eroarea (44) oscilează în esență între  $+c_{n+1}$  și  $-c_{n+1}$  și este deci de mărime „uniformă”. Acest lucru contrastează puternic cu dezvoltarea Taylor în jurul lui  $x = 0$ , unde polinomul de grad  $n$  are eroarea proporțională cu  $x^{n+1}$  pe  $[-1, 1]$ .

# Minimalitatea normei I

## Teoremă

Pentru orice polinom monic  $\overset{\circ}{p}_n$  de grad  $n$  are loc

$$\max_{-1 \leq x \leq 1} \left| \overset{\circ}{p}_n(x) \right| \geq \max_{-1 \leq x \leq 1} \left| \overset{\circ}{T}_n(x) \right| = \frac{1}{2^{n-1}}, \quad n \geq 1, \quad (45)$$

unde  $\overset{\circ}{T}_n(x)$  este dat de (39).

**Demonstrație.** Se face prin reducere la absurd. Presupunem că

$$\max_{-1 \leq x \leq 1} \left| \overset{\circ}{p}_n(x) \right| < \frac{1}{2^{n-1}}. \quad (46)$$

## Minimalitatea normei II

Atunci polinomul  $d_n(x) = \overset{\circ}{T}_n(x) - \overset{\circ}{p}_n(x)$  (de grad  $\leq n - 1$ ) satisface

$$d_n(y_0^{(n)}) > 0, \quad d_n(y_1^{(n)}) < 0, \quad d_n(y_2^{(n)}) > 0, \dots, (-1)^n d_n(y_n^{(n)}) > 0. \quad (47)$$

Deoarece  $d_n$  are  $n$  schimbări de semn, el este identic nul; aceasta contrazice (47) și astfel (46) nu poate fi adevărată. ■

Rezultatul (45) se poate interpreta în modul următor: cea mai bună aproximare uniformă din  $\mathbb{P}_{n-1}$  pe  $[-1, 1]$  a lui  $f(x) = x^n$  este dată de  $x^n - \overset{\circ}{T}_n(x)$ , adică, de agregarea termenilor până la gradul  $n - 1$  din  $\overset{\circ}{T}_n$  luate cu semnul minus. Din teoria aproximățiilor uniforme se știe că cea mai bună aproximare polinomială uniformă este unică. Deci, egalitatea în (45) poate avea loc numai dacă  $\overset{\circ}{p}_n(x) = \overset{\circ}{T}_n(x)$ .

## Polinoamele Cebîșev de speță a II-a

- Se definesc prin

$$Q_n(t) = \frac{\sin[(n+1)\arccos t]}{\sqrt{1-t^2}}, \quad t \in [-1, 1]$$

- Ele sunt ortogonale pe  $[-1, 1]$  în raport cu măsura  $d\lambda(t) = w(t)dt$ ,  
 $w(t) = \sqrt{1-t^2}$ .
- Relația de recurență este

$$Q_{n+1}(t) = 2tQ_n(t) - Q_{n-1}(t), \quad Q_0(t) = 1, \quad Q_1(t) = 2t.$$



Figura: Pafnuti Levovici Cebîșev (1821-1894)

# Polinoamele lui Laguerre I

- Sunt ortogonale pe  $[0, \infty)$  în raport cu ponderea  $w(t) = t^\alpha e^{-t}$ .
- Se definesc prin

$$\ell_n^\alpha(t) = \frac{e^t t^{-\alpha}}{n!} \frac{d^n}{dt^n}(t^{n+\alpha} e^{-t}) \text{ pentru } \alpha > 1$$

- Relația de recurență pentru polinoamele monice este

$$\ell_{k+1}^\alpha(t) = (t - 2k - \alpha - 1)\ell_k^\alpha(t) - \beta_k \ell_{k-1}^\alpha(t),$$

unde

$$\beta_k = \begin{cases} \Gamma(1 + \alpha), & \text{pentru } k = 0; \\ k(k + \alpha), & \text{pentru } k > 0. \end{cases}$$

# Polinoamele lui Laguerre II

- Exemple pentru  $\alpha = 0$ :

$$\ell_0^{(0)}(t) = 1,$$

$$\ell_1^{(0)}(t) = t - 1,$$

$$\ell_2^{(0)}(t) = t^2 - 4t + 2,$$

$$\ell_3^{(0)}(t) = t^3 - 9t^2 + 18t - 6$$



Figura: Edmond Laguerre (1834-1886)

# Polinoamele lui Hermite I

- Se definesc prin

$$H_n(t) = (-1)^n e^{t^2} \frac{d^n}{dt^n}(e^{-t^2}).$$

- Ele sunt ortogonale pe  $(-\infty, \infty)$  în raport cu ponderea  $w(t) = e^{-t^2}$  și verifică relația de recurență

$$H_{k+1}(t) = tH_k(t) - \beta_k H_{n-1}(t)$$

unde

$$\beta_k = \begin{cases} \sqrt{\pi}, & \text{pentru } k = 0; \\ \frac{k}{2}, & \text{pentru } k > 0. \end{cases}$$

# Polinoamele lui Hermite II

- Exemple:

$$H_0(t) = 1,$$

$$H_1(t) = t,$$

$$H_2(t) = t^2 - \frac{1}{2},$$

$$H_3(t) = t^3 - \frac{3}{2}t.$$



**Figura:** Charles Hermite (1822-1901)

# Polinoamele lui Jacobi I

- Sunt ortogonale pe  $[-1, 1]$  în raport cu ponderea

$$w(t) = (1-t)^\alpha (1+t)^\beta.$$

- Coeficienții din relația de recurență sunt

$$\alpha_k = \frac{\beta^2 - \alpha^2}{(2k + \alpha + \beta)(2k + \alpha + \beta + 2)}$$

și

$$\beta_0 = 2^{\alpha+\beta+1} B(\alpha+1, \beta+1),$$

$$\beta_k = \frac{4k(k+\alpha)(k+\alpha+\beta)(k+\beta)}{(2k+\alpha+\beta-1)(2k+\alpha+\beta)^2(2k+\alpha+\beta+1)}, \quad k > 0.$$

## Polinoamele lui Jacobi II

- Exemple pentru  $\alpha = 1/2$  și  $\beta = -1/2$

$$\pi_0^{(\alpha, \beta)}(t) = 1,$$

$$\pi_1^{(\alpha, \beta)}(t) = t,$$

$$\pi_2^{(\alpha, \beta)}(t) = t^2 + \frac{1}{2}t - \frac{1}{4},$$

$$\pi_3^{(\alpha, \beta)}(t) = t^3 + \frac{1}{2}t^2 - \frac{1}{2}t - \frac{1}{8}.$$



Figura: Carl Gustav Jacob Jacobi (1804-1851)

## Exemplu

Pentru funcția  $f(t) = \arccos t$ ,  $t \in [-1, 1]$ , obțineți aproximanta în sensul celor mai mici pătrate,  $\hat{\varphi} \in P_n$  a lui  $f$  relativ la funcția pondere

$w(t) = (1 - t^2)^{-\frac{1}{2}} = \frac{1}{\sqrt{1-t^2}}$  adică, găsiți soluția  $\varphi = \hat{\varphi}$  a problemei

$$\min \left\{ \int_{-1}^1 [f(t) - \varphi(t)]^2 \frac{dt}{\sqrt{1-t^2}} : \varphi \in P_n \right\}.$$

Exprimați  $\varphi$  cu ajutorul polinoamelor Cebîșev  $\pi_j(t) = T_j(t)$ .

**Soluție.**  $\hat{\varphi}(t) = \frac{c_0}{2} + c_1 T_1(x) + \cdots + c_n T_n(x)$

$$c_k = \frac{(f, T_k)}{(T_k, T_k)} = \frac{2}{\pi} (f, T_k) = \frac{2}{\pi} \int_{-1}^1 \frac{\arccos t}{\sqrt{1-t^2}} \cos(k \arccos t) dt$$

$$= \frac{2}{\pi} \int_0^\pi u \cos k u du = \frac{2}{\pi} \left[ \frac{u \sin k u}{k} \Big|_0^\pi - \frac{1}{k} \int_0^\pi \sin k u du \right]$$

$$= \frac{2}{\pi} \left[ \frac{1}{k} \frac{\cos ku}{k} \Big|_0^\pi \right] = -\frac{2}{\pi k^2} [(-1)^k - 1]$$

$k$  par  $c_k = 0$

$k$  impar  $c_k = -\frac{2}{\pi k^2}(-2) = \frac{4}{\pi k^2}$  ■

# Bibliografie I

-  Å. Björk, *Numerical Methods for Least Squares Problem*, SIAM, Philadelphia, 1996.
-  E. Blum, *Numerical Computing: Theory and Practice*, Addison-Wesley, 1972.
-  P. G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, Paris, Milan, Barcelone, Mexico, 1990.
-  Gheorghe Coman, *Analiză numerică*, Editura Libris, Cluj-Napoca, 1995.
-  W. Gautschi, *Numerical Analysis. An Introduction*, Birkhäuser, Basel, 1997.

## Bibliografie II

- W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, New York, Port Chester, Melbourne, Sidney, 1996, disponibila prin www, <http://www.nr.com/>.
- D. D. Stancu, *Analiză numerică – Curs și culegere de probleme*, Lito UBB, Cluj-Napoca, 1977.
- J. Stoer, R. Burlisch, *Introduction to Numerical Analysis*, 2nd ed., Springer Verlag, 1992.

# Aproximarea funcționalelor liniare

## Derivare și integrare numerică

Radu T. Trîmbițaș

UBB

27 mai 2020

# Aproximarea funcționalelor liniare I

- $X$  un spațiu liniar
- $L_1, \dots, L_m$  funcționale liniare reale, liniar independente, definite pe  $X$
- $L : X \rightarrow \mathbb{R}$  funcțională liniară reală astfel încât  $L, L_1, \dots, L_m$  să fie liniar independente.

## Definiția 1

O formulă de aproximare a funcționalei  $L$  în raport cu funcționalele  $L_1, \dots, L_m$  este o formulă de forma

$$L(f) = \sum_{i=1}^m A_i L_i(f) + R(f), \quad f \in X. \quad (1)$$

Parametrii reali  $A_i$  se numesc coeficienții formulei, iar  $R(f)$  termenul rest.

# Aproximarea funcționalelor liniare II

Pentru o formulă de aproximare de forma (1), dându-se funcționalele  $L_i$ , se pune problema determinării coeficienților  $A_i$  și a studiului termenului rest corespunzător valorilor obținute pentru coeficienți.

## Observația 2

*Forma funcționalelor  $L_i$  depinde de informațiile deținute asupra lui  $f$  (ele exprimând de fapt aceste informații), dar și de natura problemei de aproximare, adică de forma lui  $L$ .*

### Exemplul 3

Dacă  $X = \{f \mid f : [a, b] \rightarrow \mathbb{R}\}$ ,  $L_i(f) = f(x_i)$ ,  $i = \overline{0, m}$ ,  $x_i \in [a, b]$  și  $L(f) = f(\alpha)$ ,  $\alpha \in [a, b]$ , formula de interpolare Lagrange

$$f(\alpha) = \sum_{i=0}^m \ell_i(\alpha) f(x_i) + (Rf)\alpha$$

este o formulă de tip (1), cu coeficienții  $A_i = \ell_i(\alpha)$ , iar una din reprezentările posibile pentru rest este

$$(Rf)(\alpha) = \frac{u(\alpha)}{(m+1)!} f^{(m+1)}(\xi), \quad \xi \in [a, b]$$

dacă există  $f^{(m+1)}$  pe  $[a, b]$ .

## Exemplul 4

Dacă  $X$  și  $L_i$  sunt ca în exemplul 3 și există  $f^{(k)}(\alpha)$ ,  $\alpha \in [a, b]$ ,  $k \in \mathbb{N}^*$ , iar  $L(f) = f^{(k)}(\alpha)$  se obține o formulă de aproximare a valorii derivatei de ordinul  $k$  a lui  $f$  în punctul  $\alpha$

$$f^{(k)}(\alpha) = \sum_{i=0}^m A_i f(x_i) + R(f),$$

numită și **formulă de derivare numerică**.

## Exemplul 5

Dacă  $X$  este un spațiu de funcții definite pe  $[a, b]$ , integrabile pe  $[a, b]$  și pentru care există  $f^{(j)}(x_k)$ ,  $k = \overline{0, m}$ ,  $j \in I_k$ , cu  $x \in [a, b]$  și  $I_k$  mulțimi de indici date

$$L_{kj}(f) = f^{(j)}(x_k), \quad k = \overline{0, m}, \quad j \in I_k,$$

iar

$$L(f) = \int_a^b f(x) dx,$$

se obține formula

$$\int_a^b f(x) dx = \sum_{k=0}^m \sum_{j \in I_k} A_{kj} f^{(j)}(x_k) + R(f),$$

numită **formulă de integrare numerică**.

## Definiția 6

Dacă  $\mathbb{P}_r \subset X$ , numărul  $r \in \mathbb{N}$  cu proprietatea că  $\text{Ker}(R) = \mathbb{P}_r$  se numește **grad de exactitate** al formulei de aproximare (1).

## Observația 7

Deoarece  $R$  este o funcțională liniară proprietatea  $\text{Ker}(R) = \mathbb{P}_r$  este echivalentă cu  $R(e_k) = 0$ ,  $k = \overline{0, r}$  și  $R(e_{r+1}) \neq 0$ , unde  $e_k(x) = x^k$ .

# Problema generală de aproximare I

- *Problema generală de aproximare:* dându-se o funcțională liniară  $L$  pe  $X$ ,  $m$  funcționale liniare  $L_1, L_2, \dots, L_m$  pe  $X$  și valorile lor ("datele")  $\ell_i = L_i f$ ,  $i = \overline{1, m}$  aplicate unei anumite funcții  $f$  și un subspațiu liniar  $\Phi \subset X$  cu  $\dim \Phi = m$ , dorim să găsim o formulă de aproximare de tipul

$$Lf \approx \sum_{i=1}^m a_i L_i f \quad (2)$$

care să fie exactă (adică să aibă loc egalitatea) pentru orice  $f \in \Phi$

- **ipoteză naturală:** „problema de interpolare” să se găsească  $\varphi \in \Phi$  a.î.

$$L_i \varphi = s_i, \quad i = \overline{1, m} \quad (3)$$

să aibă soluție unică  $\varphi(\cdot) = \varphi(s, \cdot)$ , pentru  $s = [s_1, \dots, s_m]^T$  arbitrar.

## Problema generală de aproximare II

- Putem exprima ipoteza noastră mai explicit în termenii unei baze date  $\varphi_1, \varphi_2, \dots, \varphi_m$  a lui  $\Phi$  și a matricei Gram asociate

$$G = [L_i \varphi_j] = \begin{bmatrix} L_1 \varphi_1 & L_1 \varphi_2 & \dots & L_1 \varphi_m \\ L_2 \varphi_1 & L_2 \varphi_2 & \dots & L_2 \varphi_1 \\ \vdots & \vdots & \ddots & \vdots \\ L_m \varphi_1 & L_m \varphi_2 & \dots & L_m \varphi_m \end{bmatrix} \in \mathbb{R}^{m \times m} \quad (4)$$

- Cerem ca

$$\det G \neq 0 \quad (5)$$

- Solvabilitatea unică a lui (3) și (5) sunt echivalente
- Scriem  $\varphi$  din (3) sub forma

$$\varphi = \sum_{j=1}^m c_j \varphi_j$$

## Problema generală de aproximare III

- condițiile de interpolare

$$L_i \left( \sum_{j=1}^m c_j \varphi_j \right) = s_i, \quad i = 1, \dots, m$$

pot fi scrise sub forma

$$\sum_{j=1}^m c_j L_i \varphi_j = s_i, \quad i = 1, \dots, m,$$

adică

$$Gc = s, \quad c = [c_1, \dots, c_m]^T, \quad s = [s_1, \dots, s_m]^T \quad (6)$$

- (6) are soluție unică  $\iff$  (5)
- abordări: metoda interpolării și metoda coeficienților nedeterminați



Figura: Jórgen Pedersen Gram (1850-1916)

# Metoda interpolării I

- Rezolvăm problema generală de interpolare prin aproximare

$$Lf \approx L\varphi(\ell, .), \quad \ell = [\ell_1, \dots, \ell_m]^T, \quad \ell_i = L_i f \quad (7)$$

- Aplicăm  $L$  nu lui  $f$  ci soluției  $\varphi(\ell, .)$  a lui (3) în care  $s = \ell$ . Ipoteza noastră ne garantează că  $\varphi(\ell, .)$  este unic determinat. În particular, dacă  $f \in \Phi$ , atunci (7) are loc cu egalitate, deoarece  $\varphi(\ell, .) = f(.$ ), în mod trivial. Astfel, aproximanta noastră (7) satisface condițiile de exactitate cerute pentru (2). Rămîne doar să arătăm că (7) produce o aproximare de forma (2).

## Metoda interpolării II

- Pentru aceasta să observăm că interpolantul în (7) este

$$\varphi(\ell, \cdot) = \sum_{j=1}^m c_j \varphi_j(\cdot)$$

unde  $c = [c_1, \dots, c_m]^T$  satisfacă (6) cu  $s = \ell$

$$Gc = \ell, \quad \ell = [L_1 f, L_2 f, \dots, L_m f]^T.$$

- Scriind

$$\lambda_j = L\varphi_j, \quad j = \overline{1, m}, \quad \lambda = [\lambda_1, \lambda_2, \dots, \lambda_m]^T, \quad (8)$$

avem din liniaritatea lui  $L$

$$L\varphi(\ell, \cdot) = \sum_{j=1}^m c_j L\varphi_j = \lambda^T c = \lambda^T G^{-1} \ell = \left[ (G^T)^{-1} \lambda \right]^T \ell,$$

# Metoda interpolării III

adică

$$L\varphi(\ell, \cdot) = \sum_{i=1}^m a_i L_i f, \quad a = [a_1, \dots, a_m]^T = (G^T)^{-1} \lambda. \quad (9)$$

# Metoda coeficienților nedeterminați I

- Aici determinăm coeficienții din (2) astfel încât egalitatea să aibă loc  $\forall f \in \Phi$ , care conform liniarității lui  $L$  și  $L_i$  este echivalentă cu egalitatea pentru  $f = \varphi_1, f = \varphi_2, \dots, f = \varphi_m$ , adică

$$\left( \sum_{j=1}^m a_j L_j \right) \varphi_i = L \varphi_i, \quad i = \overline{1, m},$$

sau conform (7)

$$\sum_{j=1}^m a_j L_j \varphi_i = \lambda_i, \quad i = \overline{1, m}.$$

- Matricea sistemului este  $G^T$ , deci

$$a = [a_1, \dots, a_m]^T = (G^T)^{-1} \lambda,$$

în concordanță cu (9).

## Metoda coeficienților nedeterminați II

- Astfel, metoda interpolării și cea a coeficienților nedeterminați sunt matematic echivalente – ele conduc la exact aceeași aproximare.
- S-ar părea că, cel puțin în cazul polinoamelor (adică  $\Phi = \mathbb{P}_d$ ), prima metodă este mai puternică, deoarece poate conduce la o expresie a erorii de interpolare (aplicând funcționala formulei de interpolare  $f = a_n f + r_n f$ ). Dar și în cazul metodei coeficienților nedeterminați, din condiția de exactitate, se poate exprima restul cu ajutorul teoremei lui Peano.

# Derivare numerică I

- Vom considera doar derivata de ordinul I; pentru deriveate de ordin superior vom utiliza tehnici analoage.
- Rezolvare prin interpolare: în loc să derivăm  $f \in C^{m+1}[a, b]$ , vom deriva polinomul său de interpolare:

$$f(x) = (L_m f)(x) + (R_m f)(x). \quad (10)$$

- Scriem polinomul de interpolare în forma Newton

$$\begin{aligned} (L_m f)(x) &= (N_m f)(x) = f_0 + (x - x_0)f[x_0, x_1] + \cdots + \\ &\quad + (x - x_0) \dots (x - x_{m-1})f[x_0, x_1, \dots, x_m] \end{aligned} \quad (11)$$

- restul sub forma

$$(R_m f)(x) = (x - x_0) \dots (x - x_m) \frac{f^{(m+1)}(\xi(x))}{(m+1)!}. \quad (12)$$

## Derivare numerică II

- Derivând (11) în raport cu  $x$  și punând  $x = x_0$  obținem

$$(L_m f)'(x_0) = f[x_0, x_1] + (x_0 - x_1)f[x_0, x_1, x_2] + \cdots + (x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_{m-1})f[x_0, x_1, \dots, x_m]. \quad (13)$$

- Presupunând că  $f$  este continuu derivabilă pe un interval convenabil se obține pentru rest

$$(R_m f)'(x_0) = (x_0 - x_1) \cdots (x_0 - x_m) \frac{f^{(m+1)}(\xi(x_0))}{(m+1)!}. \quad (14)$$

- Deci, derivând (13) avem

$$f'(x_0) = (L_m f)'(x_0) + \underbrace{(R_m f)'(x_0)}_{e_m}. \quad (15)$$

- Dacă  $H = \max_i |x_0 - x_i|$  eroarea are forma  $e_m = O(H^m)$ , când  $H \rightarrow 0$ .

## Observația 8

- Derivarea numerică este o operație critică și de aceea este bine să fie evitată pe cât posibil, deoarece chiar dacă aproximanta este bună, nu rezultă că derivata aproximantei este o aproximare bună a derivatei (vezi figura 2).
- Aceasta rezultă și din exemplul 9.
- Formulele de derivare numerică sunt utile pentru deducerea unor metode numerice, în special pentru ecuații diferențiale ordinare și ecuații cu derive parțiale.

# Derivare numerică IV

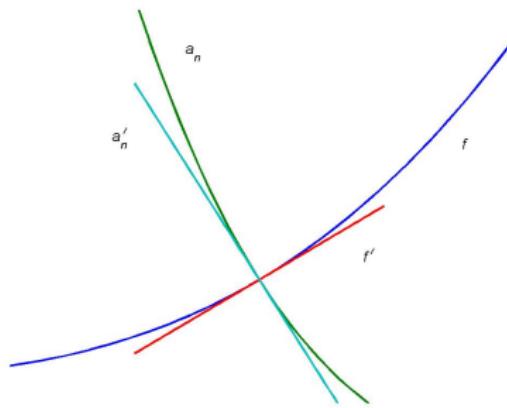


Figura: Neajunsurile derivării numerice

## Exemplul 9

Fie funcția

$$f(x) = g(x) + \frac{1}{n} \sin n^2(x - a), \quad g \in C^1[a, b].$$

Se constată că  $d(f, g) \rightarrow 0$  ( $n \rightarrow \infty$ ), dar  $d(f', g') = n \not\rightarrow 0$ .

# Integrare numerică I

Fie  $f : [a, b] \rightarrow \mathbb{R}$  integrabilă pe  $[a, b]$ ,  $F_k(f)$ ,  $k = \overline{0, m}$  informații despre  $f$  (de regulă funcționale liniare) și  $w : [a; b] \rightarrow \mathbb{R}_+$  o funcție pondere integrabilă pe  $[a, b]$ .

# Integrare numerică II

## Definiția 10

O formulă de forma

$$\int_a^b w(x)f(x)dx = Q(f) + R(f), \quad (16)$$

unde

$$Q(f) = \sum_{j=0}^m A_j F_j f$$

se numește **formulă de integrare numerică** a funcției  $f$  sau **formulă de cuadratură**. Parametrii  $A_j$ ,  $j = \overline{0, m}$  se numesc **coeficienții formulei**, iar  $R(f)$  **termenul rest** al ei.  $Q$  se numește **funcțională de cuadratură**.

## Definiția 11

*Numărul natural  $d = d(Q)$  cu proprietatea că  $\forall f \in \mathbb{P}_d$ ,  $R(f) = 0$  și  $\exists g \in \mathbb{P}_{d+1}$  astfel încât  $R(g) \neq 0$  se numește **grad de exactitate** al formulei de cuadratură.*

Deoarece  $R$  este liniar, rezultă că o formulă de cuadratură are gradul de exactitate  $d$  dacă și numai dacă  $R(e_j) = 0$ ,  $j = \overline{0, d}$  și  $R(e_{d+1}) \neq 0$ . Dacă gradul de exactitate al unei formule de cuadratură este cunoscut, restul se poate determina cu ajutorul teoremei lui Peano.

# Formula trapezului și formula lui Simpson

- Aceste formule au fost denumite de Gautschi în [2] „căii de povară” ai integrării numerice.
- Ele își fac bine munca când intervalul de integrare este mărginit și integrandul este neproblematic. Formula trapezelor este surprinzător de eficientă chiar și pentru intervale infinite.
- Ambele reguli se obțin aplicând cele mai simple tipuri de interpolare subintervalelor diviziunii - interpolare Lagrange gradul I pentru trapeze, 2 pentru Simpson

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b, \quad x_k = a + kh \quad (17)$$

$$h = \frac{b - a}{n}$$

# Formula trapezului I

- Se interpolează liniar pe fiecare subinterval  $[x_k, x_{k+1}]$  și se obține

$$\int_{x_k}^{x_{k+1}} f(x) dx = \int_{x_k}^{x_{k+1}} (L_1 f)(x) dx + \int_{x_k}^{x_{k+1}} (R_1 f)(x) dx, \quad f \in C^2[a, b]$$

(18)

cu

$$(L_1 f)(x) = f_k + (x - x_k) f [x_k, x_{k+1}].$$

- Integrând avem

$$\int_{x_k}^{x_{k+1}} f(x) dx = \frac{h}{2} (f_k + f_{k+1}) + R_1(f),$$

unde

$$R_1(f) = \int_{x_k}^{x_{k+1}} K_1(t) f''(t) dt$$

# Formula trapezului II

- Nucleul lui Peano

$$\begin{aligned}K_1(t) &= \frac{(x_{k+1} - t)^2}{2} - \frac{h}{2} [(x_k - t)_+ + (x_{k+1} - t)_+] \\&= \frac{(x_{k+1} - t)^2}{2} - \frac{h(x_{k+1} - t)}{2} \\&= \frac{1}{2} (x_{k+1} - t) (x_k - t) \leq 0.\end{aligned}$$

- Deci

$$R_1(f) = -\frac{h^3}{12} f''(\xi_k), \quad \xi_k \in [x_k, x_{k+1}]$$

și

$$\int_{x_k}^{x_{k+1}} f(x) dx = \frac{h}{2} (f_k + f_{k+1}) - \frac{h^3}{12} f''(\xi_k) \quad (19)$$

- Această formulă se numește *regula (elementară a) trapezului*.

## Formula trapezului III

- Însumând pentru toate subintervalele se obține *regula trapezelor sau formula compusă a trapezului sau formula repetată a trapezului.*

$$\int_a^b f(x)dx = h \left( \frac{1}{2}f_0 + f_1 + \cdots + f_{n-1} + \frac{1}{2}f_n \right) - \frac{h^3}{12} \sum_{k=0}^{n-1} f''(\xi_k)$$

Deoarece  $f''$  este continuă pe  $[a, b]$ , restul se poate scrie sub forma

$$R_{1,n}(f) = -\frac{(b-a)h^2}{12} f''(\xi) = -\frac{(b-a)^3}{12n^2} f''(\xi) \quad (20)$$

Deoarece  $f''$  este mărginită în modul

$$R_{1,n}(f) = O(h^2), \quad h \rightarrow 0,$$

și deci regula trapezelor converge când  $h \rightarrow 0$  (sau echivalent,  $n \rightarrow \infty$ ), atunci când  $f \in C^2[a; b]$ .

# Formula lui Simpson I

- Dacă în locul interpolării liniare se utilizează interpolarea pătratică se obține *regula lui Simpson repetată*.
- Varianta ei elementară, numită *regula lui Simpson* sau *formula lui Simpson* este

$$\int_{x_k}^{x_{k+2}} f(x) dx = \frac{h}{3} (f_k + 4f_{k+1} + f_{k+2}) - \frac{h^5}{90} f^{(4)}(\xi_k), \quad \xi_k \in [x_k, x_{k+2}],$$

(21)

$$f \in C^4[a, b].$$

## Formula lui Simpson II

- Restul -  $dex = 3$ , aplicăm teorema lui Peano

$$R_2(f) = \int_{x_k}^{x_{k+2}} K_2(t) f^{(4)}(t) dt,$$

unde

$$K_2(t) = \frac{1}{3!} \left\{ \frac{(x_{k+1}-t)^4}{4} - \frac{h}{3} \left[ (x_k-t)_+^3 + 4(x_{k+1}-t)_+^3 + (x_{k+2}-t)_+^3 \right] \right\},$$

$$K_2(t) = \begin{cases} \frac{(x_{k+1}-t)^4}{4} - \frac{h}{3} \left[ 4(x_{k+1}-t)^3 + (x_{k+2}-t)^3 \right], & t \in [x_k, x_{k+1}] \\ \frac{(x_{k+1}-t)^4}{4} - \frac{h}{3} (x_{k+2}-t)^3, & t \in [x_{k+1}, x_{k+2}] \end{cases}$$

## Formula lui Simpson III

- $t \in [a, b] \implies K_2(t) \leq 0$ , putem aplica corolarul la th. Peano

$$R_2(f) = \frac{1}{4!} f^{(4)}(\xi_k) R_2(e_4),$$

$$\begin{aligned} R_2(e_4) &= \frac{x_{k+2}^5 - x_k^5}{5} - \frac{h}{3} [x_k^4 + 4x_{k+1}^4 + x_{k+2}^4] \\ &= h \left[ 2 \frac{x_{k+2}^4 + x_{k+2}^3 x_k + x_{k+2}^2 x_k^2 + x_{k+2} x_k^3 + x_k^4}{5} \right. \\ &\quad \left. - \frac{5x_k^4 + 4x_k^3 x_{k+2} + 6x_k^2 x_{k+2}^2 + 4x_k x_{k+2}^3 + 5x_{k+2}^4}{12} \right] \\ &= \frac{h}{60} (-x_k^4 + 4x_k^3 x_{k+2} + 6x_k^2 x_{k+2}^2 + 4x_k x_{k+2}^3 - x_{k+2}^4) \\ &= -\frac{h}{60} (x_{k+2} - x_k)^4 = -4 \frac{h^5}{15} \end{aligned}$$

## Formula lui Simpson IV

- Deci,

$$R_2(f) = -\frac{h^5}{90} f^{(4)}(\xi_k).$$

- Pentru regula repetată a lui Simpson obținem

$$\int_a^b f(x)dx = \frac{h}{3} (f_0 + 4f_1 + 2f_2 + \cdots + 2f_{n-2} + 4f_{n-1} + f_n) + R_{2,n}(f) \quad (22)$$

cu

$$R_{2,n}(f) = -\frac{(b-a)h^4}{180} f^{(4)}(\xi) = -\frac{(b-a)^5}{2880n^4} f^{(4)}(\xi), \quad \xi \in (a, b).$$

- Se observă că  $R_{2,n}(f) = O(h^4)$ , de unde rezultă convergența când  $n \rightarrow \infty$ . Se observă și creșterea ordinului cu 1, ceea ce face ca regula repetată a lui Simpson să fie foarte populară și larg utilizată.



Figura: Thomas Simpson (1710-1761)

# Cuadraturi adaptive I

- La metodele de integrare numerică erorile depind de lungimea intervalului utilizat și de valoarea derivatelor de un anumit ordin ale funcției care urmează a fi integrată.
- Aceasta implică faptul că metodele nu vor lucra bine pentru funcții cu derivele de un anumit ordin mari – în special funcții care au fluctuații mari pe unele subintervale sau pe tot intervalul.
- Este rezonabil să utilizăm subintervale mici acolo unde derivele sunt mari și subintervale mari acolo unde derivele sunt mici. O metodă care face aceasta într-o manieră sistematică se numește **cuadratură adaptivă**.
- Abordarea generală într-o cuadratură adaptivă este de a utiliza două metode diferite pe fiecare subinterval, de a compara rezultatul și de a subdiviza intervalul dacă diferențele sunt mari.
- Există situația nefericită în care se utilizează două metode proaste, rezultatele sunt proaste, dar diferența dintre ele este mică.

## Cuadraturi adaptive II

- Un mod de a evita o astfel de situație este de a ne asigura că o metodă supraestimează rezultatul, iar alta îl subestimează. Vom da un exemplu de structură generală de cuadratură adaptiv-recursivă
- Să presupunem că

*metint( a, b : real; f : functie, n : integer ) : real*

este o funcție care aproximează  $\int_a^b f(x)dx$  folosind o cuadratură repetată cu  $n$  subintervale. Pentru  $n$  se alege o valoare mică (4 sau 5). Vezi algoritmul 36.

# Algoritmul 1. Cuadratură adaptivă

**Intrare:**  $f$  - funcția de integrat,  $a, b$  - limitele de integrare,  $\varepsilon$  - toleranță,  
 $metint$  - o cuadratură repetată

**Ieșire:** valoarea integralei

```
function adapt( $f, a, b, \varepsilon, metint$ )
if  $|metint(a, b, f, 2m) - metint(a, b, f, m)| < \varepsilon$  then
    adapt := metint( $a, b, f, 2m$ );
else
    adapt :=
        adapt( $f, a, (a + b)/2, \varepsilon, metint$ ) + adapt( $f, (a + b)/2, b, \varepsilon, metint$ );
end if
```

# Cuadraturi adaptive

- Structura algoritmului: DIVIDE AND CONQUER.
- Spre deosebire de alte metode, la care se decide cât de mult se muncește pentru a asigura precizia dorită, la o cuadratură adaptivă se calculează doar atât cât este necesar.
- Aceasta înseamnă că eroarea absolută  $\varepsilon$  trebuie aleasă astfel încât să nu se intre într-un ciclu infinit pentru a atinge o precizie imposibil de atins. Numărul de pași depinde de natura funcției de integrat.
- Posibilități de îmbunătățire:  $metint(a, b, f, 2m)$  este apelat de două ori, precizia poate fi scalată cu raportul dintre dimensiunea intervalului curent și dimensiunea întregului interval. Pentru detalii suplimentare recomandăm [8].

# Cuadraturi iterate. Metoda lui Romberg I

- Un dezavantaj al cuadraturilor adaptive este acela că calculează repetat valorile funcției în noduri, iar atunci când este rulat un astfel de program apare un consum suplimentar de timp de calcul datorită recursivității sau gestiunii stivei într-o implementare iterativă.
- Cuadraturile iterate înlătură aceste inconveniente. Ele aplică la primul pas o cuadratură repetată și apoi subdivid intervalele în părți egale folosind la fiecare pas aproximantele calculate anterior.
- Vom exemplifica această tehnică printr-o metodă care pornește de la formula repetată a trapezului și îmbunătățește convergența utilizând extrapolarea Richardson.
- Primul pas al procesului presupune aplicarea formulei repetate a trapezului cu  $n_1 = 1, n_2 = 2, \dots, n_p = 2^{p-1}$ , unde  $p \in \mathbb{N}^*$ . Valoarea pasului  $h_k$  corespunzătoare lui  $n_k$  va fi

$$h_k = \frac{b-a}{n_k} = \frac{b-a}{2^{k-1}}.$$

## Cuadraturi iterate. Metoda lui Romberg II

- Cu aceste notații regula trapezului devine

$$\int_a^b f(x) dx = \frac{h_k}{2} \left[ f(a) + f(b) + 2 \sum_{i=1}^{2^{n-1}-1} f(a + ih_k) \right] - \frac{b-a}{12} h_k^2 f''(\mu_k) \quad (23)$$

$\mu_k \in (a, b)$ .

- Notăm cu  $R_{k,1}$  rezultatul aproximării conform (23).

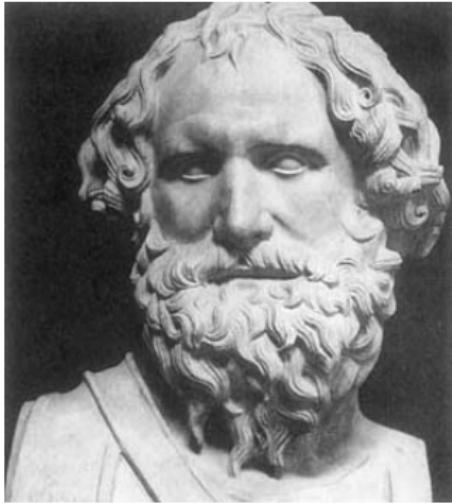
$$R_{1,1} = \frac{h_1}{2} [f(a) + f(b)] = \frac{b-a}{2} [f(a) + f(b)] \quad (24)$$

## Cuadraturi iterate. Metoda lui Romberg III

$$\begin{aligned} R_{2,1} &= \frac{h_2}{2}[f(a) + f(b) + 2f(a + h_2)] = \\ &= \frac{b-a}{4} \left[ f(a) + f(b) + 2f\left(a + \frac{b-a}{2}\right) \right] \\ &= \frac{1}{2} \left[ R_{1,1} + h_1 f\left(a + \frac{1}{2}h_1\right) \right]. \end{aligned}$$

și în general

$$R_{k,1} = \frac{1}{2} \left[ R_{k-1,1} + h_{k-1} \sum_{i=1}^{2^{k-2}} f\left(a + \left(i - \frac{1}{2}\right) h_{k-1}\right) \right], \quad k = \overline{2, n} \quad (25)$$



(a) Arhimede



(b) Lewis Fry Richardson (1881-1953)

Figura:

# Metoda lui Romberg I

- Urmează îmbunătățirea prin extrapolare Richardson

$$I = \int_a^b f(x) dx = R_{k-1,1} - \frac{(b-a)}{12} h_k^2 f''(a) + O(h_k^4).$$

- Vom elimina termenul în  $h_k^2$  combinând două ecuații

$$I = R_{k-1,1} - \frac{(b-a)}{12} h_k^2 f''(a) + O(h_k^4),$$

$$I = R_{k,1} - \frac{b-a}{48} h_k^2 f''(a) + O(h_k^4).$$

- Obținem

$$I = \frac{4R_{k,1} - R_{k-1,1}}{3} + O(h_k^4).$$

- Definim

$$R_{k,2} = \frac{4R_{k,1} - R_{k-1,1}}{3}. \quad (26)$$

## Metoda lui Romberg II

- Se aplică extrapolarea Richardson și acestor valori. În general dacă  $f \in C^{2n+2}[a, b]$ , atunci pentru  $k = \overline{1, n}$  putem scrie

$$\int_a^b f(x) dx = \frac{h_k}{2} \left[ f(a) + f(b) + 2 \sum_{i=1}^{2^{k-1}-1} f(a + ih_k) \right] + \sum_{i=1}^k K_i h_k^{2i} + O(h_k^{2k+2}), \quad (27)$$

unde  $K_i$  nu depinde de  $h_k$ .

- Formula (27) se justifică în modul următor. Fie  $a_0 = \int_a^b f(x) dx$  și

$$A(h) = \frac{h}{2} \left[ f(a) + 2 \sum_{k=1}^{n-1} f(a + kh) + f(b) \right], \quad h = \frac{b-a}{k}.$$

# Metoda lui Romberg III

- Dacă  $f \in C^{2k+1}[a, b]$ ,  $k \in \mathbb{N}^*$

$$A(h) = a_0 + a_1 h^2 + a_2 h^4 + \cdots + a_k h^{2k} + O(h^{2k+1}), \quad h \rightarrow 0 \quad (28)$$

unde

$$a_k = \frac{B_{2k}}{(2k)!} [f^{(2k-1)}(b) - f^{(2k-1)}(a)], \quad k = 1, 2, \dots, K.$$

- Cantitățile  $B_k$  sunt numerele lui Bernoulli, adică coeficienții dezvoltării

$$\frac{z}{e^z - 1} = \sum_{k=0}^{\infty} \frac{B_k}{k!} z^k, \quad |z| < 2\pi.$$

- Formula (28) se numește **formula Euler-MacLaurin**

## Metoda lui Romberg IV

- Eliminând succesiv puterile lui  $h$  din (27) se obține

$$R_{k,j} = \frac{4^{j-1} R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}, \quad k = \overline{2, n}, \quad j = \overline{2, i}. \quad (29)$$

- Calculele se pot aranja tabelar (vezi (25) și (29)), astfel:

$$\begin{matrix} R_{1,1} \\ R_{2,1} & R_{2,2} \\ R_{3,1} & R_{3,2} & R_{3,3} \\ \vdots & \vdots & \vdots & \ddots \\ R_{n,1} & R_{n,2} & R_{n,3} & \dots & R_{n,n} \end{matrix}$$

- Deoarece  $(R_{n,1})$  este convergent și  $(R_{n,n})$  este convergent, mai rapid decât  $(R_{n,1})$ . Drept criteriu de oprire se poate folosi  $|R_{n-1,n-1} - R_{n,n}| \leq \varepsilon$ .



(a) Leonhard Euler (1707-1783)



(b) Colin Maclaurin (1698-1768)

Figura:

# Exemplu I

## Exemplul 12

Aproximați  $\int_0^{\pi} \sin x dx$  prin metoda lui Romberg,  $\varepsilon = 10^{-1}$ .

- Valoarea exactă a integralei este

$$I = \int_0^{\pi} \sin x dx = 2$$

- Urmează doi pași ai formulei trapezelor

$$R_{1,1} = \frac{\pi}{2}(0 + 0) = 0$$

$$R_{2,1} = \frac{1}{2} \left( R_{1,1} + \pi \sin \frac{\pi}{2} \right) = 1.571$$

- Extrapolare Richardson

$$R_{2,2} = 1.571 + (1.571 - 0) / 3 = 2.094$$

## Exemplu II

- Verificare criteriu de oprire

$$(R_{2,2} - R_{1,1}) > 0.1$$

- Din nou trapeze și extrapolare

$$R_{3,1} = \frac{1}{2} \left[ R_{2,1} + \frac{\pi}{2} \left( \sin \frac{\pi}{4} + \sin \frac{3\pi}{4} \right) \right] = 1.895$$

$$R_{3,2} = 1.895 + \frac{1.895 - 1.571}{3} = 2.004$$

$$R_{3,3} = 2.004 + (2.004 - 2.094)/15 = 1.999$$

$$|R_{3,3} - R_{2,2}| < 0.1$$

Pentru trapez cu același număr de argumente  $I \approx 1.895$ . Pentru Simpson cu 4 noduri  $I \approx 2.005$ .

## Cuadraturi adaptive 2 I

- Coloana a doua din metoda lui Romberg corespunde aproximării prin metoda lui Simpson. Notăm

$$S_{k,1} = R_{k,2}.$$

Coloana a treia este deci o combinație a două aproximante de tip Simpson:

$$S_{k,2} = S_{k,1} + \frac{S_{k,1} - S_{k-1,1}}{15} = R_{k,2} + \frac{R_{k,2} - R_{k-1,2}}{15}.$$

Relația

$$S_{k,2} = S_{k,1} + \frac{S_{k,1} - S_{k-1,1}}{15}, \quad (30)$$

va fi folosită la elaborarea unui algoritm de cuadratură adaptivă.

## Cuadraturi adaptive 2 II

- Fie  $c = (a + b)/2$ . Formula elementară a lui Simpson este

$$S = \frac{h}{6} (f(a) + 4f(c) + f(b)).$$

Pentru două subintervale se obține

$$S_2 = \frac{h}{12} (f(a) + 4f(d) + 2f(c) + 4f(e) + f(b)),$$

unde  $d = (a + c)/2$  și  $e = (c + b)/2$ . Cantitatea  $Q$  se obține aplicând (30) celor două aproximante:

$$Q = S_2 + (S_2 - S)/15.$$

Putem să dăm acum un algoritm recursiv pentru aproximarea integralei. Funcția *adquad* evaluează integrandul aplicând regula lui Simpson. Ea apelează recursiv *quadstep* și aplică extrapolarea. Descrierea se dă în algoritmul 51.

# Algoritmul adaptiv

bazat pe metoda lui Simpson și extrapolare

- Datorat lui Gander și Gautschi [8]
- **Intrare:** funcția  $f$ , intervalul  $[a, b]$ , eroarea  $\varepsilon$
- **Ieșire:** Valoarea aproximativă a integralei

```
function adquad(f, a, b, ε) : real
    c := (a + b)/2;
    fa = f(a); fb := f(b); fc := f(c);
    Q := quadstep(f, a, b, ε, fa, fc, fb);
    return Q;
```

# Pasul cuadraturii

```
function quadstep(f, a, b, ε, fa, fc, fb) : real
    h := b - a; c := (a + b)/2;
    fd := f((a + c)/2); fe := f((c + b)/2);
    Q1 := h/6 * (fa + 4 * fc + fb);
    Q2 := h/12 * (fa + 4 * fb + 2 * fc + 4 * fe + fb);
    if |Q1 - Q2| < ε then
        Q := Q2 + (Q2 - Q1)/15;
    else
        Qa := quadstep(f, a, c, ε, fa, fd, fc);
        Qb := quadstep(f, c, b, ε, fc, fe, fb);
        Q := Qa + Qb;
    end if
    return Q;
```

# Formule Newton-Cotes cu ponderi și formule de tip Gauss

- O formulă de cuadratură cu ponderi este o formulă de tipul

$$\int_a^b f(t)w(t)dt = \sum_{k=1}^n w_k f(t_k) + R_n(f) \quad (31)$$

unde  $w$  este nenegativă, integrabilă pe  $(a, b)$ .

- Intervalul  $(a, b)$  poate fi mărginit sau nemărginit.
- Dacă este nemărginit trebuie să ne asigurăm că integrala din (31) este bine definită, cel puțin în cazul când  $f$  este polinom. Realizăm aceasta cerând ca toate momentele funcției pondere

$$\mu_s = \int_a^b t^s w(t)dt, \quad s = 0, 1, 2, \dots \quad (32)$$

să existe și să fie finite.

# Formule de cuadratură de tip interpolator I

- Spunem că (31) este de **tip interpolator**, dacă are gradul de exactitate  $d = n - 1$ .
- Formulele de tip interpolator sunt chiar formulele obținute prin interpolare, adică pentru care

$$\sum_{k=1}^n w_k f(t_k) = \int_a^b (L_{n-1}f)(t; t_1, \dots, t_n) w(t) dt \quad (33)$$

sau echivalent

$$w_k = \int_a^b \ell_k(t) w(t) dt, \quad k = 1, 2, \dots, n, \quad (34)$$

unde

$$\ell_k(t) = \prod_{\substack{j=1 \\ j \neq k}}^n \frac{t - t_j}{t_k - t_j} \quad (35)$$

## Formule de cuadratură de tip interpolator II

sunt polinoamele fundamentale Lagrange asociate nodurilor  $t_1, t_2, \dots, t_n$ .

- Faptul că (31) are gradul de exactitate  $d = n - 1$  este evident, deoarece pentru orice  $f \in \mathbb{P}_{n-1}$ ,  $L_{n-1}(f; \cdot) \equiv f(\cdot)$  în (33).
- Reciproc, dacă (31) are gradul de exactitate  $d = n - 1$ , atunci luând  $f(t) = \ell_r(t)$  în (32) obținem

$$\int_a^b \ell_r(t) w(t) dt = \sum_{k=1}^n w_k \ell_r(t_k) = w_r, \quad r = 1, 2, \dots, n,$$

adică (34).

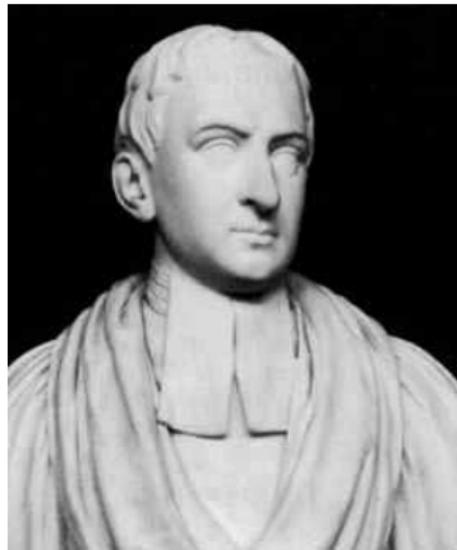
- Observăm că dacă se dau  $n$  noduri distincte  $t_1, \dots, t_n$  este posibil întotdeauna să construim o formulă de tip (31) care este exactă pentru orice polinom de grad  $\leq n - 1$ .

## Formule de cuadratură de tip interpolator III

- În cazul  $w(t) \equiv 1$  pe  $[-1, 1]$  și  $t_k$  sunt echidistante pe  $[-1, 1]$  problema a fost intuită de Newton în 1687 și rezolvată în detaliu de Cotes în jurul anului 1712.
- Prin extensie vom numi formula (31) cu  $t_k$  prescrise și  $w_k$  date de (34) **formulă de tip Newton-Cotes**.



(a) Sir Isaac Newton (1643 - 1727)



(b) Roger Cotes (1682-1716)

Figura:

## Teorema de bază

- Putem obține gradul de exactitate  $d > n - 1$  (ponderile  $w_k$  fiind date în mod necesar de (34)).
- Considerăm polinomul nodurilor

$$u_n(t) = \prod_{k=1}^n (t - t_k). \quad (36)$$

### Teorema 13

Dându-se un întreg  $k$ ,  $0 \leq k \leq n$ , formula de cuadratură (31) are gradul de exactitate  $d = n - 1 + k$  dacă și numai dacă sunt satisfăcute următoarele condiții:

- formula (31) este de tip interpolator;
- polinomul nodurilor  $u_n$  din (36) satisface

$$\int_a^b u_n(t)p(t)w(t) dt = 0, \quad \forall p \in \mathbb{P}_{k-1}.$$

# Teorema de bază I

- Condiția din (b) impune  $k$  condiții asupra nodurilor  $t_1, t_2, \dots, t_n$  din (31).
  - Dacă  $k = 0$ , nu avem nici o restricție, deoarece aşa cum ştim putem atinge gradul de exactitate  $d = n - 1$ . Într-adevăr  $u_n$  trebuie să fie ortogonal pe  $\mathbb{P}_{k-1}$  relativ la funcția pondere  $w$ . Deoarece  $w(t) \geq 0$ , avem în mod necesar  $k \leq n$ .
  - Altfel,  $u_n$  trebuie să fie ortogonal pe  $\mathbb{P}_n$ , în particular pe el însuși, ceea ce este imposibil. Astfel,  $k = n$  este optimal, obținându-se o formulă de cuadratură cu gradul maxim de exactitate  $d_{\max} = 2n - 1$ .
- Condiția (b) impune ortogonalitatea lui  $u_n$  pe toate polinoamele de grad mai mic, adică  $u_n(\cdot) = \pi_n(\cdot, w)$  este polinomul ortogonal în raport cu ponderea  $w$ . Această formulă optimală se numește **formulă de cuadratură de tip Gauss** asociată cu funcția pondere  $w$ .

## Teorema de bază II

- Deci nodurile ei sunt zerourile lui  $\pi_n(\cdot, w)$ , iar ponderile (coeficienții)  $w_k$  sunt dați de (34) adică

$$\pi_n(t_k; w) = 0$$

$$w_k = \int_a^b \frac{\pi_n(t, w)}{(t - t_k)\pi'_n(t_k, w)} w(t) dt, \quad k = 1, 2, \dots, n \quad (37)$$

- Formula a fost dezvoltată de Gauss în 1814 în cazul special  $w(t) \equiv 1$  pe  $[-1, 1]$  și extinsă la funcții pondere mai generale de către Christoffel în 1877. De aceea se mai numește **formulă de cuadratură Gauss-Christoffel**.



(a) Johann Carl Friedrich Gauss  
(1777-1855)



(b) Elvin Bruno Christoffel (1829-1900)

Figura:

## Demonstrația teoremei 13 - necesitatea.

Deoarece gradul de exactitate este  $d = n - 1 + k \geq n - 1$ , condiția (a) este trivială. Condiția (b) rezultă de asemenea imediat, deoarece pentru orice  $p \in \mathbb{P}_{k-1}$ ,  $u_n p \in \mathbb{P}_{n-1+k}$ . Deci

$$\int_a^b u_n(t)p(t)w(t) = \sum_{k=1}^n w_k u_n(t_k)p(t_k),$$

care se anulează, căci  $u_n(t_k) = 0$  pentru  $k = 1, 2, \dots, n$ .

...

## Demonstrația teoremei 13 - suficiență.

Trebuie să arătăm că pentru orice  $p \in \mathbb{P}_{n-1+k}$  avem  $R_n(p) = 0$  în (31). Dându-se orice astfel de  $p$ , îl împărțim cu  $u_n$ , astfel încât

$$p = qu_n + r, \quad q \in \mathbb{P}_{k-1}, \quad r \in \mathbb{P}_{n-1}$$

unde  $q$  este câtul și  $r$  restul. Rezultă că

$$\int_a^b p(t)w(t)dt = \int_a^b q(t)u_n(t)w(t)dt + \int_a^b r(t)w(t)dt.$$

Prima integrală din dreapta este 0, conform lui (b), deoarece  $q \in \mathbb{P}_{k-1}$ , în timp ce a doua, conform lui (a), deoarece  $r \in \mathbb{P}_{n-1}$  este egală cu

$$\sum_{k=1}^n w_k r(t_k) = \sum_{k=1}^n w_k [p(t_k) - q(t_k)u_n(t_k)] = \sum_{k=1}^n w_k p(t_k)$$

ceea ce încheie demonstrația. □

# Formule de tip Gauss-Radau și Gauss-Lobatto

Cazuri importante când  $k < n$ , de interes practic:

- *Formula de cuadratură Gauss-Radau* în care o extremitate de interval, de exemplu  $a$ , este finită și servește ca nod, să zicem  $t_1 = a$ . Gradul maxim de exactitate care se poate obține este  $d = 2n - 2$  și corespunde lui  $k = n - 1$  în teorema (13). Partea (b) a teoremei ne spune că nodurile rămase  $t_2, \dots, t_n$  trebuie să fie rădăcinile polinomului  $\pi_{n-1}(\cdot, w_a)$ , unde  $w_a(t) = (t - a)w(t)$ .
- *Formula Gauss-Lobatto*: ambele capete sunt finite și servesc ca noduri, să zicem  $t_1 = a$ ,  $t_n = b$ , iar nodurile rămase  $t_2, \dots, t_{n-1}$  sunt zerourile lui  $\pi_{n-2}(\cdot; w_{a,b})$ ,  $w_{a,b}(t) = (t - a)(b - t)w(t)$ , obținându-se astfel gradul de exactitate  $d = 2n - 3$ .

## Proprietăți ale cuadraturilor gaussiene

Regula de cuadratură a lui Gauss, dată de (31) și (37), pe lângă faptul că este optimală (adică are grad maxim de exactitate) are și unele proprietăți interesante.

- ① Toate nodurile sunt reale distințe și situate în intervalul deschis  $(a, b)$ . Aceasta este o proprietate cunoscută satisfăcută de zerourile polinoamelor ortogonale.
- ② Toți coeficienții (ponderile)  $w_k$  sunt pozitivi. Demonstrația se bazează pe o observație ingenioasă a lui Stieltjes

$$0 < \int_a^b \ell_j^2(t) w(t) dt = \sum_{k=1}^n w_k \ell_j^2(t_k) = w_j, \quad j = 1, 2, \dots, n,$$

prima egalitate rezultând din faptul că gradul de exactitate este  $d = 2n - 1$ .

# Proprietăți ale cuadraturilor gaussiene - convergența I

- Dacă  $[a, b]$  este mărginit, atunci formula lui Gauss converge pentru orice funcție continuă. Adică  $R_n(f) \rightarrow 0$ , când  $n \rightarrow \infty$ , pentru orice  $f \in C[a, b]$ .
- Conform T. lui Weierstrass , dacă  $\widehat{p}_{2n-1}(f; \cdot)$  este p.c.m.b.a a lui  $f$  pe  $[a, b]$  în sensul  $\|\cdot\|_\infty$ , atunci

$$\lim_{n \rightarrow \infty} \|f(\cdot) - \widehat{p}_{2n-1}(f; \cdot)\|_\infty = 0.$$

## Proprietăți ale cuadraturilor gaussiene - convergența II

- Deoarece  $R_n(\hat{p}_{2n-1}) = 0$  (căci  $d = 2n - 1$ ), avem succesiv

$$\begin{aligned}|R_n(f)| &= |R_n(f - \hat{p}_{2n-1})| \\&= \left| \int_a^b [f(t) - \hat{p}_{2n-1}(f; t)]w(t)dt - \sum_{k=1}^n w_k[f(t_k) - \hat{p}_{2n-1}(f; t_k)] \right| \\&\leq \int_a^b |f(t) - \hat{p}_{2n-1}(f; t)|w(t)dt + \sum_{k=1}^n w_k|f(t_k) - \hat{p}_{2n-1}(f; t_k)| \\&\leq \|f(\cdot) - \hat{p}_{2n-1}(f; \cdot)\|_\infty \left[ \int_a^b w(t)dt + \sum_{k=1}^n w_k \right].\end{aligned}$$

- Aici pozitivitatea ponderilor  $w_k$  a intervenit crucial.

- Observând că

$$\sum_{k=1}^n w_k = \int_a^b w(t) dt = \mu_0,$$

concluzionăm că

$$|R_n(f)| \leq 2\mu_0 \|f - \hat{p}_{2n-1}\|_\infty \rightarrow 0, \text{ când } n \rightarrow \infty.$$

## Proprietăți ale cuadraturilor gaussiene - restul I

- Markov a observat că formula de cuadratură a lui Gauss poate fi obținută integrând termen cu termen formula de interpolare a lui Hermite cu noduri duble.

$$f(x) = (H_{2n-1}f)(x) + u_n^2(x)f[x, x_1, x_1, \dots, x_n, x_n],$$

$$\begin{aligned}\int_a^b w(x)f(x)dx &= \int_a^b w(x)(H_{2n-1}f)(x)dx + \\ &+ \int_a^b w(x)u_n^2(x)f[x, x_1, x_1, \dots, x_n, x_n]dx.\end{aligned}$$

## Proprietăți ale cuadraturilor gaussiene - restul II

- Dar gradul de exactitate  $2n - 1$  implică

$$\int_a^b w(x)(H_{2n-1}f)(x)dx = \sum_{i=1}^n w_i(H_{2n-1}f)(x_i) = \sum_{i=1}^n w_i f(x_i),$$

$$\int_a^b w(x)f(x)dx = \sum_{i=1}^n w_i f(x_i) + \int_a^b w(x)u^2(x)f[x, x_1, x_1, \dots, x_n, x_n]dx,$$

deci

$$R_n(f) = \int_a^b w(x)u_n^2(x)f[x, x_1, x_1, \dots, x_n, x_n]dx.$$

## Proprietăți ale cuadraturilor gaussiene - restul III

- Cum  $w(x)u^2(x) \geq 0$ , aplicând teorema de medie pentru integrale și teorema de medie pentru diferențe divizate avem

$$\begin{aligned} R_n(f) &= f[\eta, x_1, x_1, \dots, x_n, x_n] \int_a^b w(x)u^2(x)dx \\ &= \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b w(x)[\pi_n(x, w)]^2 dx, \quad \xi \in [a, b]. \end{aligned}$$

- Deci,  $\exists \xi \in [a, b]$  a.î.

$$R_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b w(x)[\pi_n(x, w)]^2 dx.$$

# Calculul nodurilor și al coeficienților I

- Fie  $\alpha_k = \alpha_k(w)$  și  $\beta_k = \beta_k(w)$  coeficienții din formula de recurență pentru polinoamele ortogonale

$$\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0, 1, 2, \dots \quad (38)$$

$$\pi_0(t) = 1, \quad \pi_{-1}(t) = 0,$$

cu  $\beta_0$  definit (ca de obicei) prin

$$\beta_0 = \int_a^b w(t) dt \quad (= \mu_0).$$

## Calculul nodurilor și al coeficienților II

- **Matricea Jacobi** de ordinul  $n$  pentru funcția pondere  $w$  este o matrice simetrică tridiagonală definită prin

$$J_n(w) = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & 0 \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \ddots & & \\ & & \ddots & \ddots & \sqrt{\beta_{n-1}} \\ 0 & & & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix}.$$

- Nodurile  $t_k$  sunt valori proprii ale lui  $J_n$

$$J_n v_k = t_k v_k, \quad v_k^T v_k = 1, \quad k = 1, 2, \dots, n, \quad (39)$$

- Ponderile  $w_k$  sunt exprimabile cu ajutorul componentelor  $v_k$ , ale vectorilor proprii normalizați corespunzători prin

$$w_k = \beta_0 v_{k,1}^2, \quad k = 1, 2, \dots, n \quad (40)$$

# Calculul nodurilor și al coeficienților III

- Astfel, pentru a obține o formulă de cuadratură gaussiană trebuie rezolvată o problemă de vectori și valori proprii pentru o matrice tridiagonală simetrică. Pentru această problemă există metode foarte eficiente.

polinoamele	notația	ponderea	intervalul	$\alpha_k$	$\beta_k$
Legendre	$P_n(\ell_n)$	1	[-1,1]	0	$2 \ (k=0)$ $(4-k^{-2})^{-1} \ (k>0)$
Cebîșev #1	$T_n$	$(1-t^2)^{-\frac{1}{2}}$	[-1,1]	0	$\pi \ (k=0)$ $\frac{1}{2}\pi \ (k=1)$ $\frac{1}{4} \ (k>0)$
Cebîșev #2	$U_n(Q_n)$	$(1-t^2)^{\frac{1}{2}}$	[-1,1]	0	$\frac{1}{2}\pi \ (k=0)$ $\frac{1}{4} \ (k>0)$
Jacobi	$P_n^{(\alpha,\beta)}$	$(1-t)^\alpha(1+t)^\beta$ $\alpha>-1, \ \beta>-1$	[-1,1]		
Laguerre	$L_n^{(\alpha)}$	$t^\alpha e^{-t} \ \alpha>-1$	[0,∞)	$2k+\alpha+1$	$\Gamma(1+\alpha) \ (k=0)$ $k(k+\alpha) \ (k>0)$
Hermite	$H_n$	$e^{-t^2}$	$\mathbb{R}$	0	$\sqrt{\pi} \ (k=0)$ $\frac{1}{2}k \ (k>0)$

**Tabelă:** Funcțiile pondere clasice, polinoamele lor ortogonale corespunzătoare și coeficienții din formula de recurență  $\alpha_k, \beta_k$

# Bibliografie I

-  Gheorghe Coman, *Analiză numerică*, Editura Libris, Cluj-Napoca, 1995.
-  W. Gautschi, *Numerical Analysis*, Second edition, Birkhäuser, Basel, 2012.
-  W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, New York, Port Chester, Melbourne, Sidney, 1996, disponibila prin www, <http://www.nr.com/>.
-  D. D. Stancu, *Analiză numerică – Curs și culegere de probleme*, Lito UBB, Cluj-Napoca, 1977.
-  J. Stoer, R. Burlisch, *Introduction to Numerical Analysis*, 2nd ed., Springer Verlag, 1992.

## Bibliografie II

-  R. Trîmbițaș, *Numerical Analysis in MATLAB*, Cluj University Press, 2010
-  D. D. Stancu, G. Coman, P. Blaga, *Analiză numerică și teoria aproximării*, vol. II, Presa Universitară Clujeană, Cluj-Napoca, 2002, D. D. Stancu, Gh. Coman, (coord.).
-  W. Gander, W. Gautschi, *Adaptive quadrature - revisited*, BIT 40 (2000), 84–101.

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Rezolvarea numerică a ecuațiilor neliniare

## De aici începe adevărata Analiză numerică

Radu Trîmbițaș

UBB

10 mai 2021

# Ecuații neliniare I

- ▶ Problema discutată în acest capitol se poate scrie generic sub forma

$$f(x) = 0, \quad (1)$$

dar admite diverse interpretări, depinzând de semnificația lui  $x$  și  $f$ .

- ▶ Cel mai simplu caz este cel al unei singure ecuații cu o singură necunoscută, caz în care  $f$  este o funcție dată de o variabilă reală sau complexă și încercăm să găsim valorile acestei variabile pentru care  $f$  se anulează. Astfel de valori se numesc *rădăcini* ale ecuației (1) sau *zerouri* ale funcției  $f$ .
- ▶ Dacă  $x$  din (1) este un vector, să zicem  $x = [x_1, x_2, \dots, x_d]^T \in \mathbb{R}^d$  și  $f$  este de asemenea un vector ale cărui componente sunt funcții de cele  $d$  variabile  $x_1, x_2, \dots, x_d$ , atunci (1) reprezintă un *sistem de ecuații*.

# Ecuații neliniare II

Rezolvarea  
numerică a  
ecuațiilor neliniare

- ▶ Se spune că sistemul este *neliniar* dacă cel puțin una dintre componente ale lui  $f$  depinde neliniar de cel puțin una din variabilele  $x_1, x_2, \dots, x_d$ . Dacă toate componente ale lui  $f$  sunt funcții liniare de  $x_1, \dots, x_d$  avem de-a face cu un sistem de ecuații algebrice liniare.
- ▶ Mai general (1) ar putea reprezenta o ecuație funcțională, dacă  $x$  este un element al unui spațiu de funcții și  $f$  este un operator (liniar sau neliniar) ce acționează pe acest spațiu. În fiecare din aceste situații zeroul din dreapta lui (1) poate avea diverse interpretări: numărul zero în primul caz, vectorul nul în al doilea și funcția identic nulă în cel de-al treilea.
- ▶ Mare parte din acest capitol este consacrată unei ecuații neliniare scalare. Astfel de ecuații apar frecvent în analiza sistemelor în vibrație, unde rădăcinile corespund frecvențelor critice (rezonanță).

Radu Trîmbițaș

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Ecuații neliniare III

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Iterații, convergență și eficiență

- ▶ Nici chiar cele mai simple ecuații - de exemplu cele algebrice - nu admit soluții care să fie exprimabile prin expresii rationale sau radicali.
- ▶ Din acest motiv este imposibil, în general, să calculăm rădăcinile ecuațiilor neliniare printr-un număr finit de operații aritmetice. Este nevoie de o metodă iterativă, adică de o procedură care generează o secvență infinită de aproximății  $\{x_n\}_{n \in \mathbb{N}}$  astfel încât

$$\lim_{n \rightarrow \infty} x_n = \alpha, \quad (2)$$

unde  $\alpha$  este o rădăcină a ecuației.

- ▶ În cazul unui sistem  $x_k$  și  $\alpha$  sunt vectori de dimensiune adecvată, iar convergența trebuie înțeleasă în sensul convergenței pe componente.
- ▶ În practică, ceea ce se dorește este o convergență rapidă.

# Ordinul de convergență I

Conceptul de bază pentru măsurarea vitezei de convergență este ordinul de convergență.

## Definiția 1

Spunem că  $x_n$  converge către  $\alpha$  (*cel puțin*) liniar dacă

$$|x_n - \alpha| \leq e_n \quad (3)$$

unde  $\{e_n\}$  este un sir pozitiv ce satisface

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = c, \quad 0 < c < 1. \quad (4)$$

Dacă (3) și (4) au loc cu egalitate în (3) atunci c se numește eroare asimptotică.

- ▶ Expresia „cel puțin“ în această definiție se leagă de faptul că avem doar inegalitate în (3), ceea ce dorim în practică.

# Ordinul de convergență II

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- De fapt, strict vorbind, marginea  $e_n$  converge liniar, însemnând că, în final (pentru  $n$  suficient de mare) fiecare din aceste margini ale erorii este aproximativ o fracție constantă din precedenta.

## Definiția 2

Se spune că  $x_n$  converge către  $\alpha$  (cel puțin) cu ordinul  $p \geq 1$  dacă (3) are loc cu

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n^p} = c, \quad c > 0 \quad (5)$$

- Astfel convergența de ordinul 1 coincide cu convergența liniară, în timp ce convergența de ordinul  $p > 1$  este mai rapidă.

Ecuății neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuății algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Ordinul de convergență III

Rezolvarea  
numericală a  
ecuațiilor nelineare

Radu Trîmbițaș

- ▶ De notat că în acest ultim caz nu se pune nici o restricție asupra constantei  $c$ : odată ce  $e_n$  este suficient de mic, exponentul  $p$  va avea grijă de convergență. Și în acest caz, dacă avem egalitate în (3),  $c$  se numește *eroare asimptotică*.
- ▶ Aceleași definiții se aplică și sirurilor vectoriale, cu modulul înlocuit cu orice normă vectorială.
- ▶ Clasificarea convergenței în raport cu ordinul este destul de rudimentară, deoarece sunt tipuri de convergență la care definițiile (1) și (2) nu se aplică. Astfel, un sir  $\{e_n\}$  poate converge către zero mai încet decât liniar, de exemplu dacă  $c = 1$  în (4). Acest tip de convergență se numește *subliniară*. La fel,  $c = 0$  în (4) conduce la convergență *superliniară*, dacă (5) nu are loc pentru nici un  $p > 1$ .

Ecuății nelineare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuății algebrice

Sisteme nelineare

Metode  
quasi-Newton  
Interpolate liniară

Metode de modificare

Interpolate inversă

Metode hibride

Bibliografie

# Ordinul de convergență IV

- Este instructiv să examinăm comportarea lui  $e_n$ , dacă în loc de relația la limită avem egalitate pentru un anumit  $n$ , să zicem

$$\frac{e_{n+1}}{e_n^p} = c, \quad n = n_0, n_0 + 1, n_0 + 2, \dots \quad (6)$$

- Pentru  $n_0$  suficient de mare, relația (6) este aproape adevărată. Prinț-o simplă inducție se obține că

$$e_{n_0+k} = c^{\frac{p^k - 1}{p-1}} e_{n_0}^{p^k}, \quad k = 0, 1, 2, \dots, \quad (7)$$

care desigur are loc pentru  $p > 1$ , dar și pentru  $p = 1$  când  $p \downarrow 1$ :

$$e_{n_0+k} = c^k e_{n_0}, \quad k = 0, 1, 2, \dots, \quad (p = 1) \quad (8)$$

# Ordinul de convergență V

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuări neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuări algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

- ▶ Presupunând că  $e_{n_0}$  este suficient de mare astfel încât aproximarea  $x_{n_0}$  are un număr de zecimale corecte, scriem  $e_{n_0+k} = 10^{-\delta_k} e_{n_0}$ .
- ▶ Atunci  $\delta_k$ , în conformitate cu (3) reprezintă numărul suplimentar de cifre zecimale corecte din aproximația  $x_{n_0+k}$  (în contrast cu  $x_{n_0}$ ). Logaritmând (7) și (8) obținem

$$\delta_k = \begin{cases} k \log \frac{1}{c}, & \text{dacă } p = 1 \\ p^k \left[ \frac{1-p^{-k}}{p-1} \log \frac{1}{c} + (1 - p^{-k}) \log \frac{1}{e_{n_0}} \right], & \text{dacă } p > 1 \end{cases}$$

- ▶ Deci când  $k \rightarrow \infty$

$$\delta_k \sim c_1 k \quad (p = 1), \quad \delta_k \sim c_p p^k \quad (p > 1), \quad (9)$$

# Ordinul de convergență VI

unde  $c_1 = \log \frac{1}{c} > 0$ , dacă  $p = 1$  și

$$c_p = \frac{1}{p-1} \log \frac{1}{c} + \log \frac{1}{e_{n_0}}$$

(presupunem că  $n_0$  este suficient de mare și deci  $e_{n_0}$  suficient de mic, pentru a avea  $c_p > 0$ ).

- ▶ Aceasta ne arată că numărul de cifre zecimale corecte crește liniar odată cu  $k$  când  $p = 1$ , dar exponențial când  $p > 1$ . În ultimul caz  $\delta_{k+1}/\delta_k \sim p$  înseamnă că (pentru  $k$  mare) numărul de cifre zecimale corecte crește, pe iterație, cu un factor  $p$ .

Ecuății neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuății algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Ordinul de convergență VII

Rezolvarea  
numerică a  
ecuațiilor nelineare

Radu Trîmbițaș

- Dacă fiecare iterație necesită  $m$  unități de lucru (o „unitate de lucru” este efortul necesar pentru a calcula o valoare a funcției sau a unei anumite derivate a sa), atunci *indicele de eficiență* al iterației poate fi definit prin

$$\lim_{k \rightarrow \infty} [\delta_{k+1}/\delta_k]^{1/m} = p^{1/m}.$$

- Aceasta ne dă o bază comună de comparare între diversele metode iterative. Metodele liniare au indicele de eficiență 1.

Ecuații nelineare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme nelineare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Criteriul de oprire I

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

- ▶ Calculele practice necesită o regulă de oprire care să termine iterația atunci când s-a obținut (sau se crede că s-a obținut) precizia dorită.
- ▶ Ideal, ne oprim atunci când  $\|x_n - \alpha\| < tol$ ,  $tol$  dat.
- ▶ Deoarece  $\alpha$  nu este cunoscut se obișnuiește să se înlocuiască  $x_n - \alpha$  cu  $x_n - x_{n-1}$  și se impune cerința ca

$$\|x_n - x_{n-1}\| \leq tol \quad (10)$$

unde

$$tol = \|x_n\| \varepsilon_r + \varepsilon_a \quad (11)$$

cu  $\varepsilon_r, \varepsilon_a$  valori date ale erorii.

- ▶ Ca o măsură de siguranță, am putea cere ca (10) să aibă loc pentru mai multe valori consecutive ale lui  $n$ , nu doar pentru una singură.

# Criteriul de oprire II

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- ▶ Alegând  $\varepsilon_r = 0$  sau  $\varepsilon_a = 0$  se obține un test de eroare absolută sau relativă. Este totuși prudent să utilizăm un test mixt, cum ar fi, să zicem  $\varepsilon_r = \varepsilon_a = \varepsilon$ . Atunci, dacă  $\|x_n\|$  este mic sau moderat de mare, se controlează efectiv eroarea absolută, în timp ce pentru  $\|x_n\|$  foarte mare se controlează eroarea relativă.
- ▶ Testele de mai sus se pot combina cu  $\|f(x)\| \leq \varepsilon$ . În algoritmii din acest capitol vom presupune că avem o funcție *crit\_oprire* care implementează testul de oprire.

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda falsei poziții

- ▶ Ca în metoda înjumătățirii, presupunem că avem două numere  $a < b$  astfel încât

$$f \in C[a, b], \quad f(a)f(b) < 0 \quad (12)$$

și generăm un sir descendant de intervale  $[a_n, b_n]$ ,  $n = 1, 2, 3, \dots$  cu  $a_1 = a$ ,  $b_1 = b$  astfel încât  $f(a_n)f(b_n) < 0$ .

- ▶ Spre deosebire de metoda înjumătățirii, pentru a determina următorul interval nu luăm mijlocul lui  $[a_n, b_n]$ , ci soluția  $x = x_n$  a ecuației liniare

$$(L_1 f)(x; a_n, b_n) = 0.$$

- ▶ Aceasta pare să fie o alegere mai flexibilă decât în metoda înjumătățirii deoarece  $x_n$  va fi mai apropiat de capătul în care  $|f|$  este mai mic (Vezi figura 1)

# Metoda falsei poziții - algoritmul

Procedura decurge după cum urmează:

```
for n := 1, 2, ... do
     $x_n := a_n - \frac{a_n - b_n}{f(a_n) - f(b_n)} f(a_n);$ 
    if  $f(a_n)f(x_n) > 0$  then
         $a_{n+1} := x_n; b_{n+1} := b_n;$ 
    else
         $a_{n+1} := a_n; b_{n+1} := x_n;$ 
    end if
end for
```

Iterația se poate termina când  $\min(x_n - a_n, b_n - x_n) \leq tol$ , unde  $tol$  este o valoare dată.

[Ecuații neliniare](#)[Ordin de convergență](#)[Falsa poziție](#)[Metoda secantei](#)[Metoda lui Newton](#)[Metoda aproximățiilor successive](#)[Rădăcini multiple](#)[Ecuații algebrice](#)[Sisteme neliniare](#)[Metode quasi-Newton](#)[Interpolate liniară](#)[Metode de modificare](#)[Interpolate inversă](#)[Metode hibride](#)[Bibliografie](#)

# Metoda falsei poziții - convergența I

- Convergența se analizează mai ușor dacă presupunem că  $f$  este convexă sau concavă pe  $[a, b]$ . Dacă  $f$  este convexă, avem

$$f''(x) > 0, \quad x \in [a, b], \quad f(a) < 0, \quad f(b) > 0. \quad (13)$$

- Şirul

$$x_{n+1} = x_n - \frac{x_n - b}{f(x_n) - f(b)} f(x_n), \quad n \in \mathbb{N}^*, \quad x_1 = a \quad (14)$$

este monoton crescător și mărginit superior de  $\alpha$ , deci convergent către o limită  $x$ , iar  $f(x) = 0$ .

- Viteza de convergență se determină scăzând  $\alpha$  din ambiii membri ai lui (14) și utilizând faptul că  $f(\alpha) = 0$ :

$$x_{n+1} - \alpha = x_n - \alpha - \frac{x_n - b}{f(x_n) - f(b)} [f(x_n) - f(\alpha)].$$

# Metoda falsei poziții - convergența II

- Împărțind cu  $x_n - \alpha$  avem

$$\frac{x_{n+1} - \alpha}{x_n - \alpha} = 1 - \frac{x_n - b}{f(x_n) - f(b)} \frac{f(x_n) - f(\alpha)}{x_n - \alpha}.$$

- Făcând  $n \rightarrow \infty$  și utilizând faptul că  $x_n \rightarrow \alpha$ , obținem

$$\lim_{n \rightarrow \infty} \frac{x_{n+1} - \alpha}{x_n - \alpha} = 1 - (b - \alpha) \frac{f'(\alpha)}{f'(b)}. \quad (15)$$

- Deci metoda converge liniar, cu eroarea asimptotică

$$c = 1 - (b - a) \frac{f'(\alpha)}{f'(b)}.$$

- Datorită ipotezei convexității avem  $c \in (0, 1)$ .
- Analog se face demonstrația în cazul când  $f$  este concavă.

# Metoda falsei poziții - convergența III

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- ▶ Dacă  $f$  nu este convexă sau concavă pe  $[a, b]$ , ci  $f \in C^2[a, b]$  și  $f''(\alpha) \neq 0$ ,  $f''$  are semn constant pe o vecinătate a lui  $\alpha$  și pentru un  $n$  suficient de mare  $x_n$  ajunge în acea vecinătate și se poate proceda ca mai sus.
- ▶ Dezavantaje. (i) Convergența lentă; (ii) Faptul că unul din capete poate rămâne fix. Dacă  $f$  este turtită în vecinătatea rădăcinii și a este apropiat de  $\alpha$  și  $b$  depărtat convergența poate fi foarte lentă.

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolate liniară

Metode de modificare

Interpolate inversă

Metode hibride

Bibliografie

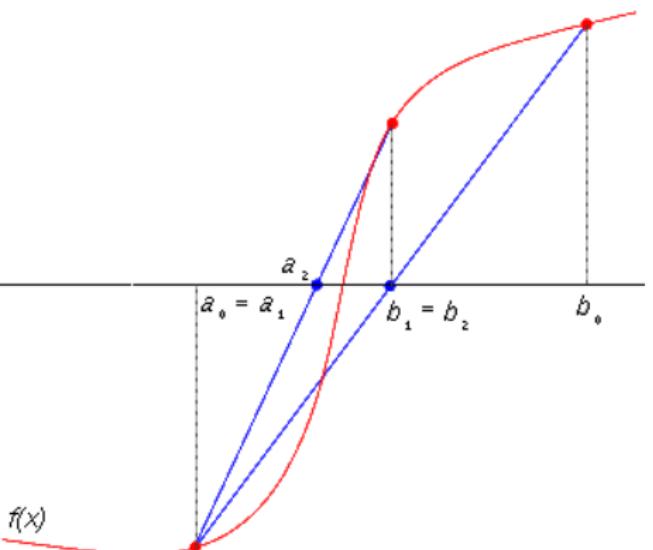


Figura: Metoda falsei poziții

# Metoda secantei I

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- ▶ Este o variantă a metodei falsei poziții, în care nu se mai cere ca  $f$  să aibă valori de semne contrare, nici măcar la capetele intervalului inițial.
- ▶ Se aleg două valori arbitrate de pornire  $x_0, x_1$  și se continuă cu

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n), \quad n \in \mathbb{N}^* \quad (16)$$

- ▶ Aceasta preîntâmpină apariția unei false poziții și sugerează o convergență mai rapidă.
- ▶ Din păcate, nu mai are loc convergența „globală“ pe  $[a, b]$  ci doar convergența „locală“, adică numai dacă  $x_0$  și  $x_1$  sunt suficient de apropiate de rădăcină.
- ▶ Vom avea nevoie de o relație între trei erori consecutive

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda secantei II

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolate liniară

Metode de modificare

Interpolate inversă

Metode hibride

Bibliografie

$$\begin{aligned}x_{n+1} - \alpha &= x_n - \alpha - \frac{f(x_n)}{f[x_{n-1}, x_n]} \\&= (x_n - \alpha) \left(1 - \frac{f(x_n) - f(\alpha)}{(x_n - \alpha)f[x_{n-1}, x_n]}\right) \\&= (x_n - \alpha) \left(1 - \frac{f[x_n, \alpha]}{f[x_{n-1}, x_n]}\right) \\&= (x_n - \alpha) \frac{f[x_{n-1}, x_n] - f[x_n, \alpha]}{f[x_{n-1}, x_n]} \\&= (x_n - \alpha)(x_{n-1} - \alpha) \frac{f[x_n, x_{n-1}, \alpha]}{f[x_{n-1}, x_n]}.\end{aligned}$$

# Metoda secantei III

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

► Deci

$$(x_{n+1} - \alpha) = (x_n - \alpha)(x_{n-1} - \alpha) \frac{f[x_n, x_{n-1}, \alpha]}{f[x_{n-1}, x_n]}, \quad n \in \mathbb{N}^* \quad (17)$$

► Din (17) rezultă imediat că dacă  $\alpha$  este o rădăcină simplă ( $f(\alpha) = 0, f'(\alpha) \neq 0$ ) și  $x_n \rightarrow \alpha$  și dacă  $f \in C^2$  pe o vecinătate a lui  $\alpha$ , convergența este superliniară.

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Ordinul de convergență I

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- ▶ Înlocuim raportul diferențelor divizate din (17) cu o constantă, ceea ce este aproape adevărat când  $n$  este mare. Punând  $e_k = |x_k - \alpha|$ , avem

$$e_{n+1} = e_n e_{n-1} C, \quad C > 0$$

- ▶ Înmulțind ambeii membri cu  $C$  și punând  $E_n = Ce_n$  obținem

$$E_{n+1} = E_n E_{n-1}, \quad E_n \rightarrow 0.$$

- ▶ Logaritmând și punând  $y_n = \frac{1}{E_n}$  obținem

$$y_{n+1} = y_n + y_{n-1}, \quad (18)$$

care este recurența pentru sirul lui Fibonacci.

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Ordinul de convergență II

Rezolvarea  
numerică a  
ecuațiilor neliniare

- Soluția este

$$y_n = c_1 t_1^n + c_2 t_2^n,$$

$c_1, c_2$  constante și

$$t_1 = \frac{1}{2}(1 + \sqrt{5}), \quad t_2 = \frac{1}{2}(1 - \sqrt{5}).$$

- Deoarece  $y_n \rightarrow \infty$ , avem  $c_1 \neq 0$  și  $y_n \sim c_1 t_1^n$ , căci  $|t_2| < 1$ . Revenind la substituție  $\frac{1}{E_n} \sim e^{c_1 t_1^n}$ ,  
 $\frac{1}{e_n} \sim Ce^{c_1 t_1^n}$  și deci

$$\frac{e_{n+1}}{e_n^{t_1}} \sim \frac{C^{t_1} e^{c_1 t_1^n t_1}}{Ce^{c_1 t_1^{n+1}}} = C^{t_1 - 1}, \quad n \rightarrow \infty.$$

- Ordinul de convergență este

$$t_1 = \frac{1 + \sqrt{5}}{2} \approx 1.61803\dots \text{ (secțiunea de aur).}$$

Radu Trîmbițaș

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Convergența metodei secantei I

Rezolvarea  
numerică a  
ecuațiilor nelineare

Radu Trîmbițaș

## Teorema 3

Fie  $\alpha$  un zero simplu al lui  $f$  și fie

$I_\varepsilon = \{x \in \mathbb{R} : |x - \alpha| < \varepsilon\}$  și presupunem că  $f \in C^2[I_\varepsilon]$ .

Definim pentru  $\varepsilon$  suficient de mic

$$M(\varepsilon) = \max_{\substack{s \in I_\varepsilon \\ t \in I_\varepsilon}} \left| \frac{f''(s)}{2f'(t)} \right|. \quad (19)$$

Presupunem că

$$\varepsilon M(\varepsilon) < 1 \quad (20)$$

Atunci metoda secantei converge către rădăcina unică  $\alpha \in I_\varepsilon$  pentru orice valoare de pornire  $x_0 \neq x_1$  cu  $x_0 \in I_\varepsilon$ ,  $x_1 \in I_\varepsilon$ .

Ecuații nelineare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme nelineare

Metode  
quasi-Newton

Interpolate liniară

Metode de modificare

Interpolate inversă

Metode hibride

Bibliografie

# Convergența metodei secantei II

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Demonstrație - pasul I

Se observă că  $\alpha$  este *singurul zero* al lui  $f$  în  $I_\varepsilon$ . Aceasta rezultă din formula lui Taylor pentru  $x = \alpha$ :

$$f(x) = f(\alpha) + (x - \alpha)f'(\alpha) + \frac{(x - \alpha)^2}{2}f''(\xi)$$

unde  $f(\alpha) = 0$  și  $\xi \in (x, \alpha)$  (sau  $(\alpha, x)$ ). Astfel dacă  $x \in I_\varepsilon$ , atunci și  $\xi \in I_\varepsilon$  și avem

$$f(x) = (x - \alpha)f'(\alpha) \left[ 1 + \frac{x - \alpha}{2} \frac{f''(\xi)}{f'(\alpha)} \right]$$

Aici, dacă  $x \neq \alpha$ , toți trei factorii sunt diferenți de 0, căci

$$\left| \frac{x - \alpha}{2} \frac{f''(\xi)}{f'(\alpha)} \right| \leq \varepsilon M(\varepsilon) < 1.$$

Deci  $f$  se poate anula pe  $I_\varepsilon$  numai în  $x = \alpha$ .

## Demonstrație - pasul II

Să arătăm că  $x_n \in I_\varepsilon$  pentru orice  $n$ , în afară de cazul când  $f(x_n) = 0$ , în care  $x_n = \alpha$  și metoda converge într-un număr finit de pași. Vom demonstra aceasta prin inducție: presupunem că  $x_{n-1}, x_n \in I_\varepsilon$  și  $x_n \neq x_{n-1}$ . Acest lucru este adevărat pentru  $n = 1$  din ipoteză.

Deoarece  $f \in C^2[I_\varepsilon]$

$$f[x_{n-1}, x_n] = f'(\xi_1), \quad f[x_{n-1}, x_n, \alpha] = \frac{1}{2}f''(\xi_2), \quad \xi_i \in I_\varepsilon, \quad i = 1, 2,$$

din (17) rezultă

$$|x_{n+1} - \alpha| \leq \varepsilon^2 \left| \frac{f''(\xi_n)}{2f'(\xi_1)} \right| \leq \varepsilon \varepsilon M(\varepsilon) < \varepsilon,$$

adică  $x_{n+1} \in I_\varepsilon$ .

# Demonstrație - pasul III

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

**Convergența.** Mai mult, din relația între trei erori consecutive, (17), rezultă  $x_{n+1} \neq x_n$  în afară de cazul când  $f(x_n) = 0$  (și atunci  $x_n = \alpha$ ). Utilizând (17) avem

$$|x_{n+1} - \alpha| \leq |x_n - \alpha| \varepsilon M(\varepsilon)$$

care aplicată repetat ne dă

$$|x_{n+1} - \alpha| \leq |x_n - \alpha| \varepsilon M(\varepsilon) \leq \cdots \leq [\varepsilon M(\varepsilon)]^{n-1} |x_1 - \alpha|.$$

Cum  $\varepsilon M(\varepsilon) < 1$ , rezultă că metoda este convergentă și  $x_n \rightarrow \alpha$  când  $n \rightarrow \infty$ .

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Algoritmul

Deoarece este nevoie de o singură evaluare a lui  $f$  pe pas, indicele de eficiență este  $p = \frac{1+\sqrt{5}}{2} \approx 1.61803\dots$

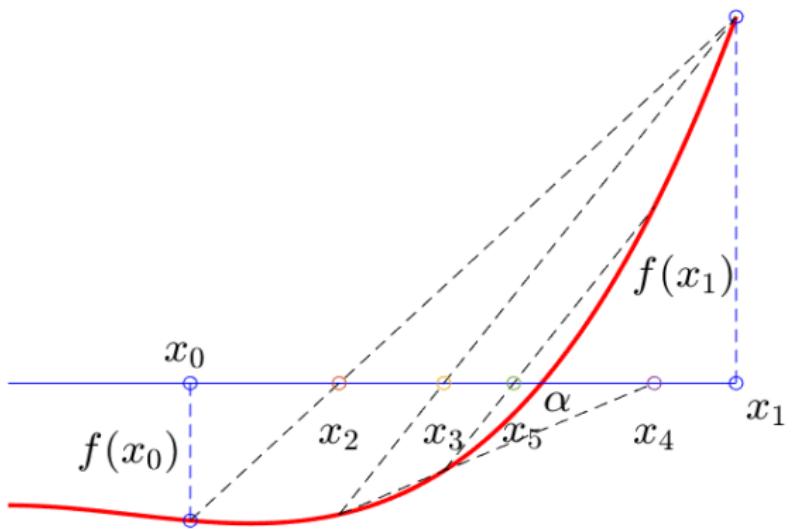


Figura: Ilustrarea metodei secantei

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuări neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuări algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Algoritmul în pseudocod

**Intrare:** Funcția  $f$ , valorile de pornire  $x_0$  și  $x_1$ , numarul maxim de iterații,  $Nmax$ , informații de toleranță  $tol$

**Ieșire:** O aproximatie a rădăcinii sau un mesaj de eroare

```
1: xc :=  $x_1$ ;      xv =  $x_0$ ;  
2: fc :=  $f(x_1)$ ;    fv :=  $f(x_0)$ ;  
3: for  $k := 1$  to  $Nmax$  do  
4:    $xn := xc - fc * (xc - xv) / (fc - fv)$ ;  
5:   if  $crit\_oprire(tol)$  then  
6:     return  $xn$ ; {Succes}  
7:   end if  
8:    $xv := xc$ ;     $fv := fc$ ;     $xc := xn$ ;     $fc = f(xn)$ ;  
9: end for  
10: error("S-a depășit numărul de iterații").
```

Ecuății neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda aproximățiilor successive

Rădăcini multiple

Ecuății algebrice

Sisteme neliniare

Metode quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda lui Newton I

Rezolvarea  
numerică a  
ecuațiilor neliniare

- ▶ Poate fi privită ca un caz la limită al metodei secantei, când  $x_{n-1} \rightarrow x_n$ :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (21)$$

- ▶ Altă interpretare: liniarizarea ecuației  $f(x) = 0$  în  $x = x_n$  (vezi figura 3)

$$f(x) \approx (T_1 f)(x) = f(x_n) + (x - x_n)f'(x_n) = 0.$$

- ▶ Astfel, metoda lui Newton se poate generaliza la ecuații neliniare de toate tipurile (sisteme neliniare, ecuații funcționale, caz în care  $f'$  trebuie înțeleasă ca derivată Fréchet), iar iterația este

$$x_{n+1} = x_n - [f'(x_n)]^{-1}f(x_n). \quad (22)$$

Radu Trîmbițaș

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda lui Newton II

- ▶ Studiul erorii în metoda lui Newton este la fel ca cel al erorii în metoda secantei

$$\begin{aligned}x_{n+1} - \alpha &= x_n - \alpha - \frac{f(x_n)}{f'(x_n)} \\&= (x_n - \alpha) \left[ 1 - \frac{f(x_n) - f(\alpha)}{(x_n - \alpha)f'(x_n)} \right] \\&= (x_n - \alpha) \left( 1 - \frac{f[x_n, \alpha]}{f[x_n, x_n]} \right) \\&= (x_n - \alpha)^2 \frac{f[x_n, x_n, \alpha]}{f[x_n, x_n]}\end{aligned}\tag{23}$$

- ▶ De aceea, dacă  $x_n \rightarrow \alpha$ , atunci

$$\lim_{n \rightarrow \infty} \frac{x_{n+1} - \alpha}{(x_n - \alpha)^2} = \frac{f''(\alpha)}{2f'(\alpha)}$$

și ordinul de convergență al metodei lui Newton este 2 dacă  $f''(\alpha) \neq 0$ .

# Interpretarea geometrică a metodei lui Newton

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

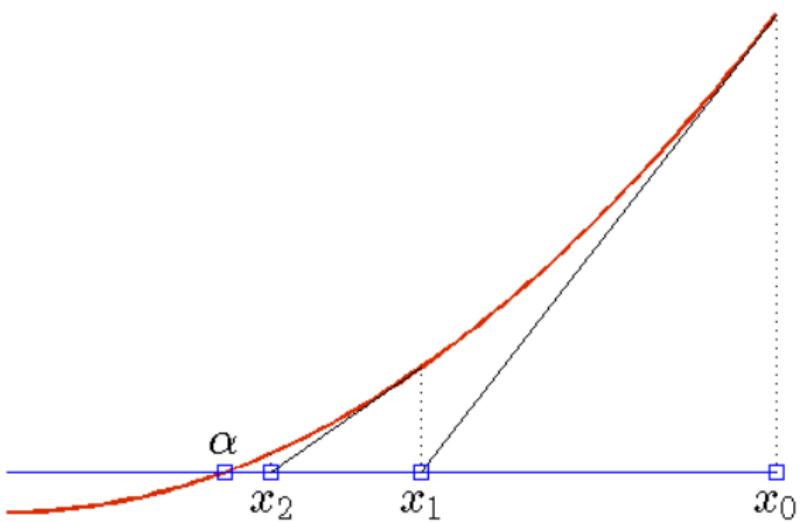


Figura: Metoda lui Newton

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Convergența

Referitor la convergența locală a metodei lui Newton avem

## Teorema 5

Fie  $\alpha$  o rădăcină simplă a ecuației  $f(x) = 0$  și

$I_\varepsilon = \{x \in \mathbb{R} : |x - \alpha| \leq \varepsilon\}$ . Presupunem că  $f \in C^2[I_\varepsilon]$ .

Definim

$$M(\varepsilon) = \max_{\substack{s \in I_\varepsilon \\ t \in I_\varepsilon}} \left| \frac{f''(s)}{2f'(t)} \right| \quad (24)$$

Dacă  $\varepsilon$  este suficient de mic astfel încât

$$2\varepsilon M(\varepsilon) < 1, \quad (25)$$

atunci pentru orice  $x_0 \in I_\varepsilon$ , metoda lui Newton este bine definită și converge pătratic către singura rădăcină  $\alpha \in I_\varepsilon$ .

Demonstrația: ca la secantă.

# Criteriul de oprire I

## Criteriul de oprire pentru metoda lui Newton

$$|x_n - x_{n-1}| < \varepsilon$$

se bazează pe următoarea propoziție:

### Propoziția 6

*Fie  $(x_n)$  sirul de aproximante generat prin metoda lui Newton. Dacă  $\alpha$  este o rădăcină simplă din  $[a, b]$ ,  $f \in C^2[a, b]$  și metoda este convergentă, atunci există un  $n_0 \in \mathbb{N}$  astfel încât*

$$|x_n - \alpha| \leq |x_n - x_{n-1}|, \quad n > n_0.$$

**Demonstrație.** Vom arăta întâi că

$$|x_n - \alpha| \leq \frac{1}{m_1} |f(x_n)|, \quad m_1 \leq \inf_{x \in [a, b]} |f'(x)|. \quad (26)$$

# Criteriul de oprire II

Utilizând teorema lui Lagrange,

$$f(\alpha) - f(x_n) = f'(\xi)(\alpha - x_n), \text{ cu } \xi \in (\alpha, x_n) \text{ (sau } (x_n, \alpha)).$$

Din relațiile  $f(\alpha) = 0$  și  $|f'(x)| \geq m_1$  pentru  $x \in (a, b)$  rezultă că  $|f(x_n)| \geq m_1 |\alpha - x_n|$ , adică chiar (26).

Pe baza formulei lui Taylor avem

$$f(x_n) = f(x_{n-1}) + (x_n - x_{n-1})f'(x_{n-1}) + \frac{1}{2}(x_n - x_{n-1})^2 f''(\mu), \quad (27)$$

cu  $\mu \in (x_{n-1}, x_n)$  sau  $\mu \in (x_n, x_{n-1})$ .

Tinând cont de modul de obținere a unei aproximații în metoda lui Newton, avem

$f(x_{n-1}) + (x_n - x_{n-1})f'(x_{n-1}) = 0$  și din (27) se obține

$$|f(x_n)| = \frac{1}{2}(x_n - x_{n-1})^2 |f''(\mu)| \leq \frac{1}{2}(x_n - x_{n-1})^2 \|f''\|_\infty,$$

# Criteriul de oprire III

Rezolvarea  
numerică a  
ecuațiilor nelineare

Radu Trîmbițaș

iar pe baza formulei (26) rezultă că

$$|\alpha - x_n| \leq \frac{\|f''\|_{\infty}}{2m_1} (x_n - x_{n-1})^2.$$

Cum am presupus că metoda este convergentă, există un  $n_0$  natural cu proprietatea că

$$\frac{\|f''\|_{\infty}}{2m_1} (x_n - x_{n-1}) < 1, \quad n > n_0$$

și deci

$$|x_n - \alpha| \leq |x_n - x_{n-1}|, \quad n > n_0.$$



Ecuații nelineare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme nelineare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Alegerea valorii de pornire

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- ▶ Alegerea valorii de pornire este, în general, o problemă dificilă.
- ▶ În practică, se alege o valoare, iar dacă după un număr maxim fixat de iterații nu s-a obținut precizia dorită, testată prin unul din criteriile uzuale, se încearcă cu altă valoare de pornire.
- ▶ Criterii

**Criteriul 1** dacă rădăcina este izolată într-un interval  $[a, b]$  și  $f''(x) \neq 0$ ,  $x \in (a, b)$ , un criteriu de alegere este  $f(x_0)f''(x_0) > 0$ .

**Criteriul 2** dacă  $f$  este convexă sau concavă pe  $[a, b]$ ,  $f(a)f(b) < 0$  și tangentele în capete intersectează  $Ox$  în  $(a, b)$ , orice valoare  $x_0$  se poate alege ca valoare de pornire.

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Algoritmul în pseudocod

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

**Intrare:** Funcția  $f$ , derivata  $f'$ , valoarea de pornire  $x_0$ , numarul maxim de iterații,  $Nmax$ , informații de toleranță  $tol$

**Ieșire:** O aproximare a rădăcinii sau un mesaj de eroare

```
1: for  $k := 0$  to  $Nmax$  do
2:    $x_{k+1} := x_k - \frac{f(x_k)}{f'(x_k)}$ ;
3:   if  $crit\_oprire(tol)$  then
4:     return  $x_{k+1};\{\text{Succes}\}$ 
5:   end if
6: end for
7: error("S-a depășit numărul de iterații").
```

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda aproximățiilor succesive I

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- Adesea, în aplicații, ecuațiile neliniare apar sub forma unei *probleme de punct fix*: să se determine  $x$  astfel încât

$$x = \varphi(x). \quad (28)$$

- Un număr  $\alpha$  ce satisfacă această ecuație se numește *punct fix* al lui  $\varphi$ .
- Orice ecuație  $f(x) = 0$  se poate scrie (în multe moduri diferite) în forma echivalentă (28). De exemplu, dacă  $f'(x) \neq 0$ , în intervalul de interes putem lua

$$\varphi(x) = x - \frac{f(x)}{f'(x)}. \quad (29)$$

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda aproximațiilor succesive II

Rezolvarea  
numericală a  
ecuațiilor nelineare

Radu Trîmbițaș

- ▶ Dacă  $x_0$  este o aproximație inițială a unui punct fix  $\alpha$  a lui (28), atunci metoda aproximațiilor succesive generează un sir de aproximații

$$x_{n+1} = \varphi(x_n). \quad (30)$$

- ▶ Dacă acest sir converge și  $\varphi$  este continuă, atunci sirul converge către un punct fix a lui  $\varphi$ .
- ▶ De notat că (30) este chiar metoda lui Newton dacă  $\varphi$  este dată de (29). Astfel metoda lui Newton poate fi privită ca o iterare de tip punct fix, dar nu și metoda secantei.
- ▶ Pentru o iterare de forma (30), presupunând că  $x_n \rightarrow \alpha$  când  $n \rightarrow \infty$ , ordinul de convergență este ușor de determinat.

Ecuații nelineare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme nelineare

Metode  
quasi-Newton

Interolare liniară

Metode de modificare

Interolare inversă

Metode hibride

Bibliografie

# Metoda aproximățiilor succesive III

- ▶ Să presupunem că în punctul fix  $\alpha$  avem

$$\varphi'(\alpha) = \varphi''(\alpha) = \dots = \varphi^{(p-1)}(\alpha) = 0, \quad \varphi^p(\alpha) \neq 0 \quad (31)$$

- ▶ Presupunem că  $\varphi \in C^p$  pe o vecinătate  $V$  a lui  $\alpha$ . Avem atunci, conform teoremei lui Taylor

$$\begin{aligned} \varphi(x_n) &= \varphi(\alpha) + (x_n - \alpha)\varphi'(\alpha) + \dots + \frac{(x_n - \alpha)^{p-1}}{(p-1)!}\varphi^{(p-1)}(\alpha) \\ &\quad + \frac{(x_n - \alpha)^p}{p!}\varphi^{(p)}(\xi_n) = \varphi(\alpha) + \frac{(x_n - \alpha)^p}{p!}\varphi^{(p)}(\xi_n), \end{aligned}$$

unde  $\xi_n \in (\alpha, x_n)$  (sau  $(x_n, \alpha)$ ).

- ▶ Deoarece  $\varphi(x_n) = x_{n+1}$  și  $\varphi(\alpha) = \alpha$  obținem

$$\frac{x_{n+1} - \alpha}{(x_n - \alpha)^p} = \frac{1}{p!}\varphi^{(p)}(\xi_n).$$

# Metoda aproximățiilor succesive IV

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- ▶ Când  $x_n \rightarrow \alpha$ , deoarece  $\xi_n$  este între  $x_n$  și  $\alpha$ , deducem pe baza continuității că

$$\lim_{n \rightarrow \infty} \frac{x_{n+1} - \alpha}{(x_n - \alpha)^p} = \frac{1}{p!} \varphi^{(p)}(\alpha) \neq 0. \quad (32)$$

- ▶ Aceasta ne arată că ordinul de convergență este exact  $p$  și eroarea asimptotică este

$$c = \frac{1}{p!} \varphi^{(p)}(\alpha). \quad (33)$$

- ▶ Combinând aceasta cu condiția uzuală de convergență locală se obține:

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda aproximățiilor succesive V

Rezolvarea  
numerică a  
ecuațiilor nelineare

Radu Trîmbițaș

## Teorema 7

Fie  $\alpha$  un punct fix al lui  $\varphi$  și  $I_\varepsilon = \{x \in \mathbb{R} : |x - \alpha| \leq \varepsilon\}$ . Presupunem că  $\varphi \in C^p[I_\varepsilon]$  și satisfacă (31). Dacă

$$M(\varepsilon) := \max_{t \in I_\varepsilon} |\varphi'(t)| < 1 \quad (34)$$

atunci iterația (30) converge către  $\alpha$ ,  $\forall x_0 \in I_\varepsilon$ . Ordinul de convergență este  $p$ , iar eroarea asimptotică este dată de (33).

Ecuații nelineare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme nelineare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

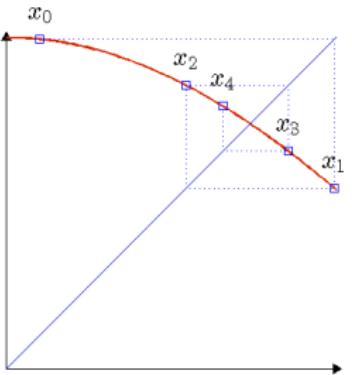
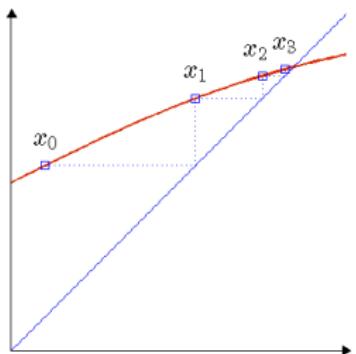
Interpolare inversă

Metode hibride

Bibliografie

# Interpretarea geometrică a metodei aproximățiilor succesive

## Convergență



Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximățiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

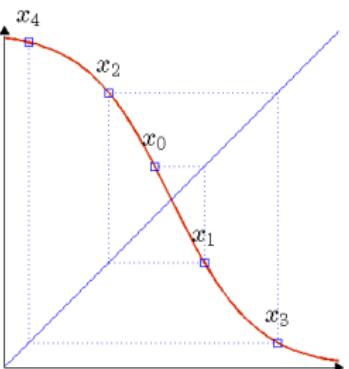
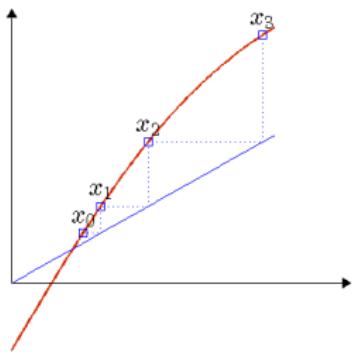
Interpolare inversă

Metode hibride

Bibliografie

# Interpretarea geometrică a metodei aproximățiilor succesive

## Divergență



Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuării neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximățiilor  
succesive

Rădăcini multiple

Ecuării algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda lui Newton pentru rădăcini multiple I

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- Dacă  $\alpha$  este o rădăcină multiplă de ordinul  $m$ , atunci ordinul de convergență a metodei lui Newton este doar 1. Într-adevăr, fie

$$\varphi(x) = x - \frac{f(x)}{f'(x)}.$$

- Deoarece

$$\varphi'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}$$

procesul va fi convergent dacă  $\varphi'(\alpha) = 1 - 1/m < 1$ .

# Metoda lui Newton pentru rădăcini multiple II

- O modalitate de a evita aceasta este să rezolvăm ecuația

$$u(x) := \frac{f(x)}{f'(x)} = 0$$

care are aceleași rădăcini ca și  $f$ , dar simple. Metoda lui Newton pentru problema modificată are forma

$$x_{k+1} = x_k - \frac{u(x_k)}{u'(x_k)} = \frac{f(x_k)f'(x_k)}{[f'(x_k)]^2 - f(x_k)f''(x_k)}. \quad (35)$$

- Deoarece  $\alpha$  este o rădăcină simplă a lui  $u$ , convergența lui (35) este pătratică. Singurul dezavantaj teoretic al lui (35) este derivata a două necesară suplimentar și complexitatea mai mare a calculului lui  $x_{k+1}$  din  $x_k$ . În practică aceasta este o slăbiciune, deoarece numitorul lui (35) poate lua valori foarte mici în vecinătatea lui  $\alpha$  când  $x_k \rightarrow \alpha$ .

# Metoda lui Newton pentru rădăcini multiple III

- ▶ Convergența pătratică a metodei lui Newton se poate realiza nu numai prin modificarea problemei ci și prin modificarea metodei. În vecinătatea unei soluții multiple de ordinul  $m$ ,  $\alpha$ , avem

$$f(x) = (x - \alpha)^m \varphi(x) \approx (x - \alpha)^m \cdot c, \quad (36)$$

de unde rezultă

$$\frac{f(x)}{f'(x)} \approx \frac{x - \alpha}{m} \Rightarrow \alpha \approx x - m \frac{f(x)}{f'(x)}.$$

- ▶ Metoda modificată corespunzătoare

$$x_{k+1} := x_k - m \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots \quad (37)$$

converge pătratic către rădăcina multiplă de ordinul  $m$  când se întrebunează o valoare corectă a lui  $m$  în (37).

# Metoda lui Newton pentru rădăcini multiple IV

- Eficiența variantei (37) a metodei lui Newton depinde de utilizarea unei valori de aproximare bune pentru  $m$ , dacă această valoare nu este cunoscută din alte surse.  
În ipoteza

$$|x_k - \alpha| < |x_{k-1} - \alpha| \wedge |x_k - \alpha| < |x_{k-2} - \alpha|$$

putem înlocui în (36)  $\alpha$  prin  $x_k$

$$f(x_{k-1}) \approx (x_{k-1} - x_k)^m \cdot c$$

$$f(x_{k-2}) \approx (x_{k-2} - x_k)^m \cdot c.$$

- În continuare se obține  $m$ :

$$m \approx \frac{\log [f(x_{k-1})/f(x_{k-2})]}{\log [(x_{k-1} - x_k)/(x_{k-2} - x_k)]}.$$

Această valoare poate fi utilizată în (37).

# Ecuații algebrice I

- ▶ Există multe metode special concepute pentru a rezolva ecuații algebrice.
- ▶ Aici vom descrie numai metoda lui Newton aplicată în acest context, concentrându-ne asupra unui mod eficient de a evalua simultan valoarea polinomului și a primei derivate.
- ▶ Considerăm o ecuație algebraică de grad  $d$

$$f(x) = 0, \quad f(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_0, \quad (38)$$

în care coeficientul dominant se presupune (fără a restrânge generalitatea) să fie egal cu 1 și unde putem presupune, fără a restrânge generalitatea că  $a_0 \neq 0$ .

- ▶ Pentru simplitate vom presupune că toți coeficienții sunt reali.

# Ecuații algebrice II

- ▶ Pentru a aplica metoda lui Newton ecuației (38) este nevoie de o metodă bună de evaluare a polinomului și derivatei. Schema lui Horner este potrivită pentru aşa ceva:

$$bd := 1; \quad cd := 1;$$

**for**  $k = d - 1$  **downto** 1 **do**

$$b_k := tb_{k+1} + a_k;$$

$$c_k := tc_{k+1} + b_k;$$

**end for**

$$b_0 := tb_1 + a_0;$$

- ▶ Atunci  $f(t) = b_0$ ,  $f'(t) = c_1$ .

- ▶ Deci procedăm astfel:

- ▶ Se aplică metoda lui Newton, calculând simultan  $f(x_n)$  și  $f'(x_n)$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

# Ecuații algebrice III

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

- ▶ Se aplică apoi metoda lui Newton polinomului  $\frac{f(x)}{x-\alpha}$ .
- ▶ Pentru rădăcini complexe se începe cu  $x_0$  complex și toate calculele se fac în aritmetică complexă.
- ▶ Este posibil să se împartă cu factori pătratici și să se folosească aritmetică reală – metoda lui Bairstow.
- ▶ Folosind metoda aceasta de scădere a gradului erorile pot fi mari.
- ▶ O modalitate de îmbunătățire este de a utiliza rădăcinile astfel calculate ca aproximării inițiale și a aplica metoda lui Newton polinomului original.

# Metoda lui Newton pentru sisteme neliniare I

Rezolvarea  
numerică a  
ecuațiilor neliniare

- ▶ Metoda lui Newton este ușor de generalizat la sisteme neliniare

$$F(x) = 0, \quad (39)$$

unde  $F : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ , iar  $x, F(x) \in \mathbb{R}^n$ .

- ▶ Sistemul (39) se scrie pe componente

$$\begin{cases} F_1(x_1, \dots, x_n) = 0 \\ \vdots \\ F_n(x_1, \dots, x_n) = 0 \end{cases}$$

- ▶ Fie  $F'(x^{(k)})$  jacobianul lui  $F$  în  $x^{(k)}$ :

$$J := F'(x^{(k)}) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1}(x^{(k)}) & \dots & \frac{\partial F_1}{\partial x_n}(x^{(k)}) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1}(x^{(k)}) & \dots & \frac{\partial F_n}{\partial x_n}(x^{(k)}) \end{bmatrix}. \quad (40)$$

Radu Trîmbițaș

Ecuări neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuări algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda lui Newton pentru sisteme neliniare II

- ▶ Cantitatea  $1/f'(x)$  se înlocuiește în acest caz cu inversa jacobianului în  $x^{(k)}$ :

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}). \quad (41)$$

- ▶ Scriem iterația sub forma

$$x^{(k+1)} = x^{(k)} + w^{(k)}. \quad (42)$$

Se observă că  $w_k$  este soluția sistemului de  $n$  ecuații liniare cu  $n$  necunoscute

$$F' \left( x^{(k)} \right) w^{(k)} = -F(x^{(k)}). \quad (43)$$

- ▶ Este mai eficient și mai convenabil ca, în loc să inversăm jacobianul la fiecare pas, să rezolvăm sistemul (43) și să folosim iterația în forma (42).

Rezolvarea numerică a ecuațiilor neliniare

Radu Trîmbițaș

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda aproximățiilor successive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda lui Newton pentru sisteme neliniare III

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

## Teorema 8

Fie  $\alpha$  o soluție a ecuației  $F(x) = 0$  și presupunem că în bila închisă  $B(\delta) \equiv \{x : \|x - \alpha\| \leq \delta\}$ , există matricea Jacobi a lui  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , este nesingulară și satisface condiția Lipschitz

$$\|F'(x) - F'(y)\|_{\infty} \leq c\|x - y\|_{\infty}, \quad \forall x, y \in B(\delta), \quad c > 0.$$

Punem  $\gamma = c \max \{ \| [F'(x)]^{-1} \|_{\infty} : \|\alpha - x\|_{\infty} \leq \delta \}$  și  $0 < \varepsilon < \min \{ \delta, \gamma^{-1} \}$ . Atunci pentru orice aproximatie initială  $x^{(0)} \in B(\varepsilon) := \{x : \|x - \alpha\|_{\infty} \leq \varepsilon\}$  metoda lui Newton este convergentă, iar vectorii  $e^{(k)} := \alpha - x^{(k)}$  satisfac următoarele inegalități:

- (a)  $\|e^{(k+1)}\|_{\infty} \leq \gamma \|e^{(k)}\|_{\infty}^2$
- (b)  $\|e^{(k)}\|_{\infty} \leq \gamma^{-1} (\gamma \|e^{(0)}\|_{\infty})^{2^k}$ .

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda lui Newton pentru sisteme neliniare IV

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

**Demonstrație.** Dacă  $F'$  este continuă pe segmentul ce unește punctele  $x, y \in \mathbb{R}^n$ , conform teoremei lui Lagrange

$$F(x) - F(y) = J_k(x - y),$$

unde

$$J_k = \begin{bmatrix} \frac{\partial F_1}{\partial x_1}(\xi_1) & \dots & \frac{\partial F_1}{\partial x_n}(\xi_1) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1}(\xi_n) & \dots & \frac{\partial F_n}{\partial x_n}(\xi_n) \end{bmatrix} \Rightarrow$$

$$\begin{aligned} e^{(k+1)} &= e^{(k)} - [F'(x^{(k)})]^{-1}(F(\alpha) - F(x^{(k)})) \\ &= e^{(k)} - [F'(x^{(k)})]^{-1} J_k e^{(k)} \\ &= [F'(x^{(k)})]^{-1}(F'(x^{(k)}) - J_k) e^{(k)} \end{aligned}$$

Ecuății neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuății algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda lui Newton pentru sisteme neliniare V

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

și de aici rezultă imediat (a). Din condiția Lipschitz

$$\|F'(x^{(k)}) - J_k\|_{\infty} \leq c \max_{j=\overline{1,n}} \|x^{(k)} - \xi^{(j)}\| \leq c \|x^{(k)} - \alpha\|$$

Deci, dacă  $\|\alpha - x^{(k)}\|_{\infty} \leq \varepsilon$ , atunci

$$\|\alpha - x^{(k+1)}\|_{\infty} \leq (\gamma \varepsilon) \varepsilon \leq \varepsilon.$$

Deoarece (a) este adevărată pentru orice  $k$ , se obține (b)  
imediat. ■

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda lui Newton - pseudocod

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

**Intrare:** Funcția  $F$ , derivata Fréchet  $F'$ , vectorul de pornire  $x^{(0)}$ , numarul maxim de iterații,  $Nmax$ , informații de toleranță  $tol$

**Ieșire:** O aproximare a rădăcinii sau un mesaj de eroare

```
1: for  $k := 0$  to  $Nmax$  do
2:   Calculează matricea jacobian  $J = F'(x^{(k)})$ ;
3:   Rezolvă sistemul  $Jw = -F(x^{(k)})$ ;
4:    $x^{(k+1)} := x^{(k)} + w$ ;
5:   if  $crit\_oprire(tol)$  then
6:     return  $x^{(k+1)}$ ; {Succes}
7:   end if
8: end for
9: error("S-a depășit numărul de iterații").
```

Ecuății neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuății algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metode quasi-Newton I

- ▶ O slăbiciune semnificativă a metodei lui Newton pentru rezolvarea sistemelor de ecuații neliniare este necesitatea ca la fiecare pas să calculăm matricea jacobiană și să rezolvăm un sistem  $n \times n$  cu această matrice.
- ▶ Pentru a ilustra dimensiunile unei astfel de slabiciuni, să evaluăm volumul de calcule asociat cu o iterare a metodei lui Newton. Matricea jacobiană asociată unui sistem de  $n$  ecuații neliniare scris în forma  $F(x) = 0$  necesită evaluarea celor  $n^2$  derivate parțiale ale celor  $n$  funcții componente ale lui  $F$ . În cele mai multe situații, evaluarea exactă a derivatelor parțiale este neconvenabilă și de multe ori imposibilă. Efortul total pentru o iterare a metodei lui Newton va fi de cel puțin  $n^2 + n$  evaluări de funcții scalare ( $n^2$  pentru evaluarea jacobianului și  $n$  pentru evaluarea lui  $F$ ) și  $O(n^3)$  operații aritmetice pentru a rezolva sistemul liniar.

Rezolvarea numerică a ecuațiilor neliniare

Radu Trîmbițaș

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda aproximățiilor succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metode quasi-Newton II

Acest volum de calcule este prohibitiv, exceptând valori mici ale lui  $n$  și funcții scalare ușor de evaluat.

- ▶ Este firesc ca atenția să fie îndreptată spre reducerea numărului de evaluări și evitarea rezolvării unui sistem liniar la fiecare pas.
- ▶ La metoda secantei aproximarea următoare  $x^{(k+1)}$  se obține ca soluție a ecuației liniare

$$\bar{\ell}_k = f(x^{(k)}) + (x - x^{(k)}) \frac{f(x^{(k)} + h_k) - f(x^{(k)})}{h_k} = 0.$$

- ▶ Aici funcția  $\bar{\ell}_k$  poate fi interpretată în două moduri:
  1. ca aproximare a ecuație tangentei

$$\ell_k(x) = f(x^{(k)}) + (x - x^{(k)}) f' \left( x^{(k)} \right);$$

2. ca interpolare liniară între punctele  $x^{(k)}$  și  $x^{(k+1)}$ .

# Metode quasi-Newton III

- ▶ Se pot obține diverse generalizări ale metodei secantei la sisteme de ecuații neliniare în funcție de modul în care se interpretează  $\bar{\ell}_k$ .
- ▶ Prima interpretare conduce la metode de tip Newton discretizate, iar a doua la metode bazate pe interpolare.
- ▶ Metodele de tip Newton discretizate se obțin dacă în metoda lui Newton (41)  $F'(x)$  se înlocuiește cu o aproximare discretă  $A(x, h)$ . Derivatele parțiale din matricea jacobiană (40) se vor înlocui prin diferențele divizate

$$A(x, h)e_i := [F(x + h_i e_i) - F(x)] / h_i, \quad i = \overline{1, n}, \quad (44)$$

Ecuății neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda aproximațiilor successive

Rădăcini multiple

Ecuății algebrice

Sisteme neliniare

Metode quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metode quasi-Newton IV

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

unde  $e_i \in \mathbb{R}^n$  este al  $i$ -lea vector al bazei canonice și  $h_i = h_i(x)$  este mărimea pasului de discretizare. O alegere posibilă a pasului este de exemplu

$$h_i := \begin{cases} \varepsilon |x_i|, & \text{dacă } x_i \neq 0; \\ \varepsilon, & \text{altfel,} \end{cases}$$

cu  $\varepsilon := \sqrt{\text{eps}}$ , unde eps este epsilon-ul mașinii.

# Interpolare liniară I

- ▶ La interpolare fiecare dintre planele tangente se înlocuiește cu un (hiper)plan care interpolează funcțiile componente  $F_i$  ale lui  $F$  în  $n+1$  puncte date  $x^{k,j}$ ,  $j = \overline{0, n}$ , într-o vecinătate a lui  $x^{(k)}$ , adică se determină vectorii  $a^{(i)}$  și scalarii  $\alpha_i$ , astfel încât pentru

$$L_i(x) = \alpha_i + a^{(i)T} x, \quad i = \overline{1, n} \quad (45)$$

are loc

$$L_i(x^{k,j}) = F_i(x^{k,j}), \quad i = \overline{1, n}, \quad j = \overline{0, n}.$$

- ▶ Următoarea aproximație  $x^{(k+1)}$  se obține ca punct de intersecție între cele  $n$  hiperplane (45) din  $\mathbb{R}^{n+1}$  cu hiperplanul  $y = 0$ .  $x^{(k+1)}$  rezultă ca soluție a sistemului de ecuații liniare

$$L_i(x) = 0, \quad i = \overline{1, n}. \quad (46)$$

# Interpolare liniară II

Rezolvarea  
numericală a  
ecuațiilor neliniare

Radu Trîmbițaș

Ecuății neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuății algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

- ▶ În funcție de alegerea punctelor de interpolare se obțin diferite metode, dintre care cele mai cunoscute sunt metoda lui Brown și metoda lui Brent.
- ▶ Metoda lui Brown combină aproximarea lui  $F'$  și rezolvarea sistemului prin eliminare gaussiană.
- ▶ În metoda lui Brent se întrebunează la rezolvarea sistemului metoda QR. Ambele metode aparțin unei clase de metode, care, la fel ca metoda lui Newton, converg pătratic, dar au nevoie doar de  $(n^2 + 3n)/2$  evaluări de funcții pe iterație.
- ▶ Într-un studiu comparativ, Moré și Cosnard [7] au ajuns la concluzia că metoda Brent este adeseori de preferat metodei lui Brown și că pentru sisteme de ecuații neliniare, la care evaluarea lui  $f$  necesită un efort mai mic, metoda lui Newton discretizată este cea mai eficientă metodă de rezolvare.

# Metode de modificare I

- ▶ Din punct de vedere al efortului de calcul, sunt deosebit de convenabile metodele în care la fiecare pas se întrebunează o aproximare  $A_k$  a lui  $F'(x^{(k)})$ , care se obține din  $A_{k-1}$  printr-o modificare de rang 1, adică prin adăugarea unei matrice de rang 1:

$$A_{k+1} := A_k + u^{(k)} \begin{bmatrix} v^{(k)} \end{bmatrix}^T, \quad u^{(k)}, v^{(k)} \in \mathbb{R}^n, \quad k = 0, 1, 2, \dots$$

- ▶ Pe baza formulei Sherman-Morrison (vezi [4])

$$(A + uv^T)^{-1} = A^{-1} - \frac{1}{1 + v^T A^{-1} u} A^{-1} u v^T A^{-1} \quad (47)$$

pentru  $B_{k+1} := A_{k+1}^{-1}$  are loc relația de recurență

$$B_{k+1} = B_k - \frac{B_k u^{(k)} \begin{bmatrix} v^{(k)} \end{bmatrix}^T B_k}{1 + [v^{(k)}]^T B_k u^{(k)}}, \quad k = 0, 1, 2, \dots,$$

## Metode de modificare II

atât timp cât  $1 + \left[ v^{(k)} \right]^T B_k u^{(k)} \neq 0$ .

- ▶ Necesitatea rezolvării unui sistem liniar la fiecare pas dispără; aceasta se înlocuiește cu înmulțiri matrice-vector, ceea ce corespunde unei reduceri a efortului de calcul de la  $O(n^3)$  la  $O(n^2)$ .
- ▶ Acest avantaj va fi plătit prin aceea că nu vom mai avea o convergență pătratică ca la metoda lui Newton, ci doar una superliniară:

$$\lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - \alpha\|}{\|x^{(k)} - \alpha\|} = 0. \quad (48)$$

- ▶ În metoda lui Broyden alegerea vectorilor  $u^{(k)}$  și  $v^{(k)}$  are loc după principiul aproximăției secantei. În cazul scalar aproximarea  $a_k \approx f'(x^{(k)})$  se face unic prin

$$a_{k+1}(x^{(k+1)} - x^{(k)}) = f(x^{(k+1)}) - f(x^{(k)}).$$

# Metode de modificare III

- ▶ Pentru  $n > 1$ , din contră, aproximarea

$$A_{k+1}(x^{(k+1)} - x^{(k)}) = F(x^{(k+1)}) - F(x^{(k)}) \quad (49)$$

(aşa numita ecuație quasi-Newton) nu mai este unic determinată; orice altă matrice de forma

$$\bar{A}_{k+1} := A_{k+1} + pq^T$$

cu  $p, q \in \mathbb{R}^n$  și  $q^T(x^{(k+1)} - x^{(k)}) = 0$  verifică de asemenea ecuația (49).

- ▶ Pe de altă parte,

$$y_k := F(x^{(k)}) - F(x^{(k-1)}) \text{ și } s_k := x^{(k)} - x^{(k-1)}$$

conțin numai informații despre variația lui  $F$  în direcția  $s_k$ , dar nici o informație în direcții ortogonale pe  $s_k$ .

# Metode de modificare IV

Rezolvarea  
numerică a  
ecuațiilor neliniare

- ▶ Pe aceste direcții trebuie ca efectul lui  $A_{k+1}$  și  $A_k$  să coincidă

$$A_{k+1}q = A_k q, \quad \forall q \in \{v : v \neq 0, v^T s_k = 0\}. \quad (50)$$

- ▶ Pornind de la prima aproximare  $A_0 \approx F'(0)$ , se generează sirul  $A_1, A_2, \dots$  utilizând formulele (49) și (50) (Broyden [2], Dennis și Moré [4]).
- ▶ Pentru sirul  $B_0 = A_0^{-1} \approx [F(x^{(0)})]^{-1}, B_1, B_2, \dots$  cu ajutorul formulei Sherman-Morrisson (47) se obține relația de recurență

$$B_{k+1} := B_k + \frac{(s_{k+1} - B_k y_{k+1}) s_{k+1}^T B_k}{s_{k+1}^T B_k y_{k+1}}, \quad k = 0, 1, 2, \dots$$

care conține doar înmulțiri matrice vector și a cărei complexitate este doar  $O(n^2)$ .

Radu Trîmbițaș

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metode de modificare V

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- ▶ Cu ajutorul matricelor  $B_k$  se poate defini metoda lui Broyden prin

$$x^{(k+1)} := x^{(k)} - B_k F(x^{(k)}), \quad k = 0, 1, 2, \dots$$

- ▶ Această metodă converge superliniar în sensul lui (48), dacă pașii  $s_k$  se apropie asymptotic (când  $k \rightarrow \infty$ ) de pașii metodei lui Newton.
- ▶ Se poate recunoaște în aceasta semnificația centrală a principiului linearizării locale la rezolvarea ecuațiilor neliniare.

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

# Metoda lui Broyden

**Intrare:**  $F$ , vectorul  $x^{(0)}$ ,  $Nmax$ , toleranța  $tol$

**Ieșire:** O aproximatie a rădăcinii sau un mesaj de eroare

$B_0 := F'(x^{(0)})$ ;  $v := F(x)$ ;  $B := B_0^{-1}$ ;

$s := -Bv$ ;  $x := x + s$ ;

**for**  $k := 1$  **to**  $Nmax$  **do**

$w := v$ ;       $v := F(x)$ ;       $y := v - w$ ;

$z := -By$ ;  $\{z = -B_{k-1}y_k\}$

$p := -s^T z$ ;  $\{p = s_k^T B_{k-1}y_k\}$

$C := pl + (s + z)s^T$ ;

$\{C = s_k^T B_{k-1}^{-1}y_k I + (s_k + B_{k-1}y_k)s_k^T\}$

$B := (1/p)CB$ ;  $\{B = B_k\}$

$s := -Bv$ ;  $\{s = -B_k F(x^{(k)})\}$

$x := x + s$ ;

**if**  $crit\_oprire(tol)$  **then**

**return**  $x$ ;  $\{\text{succes}\}$

**end if**

**end for**

**error**("S-a depășit numărul maxim de iterații")

# Interolare inversă I

- Dacă  $f$  este inversabilă pe o vecinătate  $V$  a lui  $\alpha$  și  $g$  este inversa sa ( $g = f^{-1}$ ), atunci

$$f(\alpha) = 0 \implies \alpha = g(0).$$

- Interolare inversă constă în aproximarea lui  $g(0)$  prin valoarea unui polinom de interpolare de grad mic.
- Dacă aproximăm  $g$  prin polinomul său Taylor de grad 1, avem

$$g(y) \approx (T_1 g)(y) = g(y_n) + (y - y_n)g'(y_n).$$

- Dacă  $y_n = f(x_n)$ , ținând cont că  $g'(y_n) = \frac{1}{f'(x_n)}$ , se obține

$$g(0) \approx x_n - \frac{f(x_n)}{f'(x_n)} =: x_{n+1},$$

adică metoda lui Newton! Încercați să obțineți metoda corespunzătoare pentru  $T_2 g$ .

# Interpolare inversă II

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

- Dacă luăm  $g \approx L_1 g$ , avem

$$g(y) \approx (L_1 g)(y) = g(y_n) + g[y_n, y_{n-1}](y - y_n).$$

- Tinând cont că

$$g[y_n, y_{n-1}] = \frac{g(y_n) - g(y_{n-1})}{y_n - y_{n-1}} = \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})},$$

se obține

$$g(0) \approx x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n),$$

adică metoda secantei.

Ecuații neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie

- ▶ Aceste metode combină metodele cu convergență globală, dar mai lente, cu metode cu convergență locală (de exemplu, Newton sau secantă).
- ▶ De asemenea, se utilizează scheme adaptive de monitorizare a iterațiilor și de testare a condițiilor de oprire.
- ▶ Una dintre cele mai cunoscute metode de acest tip este algoritmul lui Dekker, în varianta lui Brent, cunoscut și sub numele de *algoritmul Dekker-Brent sau zeroïn* [6],[8].
- ▶ El combină metoda înjumătățirii cu metoda secantei și cu metoda interpolării inverse pătratice (IQI).
- ▶ Funcția MATLAB fzero se bazează pe acest algoritm.

Ecuății neliniare

Ordin de  
convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda  
aproximațiilor  
succesive

Rădăcini multiple

Ecuății algebrice

Sisteme neliniare

Metode  
quasi-Newton

Interpolate liniară

Metode de modificare

Interpolate inversă

Metode hibride

Bibliografie

# Descrierea algoritmului

- ▶ Se începe cu  $a$  și  $b$  astfel încât  $f(a)$  și  $f(b)$  au semne opuse.
- ▶ Se utilizează un pas al metodei secantei pentru a obține  $c$  între  $a$  și  $b$ .
- ▶ Se repetă pașii următori până când  $|b - a| < \varepsilon |b|$  sau  $f(b) = 0$ 
  - ▶ Se permută  $a$ ,  $b$ ,  $c$  astfel încât
    - ▶  $f(b)$  și  $f(a)$  au semne opuse
    - ▶  $|f(b)| \leq |f(a)|$
    - ▶  $c$  este valoarea precedentă a lui  $b$ .
  - ▶ Dacă  $c \neq a$  se realizează un pas IQI, altfel un pas al metodei secantei.
  - ▶ Dacă rezultatul pasului IQI sau secantei este în  $[a, b]$ , se acceptă
  - ▶ Dacă nu, înjumătățire.

# Bibliografie I

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

-  Octavian Agratini, Ioana Chiorean, Gheorghe Coman, Trîmbițaș Radu, *Analiză numerică și teoria aproximării*, vol. III, Presa Universitară Clujeană, 2002, coordonatori D. D. Stancu și Gh. Coman.
-  C. G. Broyden, A Class of Methods for Solving Nonlinear Simultaneous Equations, *Math. Comp.* **19** (1965), 577–593.
-  Gheorghe Coman, *Analiză numerică*, Editura Libris, Cluj-Napoca, 1995.
-  J. E. Dennis, J. J. Moré, Quasi-Newton Methods, Motivation and Theory, *SIAM Review* **19** (1977), 46–89.
-  W. Gautschi, *Numerical Analysis. An Introduction*, Birkhäuser, Basel, 1997.

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda aproximațiilor succesive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode quasi-Newton

Interpolate liniară

Metode de modificare

Interpolate inversă

Metode hibride

Bibliografie

# Bibliografie II

Rezolvarea  
numerică a  
ecuațiilor neliniare

Radu Trîmbițaș

-  C. Moler, *Numerical Computing in MATLAB*, SIAM, 2004
-  J. J. Moré, M. Y. Cosnard, Numerical Solutions of Nonlinear Equations, *ACM Trans. Math. Softw.* **5** (1979), 64–85.
-  W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, New York, Port Chester, Melbourne, Sidney, 1996, disponibila prin [www, http://www.nr.com/](http://www.nr.com/).
-  J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, 2nd ed., Springer Verlag, 1992.

Ecuații neliniare

Ordin de convergență

Falsa poziție

Metoda secantei

Metoda lui Newton

Metoda aproximațiilor successive

Rădăcini multiple

Ecuații algebrice

Sisteme neliniare

Metode quasi-Newton

Interpolare liniară

Metode de modificare

Interpolare inversă

Metode hibride

Bibliografie