# Analiză numerică

Radu Tiberiu Trîmbiţaş

# Prefaţă

Lloyd N. Trefethen a propus următoarea definiţie a Analizei numerice:

> *Analiza numerică este studiul algoritmilor pentru rezolvarea problemelor matematicii continue.*

Cuvântul cheie este acela de *algoritmi*. Deşi foarte multe lucrări *nu* evidenţiază acest lucru, în centrul atenţiei Analizei numerice stă proiectarea şi analiza algoritmilor de rezolvare a unei anumite clase de probleme.

Problemele sunt cele din *matematica continuă*. „Continuă" înseamnă aici faptul că variabilele ce intervin aici sunt reale sau complexe; opusul lui continuu este discret. Pe scurt, am putea spune că Analiza numerică este Algoritmică continuă, în contrast cu Algoritmica clasică, care este Algoritmică discretă.

Este clar că deoarece numerele reale şi complexe nu pot fi reprezentate exact în calculator, ele trebuie să fie aproximate printr-o reprezentare finită. Din acest moment intervin erorile de rotunjire şi iar este clar că studiul lor este unul din obiectivele importante ale Analizei numerice. Au existat şi mai există încă opinii care susţin că acesta este cel mai important obiectiv. Un argument în sprijinul acestei idei, înafară de omniprezenţa erorilor, este dat de metodele exacte de rezolvare a sistemelor de ecuaţii liniare, cum ar fi eliminarea gaussiană.

Dar, cele mai multe probleme ale matematicii continue nu pot fi rezolvate prin algoritmi aşa-zişi finiţi, chiar presupunând prin absurd că am lucra în aritmetică cu precizie exactă. Un prim exemplu care se poate da aici este problema rezolvării unei ecuaţii polinomiale. Acest lucru se evidenţiază la problemele de valori şi vectori proprii, dar apar în orice problemă ce presupune termeni neliniari sau derivate – determinarea zerourilor, cuadraturi, ecuaţii diferenţiale şi integrale, optimizare ş.a.m.d.

Chiar dacă erorile de rotunjire ar dispare, Analiza numerică ar rămâne. Aproximarea numerelor, obiectivul aritmeticii în virgulă flotantă, este un subiect dificil şi obo-

sitor. Un obiectiv mai profund al Analizei numerice este aproximarea necunoscutelor, nu a cantităţilor cunoscute. Scopul este convergenţa rapidă a aproximaţiilor şi mândria specialiştilor din acest domeniu este aceea că, pentru multe probleme s-au inventat algoritmi care converg extrem de rapid. Dezvoltarea pachetelor de calcul simbolic a micşorat importanţa erorilor de rotunjire, fără a micşora importanţa vitezei de convergenţă a algoritmilor.

Definiţia de mai sus nu surprinde câteva aspecte importante : că aceşti algoritmi sunt implementaţi pe calculatoare, a căror arhitectură poate fi o parte importantă a problemei; că fiabilitatea şi eficienţa sunt obiective supreme; că anumiţi specialişti în analiza numerică scriu programe şi alţii demonstrează teoreme[1]; şi lucrul cel mai important, că toată munca este *aplicată*, aplicată cu succes la mii de aplicaţii, pe milioane de computere din toată lumea. ,,Problemele matematicii continue" sunt problemele pe care ştiinţa şi ingineria sunt construite; fără metode numerice, ştiinţa şi ingineria, aşa cum sunt ele practicate astăzi ar ajunge repede în impas. Ele sunt de asemenea problemele care au preocupat cei mai mulţi matematicieni de la Newton până azi. La fel ca şi cei ce se ocupă de matematica pură, specialiştii în Analiza numerică sunt moştenitorii marii tradiţii a lui Euler, Lagrange, Gauss şi a altor mari matematicieni.

Radu Tiberiu Trîmbiţaş
Cluj-Napoca, iulie 2003

[1] există mulţi specialişti foarte buni care le fac pe amândouă

# Cuprins

# Lista algoritmilor

CAPITOLUL 1

# Elemente de Teoria erorilor şi aritmetică în virgulă flotantă

## Cuprins

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*;
2. *erori de rotunjire*;
3. *erori de aproximare.*

Erorile în datele de intrare sunt în afara (dincolo de) controlului calculelor. Ele se pot datora, de exemplu, imperfecţiunilor inerente ale măsurătorilor fizice.

Erorile de rotunjire apar dacă se fac calcule cu numere a căror reprezentare se restrânge la un număr finit de cifre, aşa cum se întâmplă de obicei.

Pentru al treilea tip de erori, multe metode nu dau soluţia exactă a problemei date $\mathrm{P}$, chiar dacă calculele se fac fără rotunjire, ci mai degrabă soluţia unei alte probleme mai simple $\widetilde{\mathrm{P}}$, care aproximează $\mathrm{P}$. De exemplu, problema $\mathrm{P}$ a însumării unei serii infinite:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots$$

poate fi înlocuită cu problema mai simplă $\widetilde{\mathrm{P}}$ de a însuma numai un număr finit de termeni ai seriei. Eroarea de aproximare astfel obţinută se numeşte *eroare de trunchiere* (totuşi, acest termen este de asemenea utilizat pentru erorile de rotunjire comise prin ştergerea ultimelor cifre ale reprezentării). Multe probleme de aproximare se se obţin prin ,,discretizarea“ problemei originale $\mathrm{P}$: integralele definite se aproximează prin sume finite, derivatele prin diferenţe, etc. În astfel de cazuri, eroarea de aproximare se numeşte *eroare de discretizare.* Unii autori extind termenul de ,,eroare de trunchiere“ pentru a acoperi şi eroarea de discretizare.

În acest capitol vom examina efectul general al erorilor de intrare şi de rotunjire asupra rezultatelor unui calcul. Erorile de aproximare vor fi discutate în capitolele următoare când vom trata metodele numerice individual.

## 1.1. Probleme numerice

Combinaţia dintre o problemă matematică (PM), (de natură constructivă) şi specificaţiile de precizie ale rezultatului (SP) se numeşte *problemă numerică*.

**Exemplul 1.1.1.** Fie $f : \mathbb{R} \longrightarrow \mathbb{R}$ şi $x \in \mathbb{R}$. 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 $x$ şi $f$, să se calculeze $f(x)$;

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

Problemele numerice aparţin uneia din următoarele categorii:

**Evaluarea unei funcţionale**: $l : \mathcal{F} \to \mathbb{R}$, cum ar fi, de exemplu, calcularea valorii unei funcţii $f(x)$, a derivatelor $f'(x), f''(x), \ldots$ (derivare numerică), a integralelor definite $\int_a^b f(t)dt$ (integrare numerică) şi a normelor $\|f\|_p$, etc.

**Rezolvarea ecuaţiilor algebrice**: determinarea valorilor unor necunoscute aflate în relaţii algebrice prin rezolvarea unor sisteme de ecuaţii liniare sau neliniare.

**Rezolvarea unor ecuaţii analitice**: determinarea funcţiilor (sau valorilor de funcţii) soluţii ale unei ecuaţii operatoriale, cum ar fi ecuaţiile diferenţiale ordinare sau cu derivate parţiale, ecuaţiile integrale, ecuaţii funcţionale, etc.

**Probleme de optimizare**: determinarea unor valori numerice particulare ale unor funcţii, care optimizează (minimizează sau maximizează) o funcţie obiectiv, cu restricţii sau fără restricţii. Problemele matematice constructive, din care provin problemele numerice, pot fi privite ca o aplicaţie abstractă $F : X \to Y$ între două spaţii liniare normate.

În funcţie de care dintre cantităţile $y, x$ sau $F$ este necunoscută în ecuaţia

$$Fx = y,$$

avem de-a face cu o *problemă directă*, o *problemă inversă* sau o *problemă de identificare*:

| | $F$ | $x$ | $y$ |
|---|---|---|---|
| problemă directă | dată | dat | **dorit** |
| problemă inversă | dată | **dorit** | dat |
| problemă de identificare | **dorită** | dat | dat |

**Exemplul 1.1.2 (Integrare).** Evaluarea unei integrale definite

$$Fx = \int_a^b x(t)dt$$

este o *problemă directă*. În această problemă $F : \mathcal{F} \to \mathbb{R}$ este o funcţională care aplică, de exemplu $\mathcal{F} = C[a, b]$ pe $\mathbb{R}$. Integrandul – în acest caz o funcţie $x$ definită pe $[a, b]$ – este dat, iar integrala $y = Fx$ trebuie determinată. $\diamondsuit$

**Exemplul 1.1.3 (Sistem de ecuaţii liniare).** Soluţia sistemului de ecuaţii liniare

$$Ax = b, \qquad A \in \mathbb{R}^{n\times n}, \qquad x, b \in \mathbb{R}^n$$

este o *problemă inversă* tipică. Aplicaţia liniară $F$ este caracterizată de $n^2$ coeficienţi (elementele lui $A$). Vectorul $b$ (imaginea) este dat, iar vectorul $x$ (care este transformat în $b$) trebuie determinat. $\diamondsuit$

**Exemplul 1.1.4 (Analiza unei mixturi).** Trebuie determinată compoziţia unei mixturi de substanţe care se evaporă. Presupunând că în unitatea de timp se evaporă o cantitate fixată din fiecare substanţă, se poate utiliza următorul model dependent de timp al mixturii

$$y(t) = \sum_{i=1}^{m} y_{i_0} e^{-k_i t}. \tag{1.1.1}$$

În afară de cantităţile iniţiale $y_{i_0}$, $i = 1, 2, \ldots, m$ şi de ratele de evaporare $k_1, \ldots, k_m$ ale substanţelor necunoscute, numărul $m$ de substanţe este de asemenea necunoscut. Dacă la momentele $t_1, \ldots, t_n$ se măsoară cantităţile $y_1, \ldots, y_n$, trebuie determinaţi parametrii $y_{i_0}$, $k_i$, $i = \overline{1, m}$ şi $m$ ai aplicaţiei

$$F : (t_1, \ldots, t_n) \mapsto (y_1, \ldots, y_n) = \left( \sum_{i=1}^{m} y_{i_0} e^{-k_i t_1}, \ldots, \sum_{i=1}^{m} y_{i_0} e^{-k_i t_n} \right);$$

deci avem o *problemă de identificare*. Pentru orice $m \in \{1, 2, \ldots, n/2\}$ (cel mult $2 \cdot n/2 = n$ necunoscute în acest caz) se poate încerca minimizarea distanţei între funcţia model (1.1.1) şi valorile date, care poate fi definită prin

$$r_m = \sum_{j=1}^{n} \left( y_i - \sum_{i=1}^{m} y_{i_0} e^{-k_i t_j} \right)^2. \tag{1.1.2}$$

Aceasta este o problemă inversă extrem de dificilă (problemă de estimaţie neliniară). Minimul trebuie determinat folosind tehnici de optimizare cu restricţii. Valoarea reziduului minim $r_m^*$ face posibilă determinarea valorii lui $m$ care poate fi utilizată pentru cea mai bună adaptare a modelului la datele de intrare. Totuşi, aceasta nu înseamnă minimizarea lui $r_m$ fără restricţii asupra lui $m$, ci mai degrabă a nu alege $m$ mai mare decât este necesar, în scopul de a obţine modelul cel mai simplu posibil. ♢

## 1.2. Măsuri ale erorii

**Definiţia 1.2.1** *Fie $X$ un spaţiu liniar normat, $A \subseteq X$ şi $x \in X$. Un element $x^* \in A$ se numeşte* aproximantă *a lui $x$ din $A$ (notaţie $x^* \approx x$).*

**Definiţia 1.2.2** *Dacă $x^*$ este o aproximantă a lui $x$ diferenţa $\Delta x = x - x^*$ se numeşte* eroare, *iar*

$$\|\Delta x\| = \|x^* - x\| \tag{1.2.1}$$

*se numeşte* eroare absolută.

**Definiţia 1.2.3** *Expresia*

$$\delta x = \frac{\|\Delta x\|}{\|x\|}, \quad x \neq 0 \tag{1.2.2}$$

*se numeşte* eroare relativă.

**Observaţia 1.2.4.**

1. Deoarece în practică $x$ este necunoscut, se foloseşte aproximarea $\delta x = \frac{\|\Delta x\|}{\|x^*\|}$. Dacă $\|\Delta x\|$ este mic comparativ cu $x^*$, atunci aproximanta este bună.
2. Dacă $X = \mathbb{R}$, se lucrează cu $\delta x = \frac{\Delta x}{x}$ şi $\Delta x = x^* - x$. $\diamondsuit$

# 1.3. Eroarea propagată

Fie $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $x = (x_1, \ldots, x_n)$ şi $x^* = (x_1^*, \ldots, x_n^*)$. Dorim să evaluăm eroarea absolută şi relativă $\Delta f$ şi respectiv $\delta f$ când se aproximează $f(x)$ prin $f(x^*)$. Aceste erori se numesc *erori propagate*, deoarece ne spun cum se propagă eroarea iniţială (absolută sau relativă) pe parcursul calculării lui $f$. Să presupunem că $x = x^* + \Delta x$, unde $\Delta x = (\Delta x_1, \ldots, \Delta x_n)$. Pentru eroarea absolută avem (folosind formula lui Taylor)

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

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

Dacă $\Delta x_i$ sunt suficient de mici, atunci $\Delta x_i \Delta x_j$ sunt neglijabile comparativ cu $\Delta x_i$ şi obţinem

$$\Delta f \approx \sum_{i=1}^{n} \Delta x_i \frac{\partial f}{\partial x_i^*}(x_1^*, \ldots x_n^*). \tag{1.3.1}$$

Pentru eroarea relativă avem

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

Deci

$$\delta f = \sum_{i=1}^{n} x_i^* \frac{\partial}{\partial x_i^*} \ln f(x^*) \delta x_i. \tag{1.3.2}$$

De o mare importanţă practică este şi problema inversă: cu ce precizie trebuie aproximate 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 $\delta f < \varepsilon$? O metodă de rezolvare se bazează pe *principiul efectelor egale*: se presupune că toţi termenii care intervin în (1.3.1) sau (1.3.2) au acelaşi efect, adică

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

Din (1.3.1) se obţine

$$\Delta x_i \approx \frac{\Delta f}{n\left|\frac{\partial f}{\partial x_i^*}(x^*)\right|}. \tag{1.3.3}$$

Analog,

$$\delta x_i = \frac{\delta f}{n\left|x_i^* \frac{\partial}{\partial x_i^*} \ln f(x^*)\right|}. \tag{1.3.4}$$

# 1.4. Reprezentarea în virgulă flotantă

## 1.4.1. Parametrii reprezentării

Deşi au fost propuse mai multe reprezentări pentru numerele reale, reprezentarea în virgulă flotantă este cea mai răspândită. Parametrii reprezentării în virgulă flotantă sunt baza $\beta$ (întotdeauna pară), precizia $p$, exponentul maxim $e_{\max}$ şi cel minim $e_{\min}$, toate numere naturale. În general, un număr în virgulă flotantă se reprezintă sub forma

$$x = \pm d_0.d_1d_2\ldots d_{p-1} \times \beta^e, \quad 0 \le d_i < \beta \tag{1.4.1}$$

unde $d_0.d_1d_2\ldots d_{p-1}$ se numeşte *semnificant* sau *mantisă*, iar $e$ *exponent* . Valoarea lui $x$ este

$$\pm(d_0 + d_1\beta^{-1} + d_2\beta^{-2} + \cdots + d_{p-1}\beta^{-(p-1)})\beta^e. \tag{1.4.2}$$

Pentru ca reprezentarea în virgulă flotantă să fie unică, numerele flotante se *normalizează*, adică se modifică reprezentarea (nu valoarea) astfel încât $d_0 \neq 0$. Zero se reprezintă ca $1.0 \times \beta^{e_{\min}-1}$. În acest mod ordinea numerică uzuală a numerelor reale nenegative corespunde ordinii lexicografice a reprezentării lor flotante (cu exponentul în stânga semnificantului).

Termenul de *număr în virgulă flotantă* este utilizat pentru a desemna un număr real care poate fi reprezentat exact în virgulă flotantă.

Distribuţia numerelor în virgulă flotantă pe axa reală apare în figura 1.1.

Fiecare interval de forma $[\beta^e, \beta^{e+1})$ din $\mathbb{R}$ conţine $\beta^p$ numere în virgulă flotantă (numărul posibil de semnificanţi). Intervalul $(0, \beta^{e_{\min}})$ este gol şi din acest motiv se introduc *numerele denormalizate*, adică numere cu semnificantul de forma $0.d_1d_2\ldots d_{p-1}$

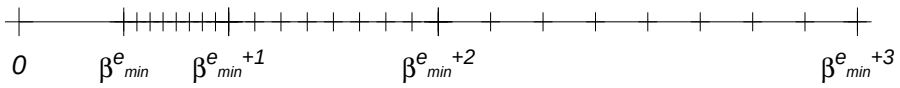



Figura 1.1: Distribuţia numerelor în virgulă flotantă pe axa reală (fără denormalizare)

şi exponentul $\beta^{e_{\min}-1}$. Faptul că se admit numere denormalizate sau nu se consideră a fi un parametru suplimentar al reprezentării. Mulţimea numerelor în virgulă flotantă pentru un set de parametri daţi ai reprezentării se va nota cu

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

Distribuţia numerelor în virgulă flotantă când se admite denormalizarea apare în figura 1.2.

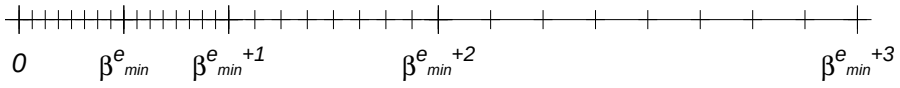



Figura 1.2: Distribuţia numerelor în virgulă flotantă pe axa reală (cu denormalizare)

Această mulţime nu coincide cu $\mathbb{R}$ din următoarele motive:

1. este o submulţime finită a lui $\mathbb{Q}$;
2. pentru $x \in \mathbb{R}$ putem avea $|x| > \beta \times \beta^{e_{\max}}$ (depăşire superioară) sau $|x| < 1.0 \times \beta^{e_{\min}}$ (depăşire inferioară).

Operaţiile aritmetice uzuale pe $\mathbb{F}(\beta, p, e_{\min}, e_{\max}, \mathrm{denorm})$ se notează cu $\oplus$, $\ominus$, $\otimes$, $\oslash$, iar funcţiile uzuale cu SIN, COS, EXP, LN, SQRT ş.a.m.d. $(\mathbb{F}, \oplus, \otimes)$ nu este corp deoarece

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

Pentru măsurarea erorii de reprezentare, în afară de eroare relativă, se foloseşte *ulps* – *u*nits in the *l*ast *p*lace (unităţi în ultima poziţie). Dacă numărul $z$ se reprezintă prin $d_0.d_1d_2\ldots d_{p-1}\times\beta^e$, atunci eroarea este

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

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 $\underbrace{0.0\ldots0}_{p}\beta'\times\beta^e$, cu $\beta'=\frac{\beta}{2}$, iar numitorul este cuprins între $\beta^e$ şi $\beta\times\beta^e$. Valoarea $\text{eps}=\frac{\beta}{2}\beta^{-p}$ se numeşte *epsilon-ul maşinii*.

Rotunjirea implicită se face după regula cifrei pare: dacă $x=d_0.d_1\ldots d_{p-1}d_p\ldots$ şi $d_p>\frac{\beta}{2}$ rotunjirea se face în sus, dacă $d_p<\frac{\beta}{2}$ rotunjirea se face în jos, iar dacă $d_p=\frac{\beta}{2}$ şi printre cifrele eliminate există una nenulă rotunjirea se face în sus, iar în caz contrar ultima cifră păstrată este pară. Dacă notăm cu $\text{fl}$ operaţia de rotunjire, operaţiile aritmetice din $\mathbb{F}$ se pot defini prin

$$x\circledcirc y=\text{fl}(x\circ y). \tag{1.4.3}$$

Se pot alege şi alte variante de rotunjire – cât mai depărtat de 0, spre $-\infty$, spre $+\infty$, spre 0 (trunchiere). În raţionamentele asupra operaţiilor în virgulă flotantă vom folosi următorul model

$$\forall x,y\in\mathbb{F},\ \exists\delta\text{ cu }|\delta|<\text{eps}\text{ astfel încât }x\circledcirc y=(x\circ y)(1+\delta). \tag{1.4.4}$$

În cuvinte, orice operaţie în aritmetica în virgulă flotantă este exactă până la o eroare relativă de cel mult $\text{eps}$.

Formula (1.4.4) se numeşte *axioma fundamentală a aritmeticii în virgulă flotantă*.

## 1.4.2. Anularea

Din formulele pentru eroarea relativă (1.3.2), dacă $x\approx x(1+\delta_x)$ şi $y\approx y(1+\delta_y)$, avem următoarele expresii pentru erorile relative ale operaţiilor în virgulă flotantă:

$$\delta_{xy}=\delta_x+\delta_y \tag{1.4.5}$$

$$\delta_{x/y}=\delta_x-\delta_y \tag{1.4.6}$$

$$\delta_{x+y}=\frac{x}{x+y}\delta_x+\frac{y}{x+y}\delta_y \tag{1.4.7}$$

Singura operaţie critică din punct de vedere al erorii este scăderea a două cantităţi apropiate $x \approx y$, caz în care $\delta_{x-y} \to \infty$. Acest fenomen se numeşte anulare şi este reprezentat grafic în figura 1.3. Aici $b, b', b''$ sunt cifre binare acceptabile, iar $g$-urile reprezintă cifre binare contaminate de eroare (gunoaie – garbage digits). De notat că, gunoi - gunoi = gunoi, dar mai important, normalizarea mută prima cifră contaminată de pe poziţia a 12-a pe poziţia a treia.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x | = | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | $b$ | $b$ | $g$ | $g$ | $g$ | $g$ | e |
| y | = | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | $b'$ | $b'$ | $g$ | $g$ | $g$ | $g$ | e |
| x-y | = | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $b''$ | $b''$ | $g$ | $g$ | $g$ | $g$ | e |
| | = | $b''$ | $b''$ | $g$ | $g$ | $g$ | $g$ | ? | ? | ? | ? | ? | ? | ? | ? | ? | e-9 |

Figura 1.3: Anularea

Anularea este de două tipuri: *benignă*, când se scad două cantităţi exacte şi *catastrofală*, când se scad două cantităţi deja rotunjite. Programatorul trebuie să fie conştient de posibilitatea apariţiei anulării şi să încerce să o evite. Expresiile în care apare anularea trebuie rescrise, iar o anulare catastrofală trebuie întotdeauna transformată în una benignă. Vom da în continuare câteva exemple de anulări catastrofale şi modul de transformare a lor .

**Exemplul 1.4.1.** Dacă $a \approx b$, atunci expresia $a^2 - b^2$ se transformă în $(a - b)(a + b)$. Forma iniţială este de preferat în cazul când $a \gg b$ sau $b \gg a$. ♢

**Exemplul 1.4.2.** Dacă anularea apare într-o expresie cu radicali, se amplifică cu conjugata:

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

♢

**Exemplul 1.4.3.** Diferenţa valorilor unei funcţii pentru argumente apropiate se transformă folosind formula lui Taylor:

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

♢

**Exemplul 1.4.4.** La ecuaţia de gradul al doilea $ax^2 + bx + c = 0$, anularea poate să apară dacă $b^2 \gg 4ac$. Formulele uzuale

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{1.4.8}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{1.4.9}$$

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

$$x_1 = \frac{2c}{-b - \sqrt{b^2 - 4ac}} \tag{1.4.10}$$

$$x_2 = \frac{2c}{-b + \sqrt{b^2 - 4ac}} \tag{1.4.11}$$

şi să utilizăm în primul caz formulele (1.4.10) şi (1.4.9), iar în al doilea caz (1.4.8) şi (1.4.11). ♢

# 1.5. Standardizarea reprezentării în virgulă flotantă

Există două standarde diferite pentru calculul în virgulă flotantă: IEEE 754 care prevede $\beta = 2$ şi IEEE 854 care permite $\beta = 2$ sau $\beta = 10$, dar lasă o mai mare libertate de reprezentare. Ne vom ocupa numai de primul standard.

Parametrii standardului se dau în tabela 1.1.

| | 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 1.1: Parametrii reprezentării flotante

Motivele pentru formatele extinse sunt:

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

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

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

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

În cazul IEEE 754, $D = 127$.

### 1.5.1. Cantităţi speciale

În standardul IEEE 754 există următoarele cantităţi speciale:

| Exponent | Semnificant | Ce reprezintă |
|---|---|---|
| $e = e_{min} - 1$ | $f = 0$ | $\pm 0$ |
| $e = e_{min} - 1$ | $f \neq 0$ | $0.f \times 2^{e_{min}}$ |
| $e_{min} \leq e \leq e_{max}$ | | $1.f \times 2^{e}$ |
| $e = e_{max} + 1$ | $f = 0$ | $\pm\infty$ |
| $e = e_{max} + 1$ | $f \neq 0$ | NaN |

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

**Infinit.** $1/0 = \infty$, $-1/0 = -\infty$. 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: $+0, -0$; relaţiile $+0 = -0$ şi $-0 < +\infty$ sunt adevărate. Avantaje: tratarea simplă a depăşirilor inferioare şi discontinuităţilor. Se face distincţie între $\log 0 = -\infty$ şi $\log x = \mathrm{NaN}$ pentru $x < 0$. Fără 0 cu semn nu s-ar putea face distincţie la logaritm între un număr negativ care dă depăşire superioară şi 0.

## 1.6. Condiţionarea unei probleme

Putem gândi o problemă ca o aplicaţie

$$f : \mathbb{R}^m \to \mathbb{R}^n, \quad y = f(x). \tag{1.6.1}$$

Ne interesează sensibilitatea aplicaţiei într-un punct dat $x$ la mici perturbaţii ale argumentului, adică cât de mare sau cât de mică este perturbaţia lui $y$ comparativ cu perturbaţia lui $x$. În particular, dorim să măsurăm gradul de sensibilitate printr-un singur număr, numărul de condiţionare al aplicaţiei $f$ în punctul $x$. Vom presupune că $f$ este calculată exact, cu precizie infinită. *Condiţionarea lui $f$ este deci o proprietate inerentă a funcţiei $f$ şi nu depinde de nici o consideraţie algoritmică legată de implementarea sa.*

Aceasta nu înseamnă că determinarea condiţionării unei probleme este nerelevantă pentru orice soluţie algoritmică a problemei. Din contră. Motivul este acela că soluţia calculată cu (1.6.1), $y^*$ (utilizând un algoritm specific şi aritmetica în virgulă flotantă) este (şi acest lucru se poate demonstra) soluţia unei probleme „apropiate“

$$y^* = f(x^*) \tag{1.6.2}$$

cu

$$x^* = x + \delta \tag{1.6.3}$$

şi, mai mult, distanţa $\|\delta\| = \|x^* - x\|$ poate fi estimată în termeni de precizie a maşinii. Deci, dacă ştim cât de tare sau cât de slab reacţionează aplicaţia la mici perturbaţii, cum ar fi $\delta$ în (1.6.3), putem spune ceva despre eroarea $y^* - y$ a soluţiei cauzată de această perturbaţie.

Se poate considera şi condiţionarea între aplicaţii mai generale, dar pentru implementări practice este suficient să ne limităm la cazul finit dimensional.

Fie

$$\begin{aligned} &x = [x_1, \ldots, x_m]^T \in \mathbb{R}^m, \quad y = [y_1, \ldots, y_n]^T \in \mathbb{R}^n, \\ &y_\nu = f_\nu(x_1, \ldots, x_m), \quad \nu = \overline{1, n}. \end{aligned}$$

$y_\nu$ va fi privit ca o funcţie de o singură variabilă $x_\mu$

$$\gamma_{\nu\mu} = (\mathrm{cond}_{\nu\mu}\, f)(x) = \left| \frac{x_\mu \frac{\partial f_\nu}{\partial x_\mu}}{f_\nu(x)} \right|. \tag{1.6.4}$$

Aceasta ne dă o matrice de numere de condiţionare

$$\Gamma(x) = \begin{pmatrix} \frac{x_1 \frac{\partial f_1}{\partial x_1}}{f_1(x)} & \cdots & \frac{x_m \frac{\partial f_1}{\partial x_m}}{f_1(x)} \\ \vdots & \ddots & \vdots \\ \frac{x_1 \frac{\partial f_n}{\partial x_1}}{f_n(x)} & \cdots & \frac{x_m \frac{\partial f_n}{\partial x_m}}{f_n(x)} \end{pmatrix} = [\gamma_{\nu\mu}(x)] \tag{1.6.5}$$

şi vom lua ca *număr de condiţionare*

$$(\mathrm{cond}\ f)(x) = \|\Gamma(x)\|. \tag{1.6.6}$$

**Altă abordare.** Considerăm norma $\|\cdot\|_\infty$

$$\Delta y_\nu \approx \sum_{\mu=1}^{m} \frac{\partial f_\nu}{\partial x_\mu} \Delta x_\mu \ (= f_\nu(x + \Delta x) - f_\nu(x))$$

$$|\Delta y_\nu| \le \sum_{\mu=1}^{n} \left| \frac{\partial f_\nu}{\partial x_\mu} \right| \Delta x_\mu \le \max_\mu |\Delta x_\mu| \sum_{\mu=1}^{m} \left| \frac{\partial f_\nu}{\partial x_\mu} \right| \le$$

$$\le \max_\mu |\Delta x_\mu| \max_\nu \sum_{\mu=1}^{m} \left| \frac{\partial f_\nu}{\partial x_\mu} \right|$$

Am obţinut

$$\|\Delta y\|_\infty \leq \|\Delta x\|_\infty \left\| \frac{\partial f}{\partial x} \right\|_\infty \tag{1.6.7}$$

unde

$$J(x) = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^n \times \mathbb{R}^m \tag{1.6.8}$$

este matricea jacobiană a lui $f$

$$\frac{\|\Delta y\|_\infty}{\|y\|_\infty} \leq \frac{\|x\|_\infty \left\| \frac{\partial f}{\partial x} \right\|_\infty}{\|f(x)\|_\infty} \cdot \frac{\|\Delta x\|}{\|x\|_\infty}. \tag{1.6.9}$$

Estimarea (1.6.9) este mai grosieră decât (1.6.4).

Dacă $m = n = 1$ în ambele abordări se obţine

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

pentru $x \neq 0$, $y \neq 0$.

Dacă $x = 0 \wedge y \neq 0$ se consideră eroarea absolută pentru $x$ şi 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 $y$ şi eroarea relativă pentru $x$, iar pentru $x = y = 0$

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

**Exemplul 1.6.1 (Sisteme de ecuaţii liniare algebrice).** Dându-se matricea $A \in \mathbb{R}^{m \times n}$ şi vectorul $b \in \mathbb{R}^n$ să se rezolve sistemul

$$Ax = b. \tag{1.6.10}$$

Aici datele de intrare sunt elementele lui $A$ şi $b$, iar rezultatul este vectorul $x$. Pentru a simplifica lucrurile să presupunem că $A$ este o matrice fixată care nu se schimbă şi că $b$ ar putea fi perturbat. Avem aplicaţia $f : \mathbb{R}^n \to \mathbb{R}^n$ dată de

$$x = f(b) := A^{-1}b,$$

care este liniară. Deci $\frac{\partial f}{\partial b} = A^{-1}$ şi utilizând (1.6.9) obţinem

$$\begin{aligned} (\text{cond } f)(b) &= \frac{\|b\|\|A^{-1}\|}{\|A^{-1}b\|} = \frac{\|Ax\|\|A^{-1}\|}{\|A^{-1}b\|}, \\ \max_{\substack{b \in \mathbb{R}^n \\ b \neq 0}} (\text{cond } f)(b) &= \max_{\substack{x \in \mathbb{R}^n \\ b \neq 0}} \frac{\|Ax\|}{\|x\|} \|A^{-1}\| = \|A\|\|A^{-1}\|. \end{aligned} \tag{1.6.11}$$

♢

Numărul $\|A\|\|A^{-1}\|$ se numeşte număr de condiţionare al matricei $A$ şi se notează $\mathrm{cond}\ A$.

$$\mathrm{cond}\ A = \|A\|\|A^{-1}\|.$$

Vom reveni asupra acestei probleme în capitolul 2, consacrat rezolvării sistemelor de ecuaţii algebrice liniare.

## 1.7. Condiţionarea unui algoritm

Fie problema

$$f : \mathbb{R}^m \to \mathbb{R}^n, \quad y = f(x). \tag{1.7.1}$$

Împreună cu $f$ se dă un algoritm $A$ care rezolvă problema. Adică dându-se un vector $x \in \mathbb{F}(\beta, p, e_{min}, e_{max}, \mathrm{denorm})$, algoritmul $A$ produce un vector $y_A$ (în aritmetica în virgulă flotantă), despre care se presupune că aproximează $y = f(x)$. Astfel avem o aplicaţie $f_A$ ce descrie modul în care problema $f$ este rezolvată de algoritmul $A$

$$f_A : \mathbb{F}^m(\dots) \to \mathbb{F}^n(\dots), \quad y_A = f_A(x).$$

Pentru a putea analiza $f_A$ facem următoarea ipoteză de bază

$$(\mathrm{IB}) \quad \forall\, x \in \mathbb{F}^m\ \exists\, x_A \in \mathbb{F}^m : \quad f_A(x) = f(x_A). \tag{1.7.2}$$

Adică, soluţia calculată corespunzând unei anumite intrări $x$ este soluţia exactă pentru o altă intrare, diferită de prima, $x_A$ (nu neapărat număr în virgulă flotantă, şi nici unic determinată) despre care sperăm că este apropiată de $x$. Cu cât găsim un $x_A$ mai apropiat de $x$, cu atât avem mai mare încredere în algoritm.

Definim *condiţionarea lui $A$ în $x$* comparând eroarea relativă la intrare cu $\mathrm{eps}$:

$$(\mathrm{cond}\ A)(x) = \inf_{x_A} \frac{\|x_A - x\|}{\|x\|} \Big/ \mathrm{eps}\,.$$

Justificare:

$$\delta_y = \frac{f_A(x) - f(x)}{f(x)} = \frac{(x_A - x)f'(\xi)}{f(x)} \approx \frac{x_A - x}{x} \cdot \frac{1}{\mathrm{eps}} \frac{xf'(x)}{f(x)}\,\mathrm{eps}$$

Infimumul se ia după toţi $x_A$ ce satisfac $y_A = f(x_A)$. În practică se poate lua orice valoare $x_A$ şi se obţine o margine superioară a numărului de condiţionare

$$(\mathrm{cond}\ A)(x) \le \frac{\frac{\|x_A - x\|}{\|x\|}}{\mathrm{eps}}. \tag{1.7.3}$$

# 1.8. Eroarea globală

Considerăm din nou problema:

$$f : \mathbb{R}^m \to \mathbb{R}^n, \quad y = f(x). \tag{1.8.1}$$

Aceasta este problema (matematică) idealizată, în care datele sunt numere reale, iar soluţia este soluţia matematică exactă. Când o rezolvăm în aritmetica în virgulă flotantă, cu precizia $\mathrm{eps}$, utilizând un algoritm $A$, rotunjim la început toate datele şi acestora nu le aplicăm $f$, ci $f_A$.

$$x^* = x \text{ rotunjit}, \quad \frac{\|x^* - x\|}{\|x\|} = \varepsilon, \quad y_A^* = f_A(x^*).$$

Aici $\varepsilon$ este eroarea de rotunjire. Ea poate proveni şi din alte surse (măsurători). Eroarea totală este

$$\frac{\|y_A^* - y\|}{\|y\|}.$$

Pe baza ipotezei (1.7.2, IB) alegând $x_A$ optimal avem

$$f_A(x^*) = f(x_A^*),$$
$$\frac{\|x_A^* - x^*\|}{\|x^*\|} = (\mathrm{cond}\ A)(x^*)\,\mathrm{eps}\,. \tag{1.8.2}$$

Fie $y^* = f(x^*)$. Avem utilizând inegalitatea triunghiului

$$\frac{\|y_A^* - y\|}{\|y\|} \le \frac{\|y_A^* - y^*\|}{\|y\|} + \frac{\|y^* - y\|}{\|y\|} \approx \frac{\|y_A^* - y^*\|}{\|y^*\|} + \frac{\|y^* - y\|}{\|y^*\|}$$

Am presupus că $\|y\| \approx \|y^*\|$. Din (1.8.2) rezultă pentru primul termen

$$\frac{\|y_A^* - y^*\|}{\|y^*\|} = \frac{\|f_A(x^*) - f(x^*)\|}{\|f(x^*)\|} = \frac{\|f(x_A^*) - f(x^*)\|}{\|f(x^*)\|} \le$$

$$\le (\mathrm{cond}\ f)(x^*)\frac{\|x_A^* - x\|}{\|x^*\|} = (\mathrm{cond}\ f)(x^*)(\mathrm{cond}\ A)(x^*)\,\mathrm{eps},$$

iar pentru al doilea

$$\frac{\|y^* - y\|}{\|y\|} = \frac{\|f(x^*) - f(x)\|}{\|f(x)\|} \le (\mathrm{cond}\ f)(x)\frac{\|x^* - x\|}{\|x\|} = (\mathrm{cond}\ f)(x)\varepsilon.$$

Presupunând că $(\mathrm{cond}\ f)(x^*) \approx (\mathrm{cond}\ f)(x)$ obţinem

$$\frac{\|y_A^* - y\|}{\|y\|} \le (\mathrm{cond}\ f)(x)[\varepsilon + (\mathrm{cond}\ A)(x^*)\,\mathrm{eps}]. \tag{1.8.3}$$

Interpretare: erorile în date şi $\mathrm{eps}$ contribuie împreună la eroarea totală. Ambele sunt amplificate de condiţionarea problemei, dar ultima este amplificată şi de condiţionarea algoritmului.

## 1.9. Probleme prost condiţionate şi probleme incorect puse

Dacă numărul de condiţionare al unei probleme este mare ($(\operatorname{cond} f)(x) \gg 1$), atunci chiar pentru erori (relative) mici trebuie să ne aşteptăm la erori foarte mari în datele de ieşire. Astfel de probleme se numesc *probleme prost condiţionate*. Nu este posibil să tragem o linie clară de demarcaţie între problemele bine condiţionate şi cele prost condiţionate. O categorizare a unei probleme în prost condiţionată sau bine condiţionată depinde de specificaţiile de precizie. Dacă dorim ca

$$\frac{\|y^* - y_A^*\|}{\|y\|} < \tau$$

şi în (1.8.3) $(\operatorname{cond} f)(x)\varepsilon \geq \tau$, atunci problema este sigur prost condiţionată.

Este important să se aleagă o limită rezonabilă pentru eroare, căci în caz contrar, chiar dacă creştem numărul de iteraţii, nu vom putea creşte precizia.

Dacă rezultatul unei probleme matematice depinde discontinuu de date ce variază continuu, atunci este imposibil de dat o soluţie numerică a problemei în vecinătatea discontinuităţii. În astfel de cazuri rezultatul poate fi perturbat substanţial, chiar dacă datele de intrare sunt precise şi utilizăm aritmetica de precizie multiplă. Astfel de probleme se numesc *probleme incorect puse*. O problemă incorect pusă poate să apară, de exemplu, dacă un rezultat întreg este calculat din date reale (adică date care variază continuu), de exemplu numărul de rădăcini reale ale unei funcţii sau rangul unei matrice.

**Exemplul 1.9.1 (Numărul de zerouri reale ale unui polinom).** Ecuaţia

$$P_3(x, c_0) = c_0 + x - 2x^2 + x^3$$

are una, două sau trei rădăcini reale, după cum $c_0$ este strict pozitiv, zero sau strict negativ (vezi figura 1.4). Deci, pentru valori ale lui $c_0$ apropiate de zero, numărul de zerouri reale ale lui $P_3$ este o problemă incorect pusă. ♢

## 1.10. Stabilitatea

### 1.10.1. Notaţii asimptotice

Această subsecţiune introduce notaţiile asimptotice de bază şi câteva abuzuri comune.

Pentru o funcţie dată $g(n)$ vom nota cu $\Theta(g(n))$ mulţimea de funcţii

$$\Theta(g(n)) = \{f(n) : \exists c_1, c_2, n_0 > 0 \;\; 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \; \forall n \leq n_0\}.$$

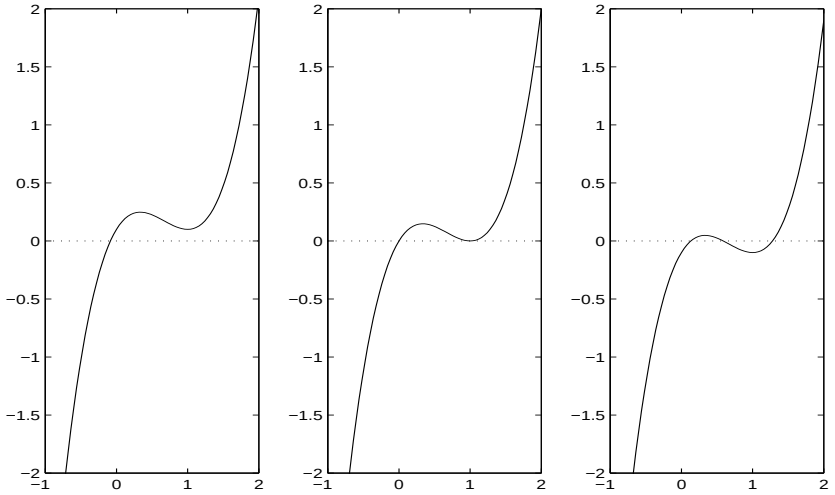


Figura 1.4: Problemă incorect pusă

Deşi $\Theta(g(n))$ este o mulţime; scriem $f(n) = \Theta(g(n))$ pentru a indica faptul că $f(n) \in \Theta(g(n))$. Acest abuz de egalitate pentru a nota apartenenţa la o mulţime poate părea la început confuz, dar vom vedea că are anumite avantaje. Vom spune că $g(n)$ este o *margine asimptotică strânsă (assymptotically tight bound)* pentru $f(n)$.

Definiţia mulţimii $\Theta(g(n))$ necesită ca fiecare membru al ei să fie *asimptotic nenegativ*, adică $f(n) \geq 0$ când $n$ este suficient de mare. În consecinţă $g(n)$ trebuie să fie şi ea asimptotic negativă, căci altfel $\Theta(g(n))$ este vidă. Din acest motiv vom presupune că fiecare funcţie utilizată în interiorul notaţiei $\Theta$ este asimptotic nenegativă. Această presupunere are loc şi pentru celelate notaţii asimptotice care vor fi definite în acest capitol.

Pentru o funcţie dată $g(n)$ vom nota cu $O(g(n))$ mulţimea de funcţii

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

Pentru a indica faptul că $f(n)$ este un membru al lui $O(g(n))$ scriem $f(n) = O(g(n))$. Observăm că $f(n) = \Theta(g(n)) \implies f(n) = O(g(n)$, deoarece notaţia $\Theta$ este mai tare decât notaţia $O$. Utilizând notaţiile din teoria mulţimilor avem $\Theta(g(n)) \subseteq O(g(n))$. Una dintre proprietăţile ciudate ale notaţiei este aceea că $n = O(n^2)$.

Pentru o funcţie dată $g(n)$ vom nota prin $\Omega(g(n))$ mulţimea de funcţii

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

Această notaţie furnizează o *margine asimptotică inferioară*. Din definiţiile notaţiilor asimptotice se obţine imediat:

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

Spunem că funcţiile $f$ şi $g : \mathbb{N} \longrightarrow \mathbb{R}$ sunt *asimptotic echivalente*, notaţie $\sim$ dacă

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

Extinderea notaţiilor asimptotice la mulţimea 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 subînţeleasă (de exemplu $t \to \infty$ sau $t \to 0$) avem

$$|f(t)| \leq Cg(t). \tag{1.10.1}$$

## 1.10.2. Precizie şi stabilitate

Vom considera o *problemă* ca fiind o aplicaţie $f : X \longrightarrow Y$, unde $X$ şi $Y$ sunt spaţii liniare normate (pentru scopurile noastre ne vom limita la cazul finit dimensional). Ne va interesa comportarea problemei într-un punct particular $x \in X$ (comportarea poate diferi de la un punct la altul). Combinaţia unei probleme $f$ cu nişte date prescrise $x$ se va numi *instanţă a problemei*, dar se obişnuieşte să se utilizeze termenul de *problemă* pentru ambele noţiuni.

Numerele complexe sunt reprezentate printr-o pereche de numere în virgulă flotantă şi operaţiile elementare se realizează pe această reprezentare. Rezultatul este acela că axioma (1.4.4) este valabilă şi pentru numere complexe, exceptând faptul că la operaţiile $\otimes$ şi $\oslash$ $\mathrm{eps}$ trebuie mărit cu un factor de $2^{3/2}$ şi respectiv $2^{5/2}$.

Un algoritm va fi privit ca o aplicaţie $f_A : X \longrightarrow Y$, unde $X$ şi $Y$ sunt ca mai sus. Fie $f$ o problemă, un calculator al cărui sistem de numere în virgulă flotantă satisface (1.4.4)), dar nu neapărat (1.4.3), un algoritm $f_A$ pentru $f$ şi o implementare a acestui algoritm sub formă de program pe calculator, $A$, toate fixate. Dându-se o dată $x \in X$, o vom rotunji la un număr în virgulă flotantă şi apoi o vom furniza programului. Rezultatul este o colecţie de numere în virgulă flotantă care aparţin spaţiului $Y$ (deoarece algoritmul a fost conceput să rezolve $f$). Vom numi acest rezultat calculat $f_A(x)$.

Exceptând cazul trivial, $f_A$ nu poate fi continuă. Vom spune că un algoritm $f_A$ pentru problema $f$ este *precis*, dacă pentru orice $x \in X$, eroarea sa relativă verifică

$$\frac{\|f_A(x) - f(x)\|}{\|f(x)\|} = O(\mathrm{eps}). \tag{1.10.2}$$

Dacă problema $f$ este prost condiţionată, obiectivul preciziei, aşa cum este definit de (1.10.2) este nerezonabil de ambiţios. Erorile de rotunjire în datele de intrare sunt inevitabile pe calculatoare numerice şi chiar dacă toate calculele următoare se realizează perfect, această perturbaţie ne conduce la o modificare semnificativă a rezultatului. În loc să urmărim precizia în toate cazurile este mai rezonabil să urmărim stabilitatea.

Spunem că algoritmul $f_A$ pentru problema $f$ este *stabil* dacă pentru orice $x \in X$ există un $\widetilde{x}$ cu

$$\frac{\|\widetilde{x} - x\|}{\|x\|} = O(\text{eps}) \tag{1.10.3}$$

astfel încât

$$\frac{\|f_A(x) - f(\widetilde{x})\|}{\|f(\widetilde{x})\|} = O(\text{eps}). \tag{1.10.4}$$

În cuvinte, *un algoritm stabil ne dă un răspuns aproape corect la o problemă aproape exactă.*

Mulţi dintre algoritmii din algebra liniară numerică satisfac o condiţie care este mai puternică şi mai simplă decât stabilitatea. Spunem că un algoritm $f_A$ pentru problema $f$ este *regresiv stabil* (backward stable) dacă

$$\forall x \in X \ \exists \widetilde{x} \text{ cu } \frac{\|\widetilde{x} - x\|}{\|x\|} = O(\text{eps}) \text{ astfel încât } f_A(x) = f(\widetilde{x}). \tag{1.10.5}$$

Aceasta este o întărire a definiţiei stabilităţii în sensul că $O(\text{eps})$ din (1.10.4) a fost înlocuit cu 0. În cuvinte, *un algoritm este regresiv stabil dacă dă răspunsul corect la o problemă aproape exactă.*

**Observaţia 1.10.1.** Semnificaţia notaţiei

$$||\text{cantitate calculată}|| = O(\text{eps}) \tag{1.10.6}$$

este următoarea:

- $||$cantitate calculată$||$ reprezintă norma unui număr sau a unei colecţii de numere determinate de algoritmul $f_A$ pentru o problemă $f$, depinzând atât de datele de intrare $x \in X$ ale lui $f$ cât şi de $\text{eps}$. Un exemplu este eroarea relativă.
- Procesul implicit de trecere la limită este $\text{eps} \to 0$ (adică $\text{eps}$ corespunde lui $t$ din (1.10.1)).
- $O$ se aplică uniform tuturor datelor $x \in X$. La formularea rezultatelor de stabilitate această uniformitate va fi implicită.
- Pentru orice aritmetică pe calculator particulară $\text{eps}$ este o cantitate fixată. Când vorbim despre limita $\text{eps} \to 0$, considerăm o idealizare a unui calculator sau a unei familii de calculatoare. Ecuaţia (1.10.6) înseamnă că dacă rulăm algoritmul în cauză pe calculatoare ce satisfac (1.4.3) şi (1.4.4) pentru un şir de $\text{eps}$-uri descrescător ce tinde către zero, se garantează că $||$cantitate calculată$||$ descreşte proporţional cu $\text{eps}$ sau mai repede. Acestor calculatoare ideale li se cere să satisfacă doar (1.4.3) şi (1.4.4) şi nimic altceva.

- Constanta implicită din notaţia $O$ poate depinde şi de dimensiunile argumentelor (de exemplu pentru rezolvarea unui sistem $Ax = b$ de dimensiunile lui $A$ şi $b$). În general, în probleme practice, creşterea erorii datorată dimensiunii este lentă, dar pot exista situaţii în care să apară factori cum ar fi $2^m$, care fac astfel de margini inutile în practică. $\diamondsuit$

Datorită echivalenţei normelor pe spaţii finit dimensionale, pentru problemele $f$ şi algoritmii $f_A$ definite pe astfel de spaţii, proprietăţile de precizie, stabilitate şi stabilitate regresivă au loc sau nu independent de alegerea normelor.

### 1.10.3. Analiza regresivă a erorilor

Stabilitatea regresivă implică precizia în sens relativ.

**Teorema 1.10.2** *Presupunem că se aplică un algoritm regresiv stabil $f_A$ unei probleme $f : X \longrightarrow Y$ cu numărul de condiţionare $(\operatorname{cond} f)(x)$ pe un calculator ce satisface* (1.4.3) *şi* (1.4.4)*. Atunci eroarea relativă satisface*

$$\frac{\|f_A(x) - f(x)\|}{\|f(x)\|} = O\left((\operatorname{cond} f)(x)\,\mathrm{eps}\right). \tag{1.10.7}$$

*Demonstraţie.* Din definiţia (1.10.5) a stabilităţii regresive, există un $\widetilde{x} \in X$ ce satisface

$$\frac{\|\widetilde{x} - x\|}{\|x\|} = O(\mathrm{eps}),$$

astfel încât $f_A(x) = f(\widetilde{x})$. Din definiţia (1.6.5) şi (1.6.6) a lui $(\operatorname{cond} f)(x)$ aceasta implică

$$\frac{\|f_A(x) - f(x)\|}{\|f(x)\|} \leq \left((\operatorname{cond} f)(x) + o(1)\right) \frac{\|\widetilde{x} - x\|}{\|x\|}, \tag{1.10.8}$$

unde $o(1)$ desemnează o cantitate ce converge către zero când $\mathrm{eps} \to 0$. Combinând aceste delimitări se obţine (1.10.7). $\square$

Procesul urmat în demonstraţia teoremei 1.10.2 este cunoscut sub numele de *analiza regresivă a erorilor* (backward error analysis). Se obţine o estimaţie a preciziei în doi paşi. Primul pas este investigarea condiţionării problemei. Celălalt este investigarea stabilităţii propriu-zise a algoritmului. Conform teoremei 1.10.2, dacă algoritmul este regresiv stabil, atunci precizia finală reflectă acel număr de condiţionare.

În afară de analiza regresivă a erorilor, există şi o analiză directă sau *progresivă*. Aici se estimează erorile de rotunjire la fiecare pas şi apoi modul cum se compun şi în final un total (secţiunea 1.3).

Experienţa a arătat că pentru cei mai mulţi algoritmi ai algebrei liniare numerice analiza progresivă a erorilor este mai greu de realizat decât cea regresivă. Cei mai buni algoritmi pentru cele mai multe probleme ale algebrei liniare numerice nu fac altceva mai bun decât să calculeze soluţia exactă pentru nişte date uşor perturbate. Analiza regresivă este o metodă de raţionament bine adaptată acestei situaţii.

CAPITOLUL 2

# Rezolvarea numerică a sistemelor de ecuaţii algebrice liniare

## Cuprins



Există două clase de metode de rezolvare a sistemelor algebrice liniare: metode *directe* sau *exacte*, care furnizează soluţia într-un număr finit de paşi, în ipoteza că toate calculele se fac exact (Cramer, eliminarea gaussiană, Cholesky) şi metode *iterative*, care aproximează soluţia generând un şir care converge către aceasta (Jacobi, Gauss-Seidel, SOR).

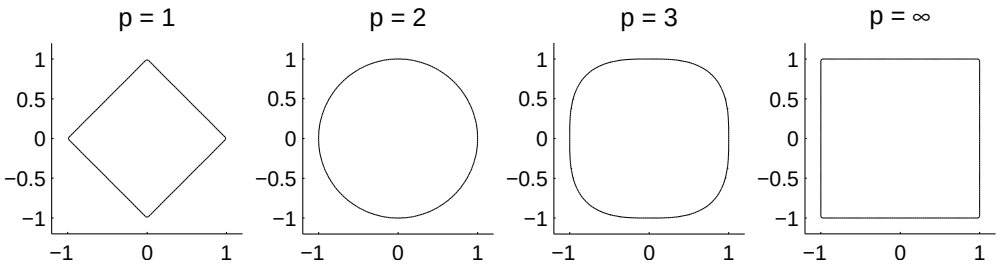



Figura 2.1: Sfera unitate pentru patru $p$-norme

## 2.1. Elemente de Analiză matricială

$p$-norma unui vector $x \in \mathbb{K}^n$ se defineşte prin

$$\|x\|_p = \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p} \qquad 1 \leq p < \infty.$$

Pentru $p = \infty$ norma este definită prin

$$\|x\|_\infty = \max_{i=\overline{1,n}} |x_i|.$$

Norma $\|\cdot\|_2$ se numeşte *normă euclidiană*, $\|\cdot\|_1$ se numeşte *normă Minkowski*, iar $\|\cdot\|_\infty$ se numeşte *normă Cebîşev*. În figura 2.1 apar reprezentate sferele unitate în diverse $p$-norme.

Fie $A \in \mathbb{K}^{n\times n}$. Polinomul $p(\lambda) = \det(A - \lambda I)$ se numeşte *polinomul caracteristic* al lui $A$. Rădăcinile lui $p$ se numesc *valori proprii* ale lui $A$, iar dacă $\lambda$ este o valoare proprie a lui $A$ vectorul $x \neq 0$ cu proprietatea că $(A - \lambda I)x = 0$ se numeşte *vector propriu* al lui $A$ corespunzător valorii proprii $\lambda$.

Valoarea $\rho(A) = \max\{|\lambda| \ \lambda \text{ valoare proprie a lui } A\}$ se numeşte *rază spectrală* a matricei $A$. Vom nota cu $A^T$ transpusa lui $A$ şi cu $A^*$ transpusa conjugată a lui A.

**Definiţia 2.1.1** *O matrice $A$ se numeşte:*

1. normală, *dacă* $AA^* = A^*A$;
2. unitară, *dacă* $AA^* = A^*A = I$;
3. hermitiană, *dacă* $A = A^*$;
4. ortogonală, *dacă* $AA^T = A^TA = I$, $A$ *reală;*
5. simetrică, *dacă* $A = A^T$, $A$ *reală.*

**Definiţia 2.1.2** *O* normă matricială *este o aplicaţie* $\|\cdot\| : \mathbb{K}^{m\times n} \to \mathbb{R}$ *care pentru orice* $A, B \in \mathbb{K}^{m\times n}$ *şi* $\alpha \in \mathbb{K}$ *verifică următoarele relaţii*

*(NM1)* $\|A\| \geq 0,\ \|A\| = 0 \Leftrightarrow A = O_n$;

*(NM2)* $\|\alpha A\| = |\alpha|\|A\|$;

*(NM3)* $\|A + B\| \leq \|A\| + \|B\|$;

*(NM4)* $\|AB\| \leq \|A\|\|B\|$.

Primele trei proprietăţi ne spun că $\|\cdot\|$ este o normă pe $\mathbb{K}^{m\times n}$, care este şi spaţiu vectorial de dimensiune $mn$, iar (NM4) este specifică normelor matriciale. Un mijloc simplu de construire a normelor matriciale este următorul: fiind dată o normă vectorială $\|\cdot\|$ pe $\mathbb{C}^n$, aplicaţia $\|\cdot\| : \mathbb{C}^{n\times n} \to \mathbb{R}$

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

este o normă matricială numită *normă matricială subordonată* (normei vectoriale date) sau *normă naturală* (indusă de norma dată).

**Observaţia 2.1.3.**

1. Aceste norme matriciale subordonate sunt un caz particular al normei unei aplicaţii liniare $A : \mathbb{K}^m \to \mathbb{K}^n$.
2. O normă subordonată verifică $\|I\| = 1$.
3. Dacă matricea $A$ este reală, marginea superioară a raportului $\|Av\|/\|v\|$ este atinsă pentru vectori reali (vezi teorema următoare). ♢

Să calculăm acum normele subordonate normelor vectoriale $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$.

**Teorema 2.1.4** *Fie* $A \in \mathbb{K}^{n\times n}(\mathbb{C})$. *Atunci*

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

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

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

*Norma* $\|\cdot\|_2$ *este invariantă la transformările unitare,*

$$UU^* = I \Rightarrow \|A\|_2 = \|AU\|_2 = \|UA\|_2 = \|U^*AU\|_2.$$

*Altfel spus, dacă* $A$ *este normală, atunci*

$$AA^* = A^*A \Rightarrow \|A\|_2 = \rho(A).$$

*Demonstraţie.* Pentru orice vector $v$ avem

$$\|Av\|_1 = \sum_i \left| \sum_j a_{ij} v_j \right| \leq \sum_j |v_j| \sum_i |a_{ij}| \leq$$
$$\leq \left( \max_j \sum_i |a_{ij}| \right) \|v\|_1.$$

Pentru a arăta că $\max_j \sum_i |a_{ij}|$ este efectiv cel mai mic număr $\alpha$ pentru care are loc $\|Av\|_1 \leq \alpha \|v\|_1, \ \forall\, v \in \mathbb{C}^n$, să construim un vector $u$ (care depinde de $A$) astfel încât

$$\|Au\|_1 = \left\{ \max_j \sum_i |a_{ij}| \right\} \|u\|_1.$$

Dacă $j_0$ este un indice ce verifică

$$\max_j \sum_i |a_{ij}| = \sum_i |a_{ij_0}|,$$

atunci vectorul $u$ are componentele $u_i = 0$ pentru $i \neq j_0$, $u_{j_0} = 1$.

La fel

$$\|Av\|_\infty = \max_i \left| \sum_j a_{ij} v_j \right| \leq \left( \max_i \sum_j |a_{ij}| \right) \|v\|_\infty.$$

Fie $i_0$ un indice ce verifică

$$\max_i \sum_j |a_{ij}| = \sum_j |a_{i_0 j}|.$$

Vectorul $u$ de componente $u_j = \dfrac{\overline{a_{i_0 j}}}{|a_{i_0 j}|}$ dacă $a_{i_0 j} \neq 0$, $u_j = 1$ dacă $a_{i_0 j} = 0$ verifică

$$\|Au\|_\infty = \left\{ \max_i \sum_j |a_{ij}| \right\} \|u\|_\infty.$$

Deoarece $AA^*$ este hermitiană, există o descompunere proprie $AA^* = Q\Lambda Q^*$, unde $Q$ este o matrice unitară (ale cărei coloane sunt vectori proprii) şi $\Lambda$ este matricea diagonală a valorilor proprii, care trebuie să fie toate reale. Dacă ar exista o valoare proprie negativă şi $q$ ar fi vectorul propriu corespunzător, am avea $0 \leq \|Aq\|_2^2 = q^T A^T A q =$

$q^T\lambda q = \lambda\|q\|_2^2$. Deci

$$\begin{aligned}\|A\|_2 &= \max_{x\neq 0}\frac{\|Ax\|_2}{\|x\|_2} = \max_{x\neq 0}\frac{(x^*A^*Ax)^{1/2}}{\|x\|_2} = \max_{x\neq 0}\frac{(x^*Q\Lambda Q^*x)^{1/2}}{\|x\|_2}\\ &= \max_{x\neq 0}\frac{((Q^*x)^*\Lambda Q^*x)^{1/2}}{\|Q^*x\|_2} = \max_{y\neq 0}\frac{(y^*\Lambda y)^{1/2}}{\|y\|_2} = \max_{y\neq 0}\sqrt{\frac{\sum\lambda_i y_i^2}{\sum y_i^2}}\\ &\leq \max_{y\neq 0}\sqrt{\lambda_{\max}}\sqrt{\frac{\sum y_i^2}{\sum y_i^2}};\end{aligned}$$

egalitatea are loc dacă $y$ este o coloană convenabil aleasă a matricei identitate.

Să arătăm că $\rho(A^*A) = \rho(AA^*)$. Dacă $\rho(A^*A) > 0$ există $p$ astfel încât $p \neq 0$, $A^*Ap = \rho(A^*A)p$ şi $Ap \neq 0$ ($\rho(A^*A) > 0$). Cum $Ap \neq 0$ şi $AA^*(Ap) = \rho(A^*A)Ap$ rezultă că $0 < \rho(A^*A) \leq \rho(AA^*)$ şi deci $\rho(AA^*) = \rho(A^*A)$ căci $(A^*)^* = A$. Dacă $\rho(A^*A) = 0$, avem $\rho(AA^*) = 0$. Deci în toate cazurile $\|A\|_2^2 = \rho(A^*A) = \rho(AA^*) = \|A^*\|_2^2$.

Invarianţa normei $\|\cdot\|_2$ la transformări unitare nu este decât traducerea egalităţilor

$$\rho(A^*A) = \rho(U^*A^*AU) = \rho(A^*U^*UA) = \rho(U^*A^*UU^*AU).$$

În fine, dacă $A$ este normală, există o matrice $U$ astfel încât $U^*AU = diag(\lambda_i(A)) \stackrel{def}{=} \Lambda$. În aceste condiţii

$$A^*A = (U\Lambda U^*)^*U\Lambda U = UD^*\Lambda U^*,$$

ceea ce ne arată că

$$\rho(A^*A) = \rho(\Lambda^*\Lambda) = \max_i|\lambda_i(A)|^2 = (\rho(A))^2.$$

□

**Observaţia 2.1.5.** 1) Dacă $A$ este hermitiană sau simetrică (deci normală),

$$\|A\|_2 = \rho(A).$$

2) Dacă $A$ este unitară sau ortogonală (deci normală),

$$\|A\|_2 = \sqrt{\rho(A^*A)} = \sqrt{\rho(I)} = 1.$$

3) Teorema 2.1.4 ne spune că matricele normale şi norma $\|\cdot\|_2$ verifică

$$\|A\|_2 = \rho(A).$$

(4) Norma $||.||_\infty$ se mai numeşte norma Cebâşev sau m-normă, norma $||.||_1$ norma lui Minkowski sau l-normă, iar norma $||.||_2$ normă euclidiană. $\diamondsuit$

**Teorema 2.1.6** *(1) Fie $A$ o matrice pătratică oarecare şi $\|\cdot\|$ o normă matricială oarecare (subordonată sau nu). Atunci*

$$\rho(A) \leq \|A\|. \tag{2.1.1}$$

*(2) Fiind dată o matrice $A$ şi un număr $\varepsilon > 0$, există cel puţin o normă matricială subordonată astfel încât*

$$\|A\| \leq \rho(A) + \varepsilon. \tag{2.1.2}$$

*Demonstraţie.* (1) Fie $p$ un vector ce verifică $p \neq 0$, $Ap = \lambda p$, $|\lambda| = \rho(A)$ şi $q$ un vector astfel încât $pq^T \neq 0$. Deoarece

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

rezultă (2.1.1).

(2) Fie $A$ o matrice dată. Există o matrice inversabilă $U$ astfel încât $U^{-1}AU$ este triunghiulară superior (de fapt $U$ este unitară). De exemplu

$$U^{-1}AU = \begin{pmatrix} \lambda_1 & t_{12} & t_{13} & \dots & t_{1,n} \\ & \lambda_2 & t_{23} & \dots & t_{2,n} \\ & & \ddots & & \vdots \\ & & & \lambda_{n-1} & t_{n-1,n} \\ & & & & \lambda_n \end{pmatrix}.$$

scalarii $\lambda_i$ fiind valorile proprii ale matricei $A$. (Pentru demonstraţie a se vedea teorema 7.2.6 din capitolul refcapvalpr.) Fiecărui scalar $\delta \neq 0$ îi asociem matricea

$$D_\delta = diag(1, \delta, \delta^2, \dots, \delta^{n-1}),$$

astfel ca

$$(UD_\delta)^{-1}A(UD_\delta) = \begin{pmatrix} \lambda_1 & \delta t_{12} & \delta^2 t_{13} & \dots & \delta^{n-1} t_{1n} \\ & \lambda_2 & \delta t_{23} & \dots & \delta^{n-2} t_{2n} \\ & & \ddots & & \vdots \\ & & & \lambda_{n-1} & \delta t_{n-1n} \\ & & & & \lambda_n \end{pmatrix}.$$

Fiind dat $\varepsilon > 0$, fixăm $\delta$ astfel ca

$$\sum_{j=i+1}^{n} |\delta^{j-i} t_{ij}| \leq \varepsilon, \quad 1 \leq i \leq n-1.$$

Atunci aplicaţia

$$\|\cdot\| : B \in \mathbb{K}^{n\times n} \to \|B\| = \|(UD_\delta)^{-1}B(UD_\delta)\|_\infty,$$

care depinde de $A$ şi de $\varepsilon$ răspunde problemei. Într-adevăr, avem pe de o parte

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

conform alegerii lui $\delta$ şi definiţiei normei $\|\cdot\|_\infty$ ($\|c_{ij}\|_\infty = \max_i \sum_j |c_{ij}|$) şi pe de altă parte ea este o normă matricială subordonată normei vectoriale

$$v \in \mathbb{K}^n \to \|(UD_\delta)^{-1}v\|_\infty.$$

□

Un exemplu important de normă nesubordonată este dat de teorema următoare.

**Teorema 2.1.7** *Aplicaţia $\|\cdot\|_E : \mathbb{K}^{n\times n} \to \mathbb{R}$ definită prin*

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

*este o normă matricială nesubordonată, invariantă la transformările unitare*

$$UU^* = I \Rightarrow \|A\|_E = \|AU\|_E = \|UA\|_E = \|U^*AU\|_E$$

*şi care verifică*

$$\|A\|_2 \leq \|A\|_E \leq \sqrt{n}\|A\|_2,\ \forall\, A \in \mathbb{K}^{n\times n}.$$

*Demonstraţie.* $\|\cdot\|_E$ este norma euclidiană pe $\mathbb{K}^{n\times n}$ de dimensiune $n^2$. Proprietatea (NM4) se demonstrează cu inegalitatea Cauchy-Buniakowski-Schwarz

$$\|AB\|_E^2 = \sum_{i,j}\left|\sum_k a_{jk}b_{kj}\right| \leq \sum_{i,j}\left\{\sum_k |a_{ik}|^2\right\}\left\{\sum_l |b_{lj}|^2\right\} =$$

$$= \left\{\sum_{i,k} |a_{ik}|^2\right\}\left\{\sum_{j,l} |b_{lj}|^2\right\} = \|A\|_E^2\|B\|_E^2.$$

Această normă nu este subordonată, deoarece $\|I\|_E = \sqrt{n}$. Dacă $U$ este o matrice unitară

$$\|A\|_E^2 = tr(A^*A) = tr(U^*A^*AU) =$$

$$= \|AU\|_E^2 = tr(U^*A^*UA) = \|UA\|_E.$$

În fine inegalităţile din enunţ rezultă din inegalităţile

$$\rho(A^*A) \leq tr(A^*A) \leq n\rho(A^*A).$$

□

Norma $||.||_E$ se numeşte *normă Frobenius* .

**Teorema 2.1.8** *Fie $B$ o matrice pătratică. Următoarele afirmaţii sunt echivalente:*

*(1)* $\lim_{k\to\infty} B^k = 0$;

*(2)* $\lim_{k\to\infty} B^k v = 0,\ \forall\, v \in \mathbb{K}^n$;

*(3)* $\rho(B) < 1$;

*(4) Există o normă matricială subordonată astfel încât* $\|B\| < 1$.

*Demonstraţie.* (1) $\Rightarrow$ (2)

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

(2) $\Rightarrow$ (3) Dacă $\rho(B) \geq 1$ putem găsi $p$ astfel încât $p \neq 0$, $Bp = \lambda p$, $|\lambda| \geq 1$. Atunci şirul de vectori $(B^k p)_{k\in\mathbb{N}}$ ar putea să nu conveargă către 0.
(3) $\Rightarrow$ (4) $\rho(B) < 1 \Rightarrow \exists \|\cdot\|$ astfel încât $\|B\| \leq \rho(B) + \varepsilon,\ \forall\, \varepsilon > 0$ deci $\|B\| < 1$.
(4) $\Rightarrow$ (1) Este suficient să aplicăm inegalitatea $\|B^k\| \leq \|B\|^k$. $\square$

## 2.2. Condiţionarea unui sistem liniar

Fie sistemul (exemplul este datorat lui Wilson)

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

cu soluţia $(1,1,1,1)^T$ şi considerăm sistemul perturbat, în care membrul drept este foarte puţin modificat, matricea rămânând neschimbată

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

cu soluţia $(9.2, -12.6, 4.5, -1.1)^T$. Altfel spus, o eroare de 1/200 în date (aici componentele din membrul drept) atrage o eroare relativă de 10/1 asupra rezultatului, deci o mărire a erorii relative de ordin 2000!

Considerăm acum sistemul având de această dată matricea perturbată

$$\begin{pmatrix} 10 & 7 & 8.1 & 7.2 \\ 7.08 & 5.04 & 6 & 5 \\ 8 & 5.98 & 9.89 & 9 \\ 6.99 & 4.99 & 9 & 9.98 \end{pmatrix} \begin{pmatrix} x_1 + \Delta x_1 \\ x_2 + \Delta x_2 \\ x_3 + \Delta x_4 \\ x_4 + \Delta x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

cu soluţia $(-81, 137, -34, 22)^T$. Din nou, o variaţie mică în datele de intrare (aici, elementele matricei) modifică complet rezultatul (soluţia sistemului liniar). Matricea are un aspect „bun“, ea este simetrică, determinantul ei este 1, iar inversa ei este

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

care este de asemenea simpatică.

Să analizăm aceste fenomene. În primul caz se dă o matrice *inversabilă* $A$ şi se compară soluţiile *exacte* $x$ şi $x + \delta x$ ale sistemelor

$$\begin{aligned} Ax &= b \\ A(x + \delta x) &= b + \delta b. \end{aligned}$$

Fie $||\cdot||$ o normă vectorială oarecare şi $||\cdot||$ norma matricială subordonată. Din egalităţile $\delta x = A^{-1}\delta b$ şi $b = Ax$ se deduce

$$||\Delta x|| \leq ||A^{-1}||||\delta b||, \quad ||b|| \leq ||A||||x||.$$

Eroarea relativă a rezultatului $\frac{||\delta x||}{||x||}$ este majorată în funcţie de eroarea relativă a datelor prin

$$\frac{||\delta x||}{||x||} \leq ||A||||A^{-1}||\frac{||\delta b||}{||b||}.$$

În al doilea caz, când matricea variază, avem de comparat soluţiile exacte ale sistemelor

$$\begin{aligned} Ax &= b \\ (A + \Delta A)(x + \Delta x) &= b. \end{aligned}$$

Din egalitatea $\Delta x = -A^{-1}\Delta A(x + \Delta x)$ se deduce

$$||\Delta x|| \leq ||A^{-1}||||\Delta A||||x + \Delta x||$$

care se mai poate scrie

$$\frac{||\Delta x||}{||x+\Delta x||} \leq ||A||||A^{-1}||\frac{||\Delta A||}{||A||}.$$

Dacă $A$ este nesingulară, numărul

$$cond(A) = ||A||||A^{-1}|| \tag{2.2.1}$$

se numeşte *număr de condiţionare* al matricei $A$.

Se poate da o estimare a numărului de condiţionare în care să intervină simultan şi perturbaţiile lui $A$ şi ale lui $b$. Considerăm sistemul parametrizat, cu parametrul $t$

$$(A + t\Delta A)x(t) = b + t\Delta b, \qquad x(0) = x.$$

Matricea $A$ fiind nesingulară, funcţia $x$ este diferenţiabilă în $t = 0$:

$$\dot{x}(0) = A^{-1}(\Delta b - \Delta A x).$$

Dezvoltarea Taylor a lui $x(t)$ este dată de

$$x(t) = x + t\dot{x}(0) + O(t^2).$$

Rezultă că eroarea absolută poate fi estimată utilizând

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

şi datorită lui $||b|| \leq ||A||||x||$ obţinem pentru eroarea relativă

$$\begin{aligned} \frac{\|\Delta x(t)\|}{\|x\|} &\leq |t|\,\left\|A^{-1}\right\|\left(\frac{\|\Delta b\|}{\|x\|} + \|\Delta A\|\right) + O(t^2) \\ &\leq \|A\|\,\left\|A^{-1}\right\|\,|t|\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}\right) + O(t^2). \end{aligned} \tag{2.2.2}$$

Introducând notaţiile

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

pentru erorile relative în $A$ şi $b$, estimarea erorii (2.2.2) se scrie sub forma

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

**Exemplul 2.2.1 (Exemple de matrice prost condiţionate).** Matricea lui Hilbert [1] $H_n = (h_{ij})$, cu $h_{ij} = \frac{1}{i+j-1}$, $i, j = \overline{1,n}$ are ordinul de mărime al numărului de condiţionare relativ la norma euclidiană (dat de Szegö[2])

$$cond_2(H_n) \sim \frac{(\sqrt{2}+1)^{4n+4}}{2^{14/4}\sqrt{\pi n}}.$$

Pentru diverse valori ale lui $n$ se obţine

| $n$ | $cond_2(H_n)$ |
|---|---|
| 10 | $1.6 \cdot 10^{13}$ |
| 20 | $2.45 \cdot 10^{28}$ |
| 40 | $7.65 \cdot 10^{58}$ |

Un alt exemplu este matricea Vandermonde. Dacă elementele sunt echidistante în [-1,1], atunci

$$cond_\infty(V_n) \sim \frac{1}{\pi} e^{-\frac{\pi}{4}} e^{n(\frac{\pi}{4}+\frac{1}{2}\ln 2)},$$

iar pentru $t_i = \frac{1}{i},\ i = \overline{1,n}$ avem

$$cond_\infty(V_n) > n^{n+1}. \qquad \diamondsuit$$

---

[1] 

David Hilbert (1862-1943) a fost cel mai important reprezentant al şcolii matematice din Göttingen. Contribuţiile sale fundamentale în aproape toate domeniile matematicii — algebră, teoria numerelor, geometrie, ecuaţii integrale, calcul variaţional şi fundamentele matematicii — şi în particular cele 23 de probleme celebre pe care le-a propus în 1900 la un congres internaţional al matematicienilor de la Paris, au dat un nou impuls şi o nouă direcţie matematicii din secolul al XX-lea.

[2] 

Gabor Szegő (1895-1985) Unul dintre cei mai importanţi matematicieni maghiari din secolul al XX-lea. Contribuţii importante în domeniul problemelor extremale şi matricelor Toeplitz.

# 2.3. Metode exacte

## 2.3.1. Metoda eliminării a lui Gauss

Să considerăm sistemul liniar de $n$ ecuaţii cu $n$ necunoscute

$$Ax = b, \tag{2.3.1}$$

unde $A \in \mathbb{K}^{n\times n}$, $b \in \mathbb{K}^{n\times 1}$ sunt date, iar $x \in \mathbb{K}^{n\times 1}$ este necunoscuta, sau scris detaliat

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 & (E_1) \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 & (E_2) \\ \vdots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n & (E_n) \end{cases} \tag{2.3.2}$$

Metoda eliminării a lui Gauss[3] are două etape:

e1) transformarea sistemului dat într-unul echivalent, triunghiular;

e2) rezolvarea sistemului triunghiular prin substituţie inversă.

La rezolvarea sistemului (2.3.1) sau (2.3.2) sunt permise următoarele operaţii:

1. Ecuaţia $E_i$ poate fi înmulţită cu $\lambda \in \mathbb{K}^*$. Această operaţie se va nota cu $(\lambda E_i) \to (E_i)$.
2. Ecuaţia $E_j$ poate fi înmulţită cu $\lambda \in \mathbb{K}^*$ şi adunată la ecuaţia $E_i$, iar rezultatul utilizat în locul lui $E_i$, notaţie $(E_i + \lambda E_j) \to (E_i)$.

[3] Johann Carl Friedrich Gauss (1777-1855) a fost unul dintre cei mai mari matematicieni ai secolului al nouăsprezecelea şi probabil al tuturor timpurilor. A trăit aproape toată viaţa în Göttingen, unde a fost directorul observatorului astronomic 40 de ani. În timpul studenţiei la Göttingen, Gauss a descoperit că poligonul cu 17 laturi poate fi construit cu rigla şi compasul, rezolvând astfel o problemă deschisă a antichităţii. În dizertaţia sa a dat prima demonstraţie a teoremei fundamentale a algebrei. A avut contribuţii fundamentale în teoria numerelor, geometrie diferenţială şi neeuclidiană, funcţii eliptice şi hipergeometrice, mecanică cerească şi geodezie şi diverse ramuri ale fizicii, în special magnetism şi optică. Eforturile sale de calcul în mecanica cerească şi geodezie, bazate pe principiul celor mai mici pătrate, au necesitat rezolvarea manuală a unor sisteme de ecuaţii liniare mari, la care a utilizat metodele cunoscute astăzi sub numele de eliminare gaussiană şi metoda relaxării. Lucrările lui Gauss în domeniul cuadraturilor numerice continuă munca predecesorilor săi Newton şi Cotes.

3. Ecuaţiile $E_i$ şi $E_j$ pot fi interschimbate, notaţie $(E_i) \longleftrightarrow (E_j)$.

Pentru a exprima convenabil operaţiile necesare pentru transformarea sistemului în unul triunghiular vom lucra cu matricea extinsă:

$$\widetilde{A} = [A, b] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \dots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & a_{n,n+1} \end{array}\right]$$

cu $a_{i,n+1} = b_i$.

Presupunând că $a_{11} \neq 0$, vom elimina coeficienţii lui $x_1$ din $E_j$, pentru $j = \overline{2,n}$ prin operaţia $(E_j - (a_{j1}/a_{11})E_1) \to (E_j)$. Vom proceda apoi la fel cu coeficienţii lui $x_i$, pentru $i = \overline{2,n-1}$, $j = \overline{i+1,n}$. Aceasta este posibil dacă $a_{ii} \neq 0$.

Procedura poate fi descrisă astfel: se formează o secvenţă de matrice extinse $\widetilde{A}^{(1)}$, $\widetilde{A}^{(2)}$, ..., $\widetilde{A}^{(n)}$, unde $\widetilde{A}^{(1)} = A$ şi $\widetilde{A}^{(k)}$ are elementele $a_{ij}^{(k)}$ date de

$$\left(E_i - \frac{a_{i,k-1}^{(k-1)}}{a_{k-1,k-1}^{(k-1)}} E_{k-1}\right) \longrightarrow (E_i)$$

sau desfăşurat

$$a_{i,j}^{(k)} = \begin{cases} a_{ij}^{(k-1)}, & \text{pentru} \quad i = \overline{1,k-1},\ j = \overline{1,n+1} \\ 0, & \text{pentru} \quad i = \overline{k,n},\ j = \overline{1,k-1} \\ a_{ij}^{(k-1)} - \dfrac{a_{i,k-1}^{(k-1)}}{a_{k-1,k-1}^{(k-1)}} a_{k-1,j}^{(k-1)} & \text{pentru} \quad i = \overline{k,n},\ j = \overline{k,n+1} \end{cases}.$$

**Observaţia 2.3.1.** Notaţia $a_{ij}^{(p)}$ semnifică valoarea elementului $a_{ij}$ la pasul $p$. ♢

Astfel

$$\widetilde{A}^{(k)} = \left[\begin{array}{ccccccccc|c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \dots & a_{1,k-1}^{(1)} & a_{1k}^{(1)} & \dots & a_{1n}^{(1)} & a_{1,n+1}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2,k-1}^{(2)} & a_{2,k}^{(2)} & \dots & a_{2,n}^{(2)} & a_{2,n+1}^{(2)} \\ \vdots & & \ddots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & & & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & & & & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} & \dots & a_{k-1,n}^{(k-1)} & a_{k-1,n+1}^{(k-1)} \\ \vdots & & & & 0 & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} & a_{k,n+1}^{(k)} \\ \vdots & & & & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & 0 & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} & a_{n,n+1}^{(k)} \end{array}\right]$$

reprezintă sistemul liniar echivalent în care variabila $x_{k-1}$ a fost eliminată din ecuaţiile $E_k$, $E_{k+1}, \ldots, E_n$. Sistemul corespunzător lui $\widetilde{A}^{(n)}$ este un sistem triunghiular echivalent cu sistemul iniţial

$$\begin{cases} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n &= a_{1,n+1}^{(1)} \\ a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n &= a_{2,n+1}^{(2)} \\ \vdots & \\ a_{nn}^{(n)}x_n &= a_{n,n+1}^{(n)} \end{cases}.$$

Se obţine

$$x_n = \frac{a_{n,n+1}^{(n)}}{a_{n,n}^{(n)}}$$

şi în general

$$x_i = \frac{1}{a_{ii}^{(i)}}\left(a_{i,n+1}^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)}x_j\right), \quad i = \overline{n-1,1}.$$

Pentru ca procedura să fie aplicabilă trebuie ca $a_{ii}^{(i)} \neq 0$, $i = \overline{1,n}$. Elementul $a_{ii}^{(i)}$ se numeşte *element pivot*. Dacă în timpul algoritmului de eliminare gaussiană la pasul $k$ se obţine pivotul $a_{kk}^{(k)} = 0$, se poate face interschimbarea de linii $(E_k) \leftrightarrow (E_p)$, unde $k+1 \leq p \leq n$ este cel mai mic întreg cu proprietatea $a_{pk}^{(k)} \neq 0$. În practică sunt necesare astfel de operaţii chiar şi în cazul când pivotul este nenul. Aceasta din cauză că un pivot mic în comparaţie cu elementele care urmează după el în aceeaşi coloană duce la erori de rotunjire substanţiale. Aceasta se poate remedia alegând ca pivot elementul din aceeaşi coloană care este situat sub diagonală şi care are cea mai mare valoare absolută, adică determinând $p$ astfel încât

$$|a_{pk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|$$

şi apoi făcând $(E_k) \leftrightarrow (E_p)$. Această tehnică se numeşte *pivotare maximală pe coloană* sau *pivotare parţială*.

O altă tehnică care reduce erorile şi preîntâmpină anulările în aritmetica flotantă este *pivotarea scalată pe coloană.* La primul pas se defineşte un factor de scalare pentru fiecare linie

$$s_i = \max_{j=\overline{1,n}} |a_{ij}| \text{ sau } s_i = \sum_{j=1}^{n} |a_{ij}|.$$

Dacă există $i$ astfel încât $s_i = 0$, matricea este singulară.

La paşii următori se determină ce interschimbări se vor realiza. La pasul $i$ se determină cel mai mic întreg $p$, $i \leq p \leq n$, pentru care

$$\frac{|a_{pi}|}{s_p} = \max_{1 \leq j \leq n} \frac{|a_{ji}|}{s_j}$$

şi apoi $(E_i) \leftrightarrow (E_p)$. Efectul scalării este de a ne asigura că elementul cel mai mare din fiecare coloană are mărimea relativă 1 înainte de a realiza comparaţiile pentru interschimbarea liniilor. Scalarea se realizează doar în scop de comparaţie, aşa că împărţirea cu factorul de scalare nu produce erori de rotunjire. O a treia metodă este cea a pivotării totale (sau maximale). La această metodă la pasul $k$ se determină

$$\max\{|a_{ij}|,\ i = \overline{k,n},\ j = \overline{k,n}\}$$

şi se realizează şi interschimbări de linii şi de coloane.

Pivotarea a fost introdusă de Goldstine şi von Neumann în 1947 [23].

Dacă matricea $A$ este singulară şi are rangul $p-1$ atunci se obţine

$$\widetilde{A}^{(p)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1,p-1}^{(1)} & a_{1p}^{(1)} & \dots & a_{1n}^{(1)} & a_{1,n+1}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2,p-1}^{(2)} & a_{2p}^{(2)} & \dots & a_{2n}^{(2)} & a_{2,n+1}^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{p-1,p-1}^{(p-1)} & a_{p-1,p}^{(p-1)} & \dots & a_{p-1,n}^{(p-1)} & a_{p-1,n+1}^{(p-1)} \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & a_{p,n+1}^{(p)} \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & a_{n,n+1}^{(n)} \end{bmatrix}.$$

Dacă $a_{i,n+1}^{(p)} = b_i^{(p)} = 0$, $i = \overline{p,n}$, atunci sistemul este compatibil nedeterminat, iar dacă există $q \in \{p, \dots, n\}$ astfel încât $a_{q,n+1}^{(p)} = b_q^{(p)} \neq 0$ sistemul este incompatibil.

Deci metodele de eliminare de tip Gauss permit atât rezolvarea cât şi discuţia sistemelor de ecuaţii liniare.

**Observaţia 2.3.2.** Dăm câteva sugestii care pot duce la îmbunătăţirea timpului de execuţie.

1. La metodele de pivotare nu este nevoie să se realizeze fizic interschimbarea de linii şi/sau coloane, ci se pot păstra unul (sau doi vectori) de permutări $p(q)$ cu semnificaţia $p[i](q[i])$ este linia (coloana) care a fost interschimbată cu linia (coloana) $i$.

2. Elementele de sub diagonală (care devin 0) pot să nu fie calculate.

3. Matricea $A$ se poate inversa rezolvând sistemul $Ax = e_k$, $k = \overline{1,n}$, unde $e_k$ sunt vectorii bazei canonice din $\mathbb{K}^n$. Metoda se numeşte metoda ecuaţiilor simultane.$\diamondsuit$

Să analizăm metoda eliminării a lui Gauss. Descrierea ei apare în algoritmul 2.1.

Ca *măsură a complexităţii* vom considera numărul de operaţii aritmetice flotante, desemnate prescurtat prin *flops*.

În corpul ciclului interior, liniile 10–11 avem $2n - 2i + 3$ flops, deci pentru întreg ciclul $(n - i)(2n - 2i + 3)$ flops. Pentru ciclul exterior avem un total general de

$$\sum_{i=1}^{n-1}(n - i)(2n - 2i + 3) \sim \frac{2n^3}{3}.$$

Pentru substituţia inversă avem $\Theta(n^2)$ flops.

Total general, $\Theta(n^3)$.

**Algoritmul 2.1** Rezolvă sistemul $Ax = b$ prin metoda eliminării a lui Gauss

**Intrare:** Matricea extinsă $A = (a_{ij}),\ i = \overline{1, n},\ j = \overline{1, n + 1}$
**Ieşire:** Soluţiile $x_1, \ldots, x_n$ sau un mesaj de eroare

{Eliminare}
1: **for** $i := 1$ **to** $n - 1$ **do**
2: Fie $p$ cel mai mic întreg $i \leq p \leq n$ şi $a_{p_i} \neq 0$
3: **if** $\nexists\, p$ **then**
4: mesaj (’$\nexists$ soluţie unică’); STOP
5: **end if**
6: **if** $p \neq i$ **then**
7: $(E_p) \leftrightarrow (E_i)$
8: **end if**
9: **for** $j := i + 1$ **to** $n$ **do**
10: $m_{ji} := a_{ji}/a_{ii}$;
11: $(E_j - m_{ji}E_i) \rightarrow (E_j)$;
12: **end for**
13: **end for**
14: **if** $a_{mn} = 0$ **then**
15: mesaj (’$\nexists$ soluţie unică’); STOP
16: **end if**
{Substituţie inversă}
17: $x_n := a_{n,n+1}/a_{nn}$;
18: **for** $i := n - 1$ **downto** $1$ **do**
19: $x_i = \left[a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j\right] /a_{ii}$;
20: **end for**
21: Extrage $(x_1, \ldots, x_n)$ {succes} STOP.

# 2.4. Metode bazate pe factorizare

## 2.4.1. Descompunerea LU

**Teorema 2.4.1** *Dacă eliminarea gaussiană pentru sistemul $Ax = b$ se poate realiza fără interschimbări de linii, atunci $A$ se poate factoriza în $A = LU$, $L$ - triunghiulară inferior, $U$ - triunghiulară superior (Perechea $(L, U)$ se numeşte* descompunerea $LU$ *a matricei $A$).*

**Avantaje.** $Ax = b \Leftrightarrow LUx = b \Leftrightarrow Ly = b \wedge Ux = y$.

Dacă avem de rezolvat mai multe sisteme $Ax = b_i$, $i = \overline{1, m}$, fiecare rezolvare durează $\Theta(n^3)$; dacă se factorizează la început rezolvarea unui sistem durează $\Theta(n^2)$, factorizarea durează $\Theta(n^3)$.

**Observaţia 2.4.2.** $U$ este matricea triunghiulară superior obţinută în urma eliminării gaussiene, iar $L$ este matricea multiplicatorilor $m_{ij}$. ♢

Dacă eliminarea gaussiană se face cu interschimbări avem de asemenea $A = LU$, dar $L$ nu este triunghiulară inferior.

Metoda obţinută se numeşte factorizare $LU$.

Situaţii când eliminarea gaussiană se face fără interschimbări:

- $A$ este diagonal dominantă pe linii, adică

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \quad i = \overline{1, n}$$

- $A$ este pozitiv definită $(\forall\, x \neq 0\ x^* A x > 0)$.

*Demonstraţia teoremei 2.4.1.* (schiţă) Pentru $n > 1$ partiţionăm $A$ astfel

$$A = \left[\begin{array}{c|ccc} a_{11} & a_{12} & \dots & a_{1n} \\ \hline a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array}\right] = \begin{bmatrix} a_{11} & w^* \\ v & A' \end{bmatrix},$$

unde: $v$ - vector coloană de dimensiune $n-1$, $w^*$ - vector linie de dimensiune $n-1$. Putem factoriza $A$ prin

$$A = \begin{bmatrix} a_{11} & w^* \\ v & A' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ v/a_{11} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{11} & w^* \\ 0 & A' - vw^*/a_{11} \end{bmatrix}.$$

Matricea $A' - vw^*/a_{11}$ se numeşte *complement Schur* al lui $A$ în raport cu $a_{11}$. Se continuă apoi cu descompunerea recursivă a complementului Schur:

$$A' - vw^*/a_{11} = L'U'.$$

$$\begin{aligned} A &= \begin{bmatrix} 1 & 0 \\ v/a_{11} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{11} & w^* \\ 0 & A' - vw^*/a_{11} \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 \\ v/a_{11} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{11} & w^* \\ 0 & L'U' \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 \\ v/a_{11} & L' \end{bmatrix} \begin{bmatrix} a_{11} & w^* \\ 0 & U' \end{bmatrix}. \end{aligned}$$

□

**Exemplul 2.4.3.** Să se calculeze descompunerea LU a matricei

$$A = \left[\begin{array}{c|ccc} 2 & 3 & 1 & 5 \\ \hline 6 & 13 & 5 & 19 \\ 2 & 19 & 10 & 23 \\ 4 & 10 & 11 & 31 \end{array}\right]$$

Matricea iniţială este

$$\begin{array}{c|ccc} 2 & 3 & 1 & 5 \\ \hline 3 & 4 & 2 & 4 \\ 1 & 16 & 9 & 18 \\ 2 & 4 & 9 & 21 \end{array}$$

iar primul complement Schur

$$A' - vw^*/a_{11} = \begin{pmatrix} 13 & 5 & 19 \\ 19 & 10 & 23 \\ 10 & 11 & 31 \end{pmatrix} - \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix} (3\ 1\ 5) =$$

$$= \begin{pmatrix} 13 & 5 & 19 \\ 19 & 20 & 23 \\ 10 & 11 & 31 \end{pmatrix} - \begin{pmatrix} 9 & 3 & 15 \\ 3 & 1 & 5 \\ 6 & 2 & 10 \end{pmatrix} = \begin{pmatrix} 4 & 2 & 4 \\ 16 & 9 & 18 \\ 4 & 9 & 21 \end{pmatrix}.$$

Continuăm cu descompunerea recursivă a complementţilor Schur de ordinul 2 şi 1:

$$\left[\begin{array}{cc|cc} 2 & 3 & 1 & 5 \\ 3 & 4 & 2 & 4 \\ \hline 1 & 4 & 1 & 2 \\ 2 & 1 & 7 & 17 \end{array}\right]$$

$$\begin{pmatrix} 9 & 18 \\ 9 & 21 \end{pmatrix} - \begin{pmatrix} 4 \\ 1 \end{pmatrix} (2, 4) = \begin{pmatrix} 1 & 2 \\ 7 & 17 \end{pmatrix}$$

$$\begin{array}{cccc|c} 2 & 3 & 1 & & 5 \\ 3 & 4 & 2 & & 4 \\ 1 & 4 & 1 & & 2 \\ \hline 2 & 1 & 7 & & 3 \end{array}$$

$$A' - vw^*/a_{11} = 17 - 7 \cdot 2 = 3$$

Verificare:

$$\begin{pmatrix} 2 & 3 & 1 & 5 \\ 6 & 13 & 5 & 19 \\ 2 & 19 & 10 & 23 \\ 4 & 10 & 11 & 31 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 2 & 1 & 7 & 1 \end{pmatrix} \begin{pmatrix} 2 & 3 & 1 & 5 \\ 0 & 4 & 2 & 4 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{pmatrix} \qquad \diamondsuit$$

Avem mai multe posibilităţi de alegere pentru $u_{ii}$ şi $\ell_{ii}$, $i = \overline{1,n}$. De exemplu dacă $\ell_{ii} = 1$ avem *factorizare Doolittle*, dacă $u_{ii} = 1$ avem *factorizare Crout*.

## 2.4.2. Descompunere LUP

Ideea din spatele descompunerii $LUP$ este de a găsi 3 matrice pătratice $L, U$ şi $P$ unde $L$ - triunghiulară inferior, $U$ - triunghiulară superior, $P$ matrice de permutare, astfel încât $PA = LU$.

Tripletul $(L, U, P)$ se va numi *descompunerea* $LUP$ a matricei $A$.

Rezolvarea sistemului $Ax = b$ este echivalentă cu rezolvarea a două sisteme triunghiulare, deoarece

$$Ax = b \Leftrightarrow LUx = Pb \Leftrightarrow Ly = Pb \wedge Ux = y$$

şi

$$Ax = P^{-1}LUx = P^{-1}Ly = P^{-1}Pb = b.$$

Vom alege ca pivot în locul lui $a_{11}$ elementul $a_{k1}$. Efectul este înmulţirea cu o matrice de permutare $Q$:

$$QA = \begin{bmatrix} a_{k1} & w^* \\ v & A' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ v/a_{k1} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{k1} & w^* \\ 0 & A' - vw^*/a_{k1} \end{bmatrix}.$$

Determinăm mai departe descompunerea $LUP$ a complementului Schur.

$$P'(A' - vw^*/a_{k1}) = L'U'.$$

Definim

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

care este tot o matrice de permutare. Avem acum

$$\begin{aligned}
PA &= \begin{bmatrix} 1 & 0 \\ 0 & P' \end{bmatrix} QA = \\
&= \begin{bmatrix} 1 & 0 \\ 0 & P' \end{bmatrix} \begin{bmatrix} 1 & 0 \\ v/a_{k1} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{k1} & w^* \\ 0 & A' - vw^*/a_{k1} \end{bmatrix} = \\
&= \begin{bmatrix} 1 & 0 \\ P'v/a_{k1} & P' \end{bmatrix} \begin{bmatrix} a_{k1} & w^* \\ 0 & A' - vw^*/a_{k1} \end{bmatrix} = \\
&= \begin{bmatrix} 1 & 0 \\ P'v/a_{k1} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{k1} & w^* \\ 0 & P'(A' - vw^*/a_{k1}) \end{bmatrix} = \\
&= \begin{bmatrix} 1 & 0 \\ P'v/a_{k1} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{k1} & w^* \\ 0 & L'U' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ P'v/a_{k1} & L' \end{bmatrix} \begin{bmatrix} a_{k1} & w^* \\ 0 & U' \end{bmatrix}.
\end{aligned}$$

De notat că în acest raţionament, atât vectorul coloană cât şi complementul Schur se înmulţesc cu matricea de permutare $P'$.

**Exemplul 2.4.4.** Să se calculeze descompunerea LUP a matricei

$$\begin{pmatrix} 2 & 0 & 2 & 0.6 \\ 3 & 3 & 4 & -2 \\ 5 & 5 & 4 & 2 \\ -1 & -2 & 3.4 & -1 \end{pmatrix}$$

$$\begin{array}{c|cccc} 1 & 2 & 0 & 2 & 0.6 \\ \hline 2 & 3 & 3 & 4 & -2 \\ 3 & 5 & 5 & 4 & 2 \\ 4 & -1 & -2 & 3.4 & -1 \end{array} \qquad \begin{array}{c|cccc} 3 & 5 & 5 & 4 & 2 \\ \hline 2 & 3 & 3 & 4 & -2 \\ 1 & 2 & 0 & 2 & 0.6 \\ 4 & -1 & -2 & 3.4 & -1 \end{array}$$

$$\begin{pmatrix} 3 & 4 & -2 \\ 0 & 2 & 0.6 \\ -2 & 3.4 & -1 \end{pmatrix} - \begin{pmatrix} 0.6 \\ 0.4 \\ 0.2 \end{pmatrix} (5, 4, 2) = \begin{pmatrix} 0 & 1.6 & -3.2 \\ -2 & 0.4 & -0.2 \\ -1 & 4.2 & -0.6 \end{pmatrix}$$

$$\begin{array}{cccccc} 3 & 5 & 5 & 4 & 2 \\ 2 & 0.6 & 0 & 1.6 & -3.2 \\ 1 & 0.4 & -2 & 0.4 & -0.2 \\ 4 & 0.2 & -1 & 4.2 & -0.6 \end{array} \qquad \begin{array}{cccccc} 3 & 5 & 5 & 4 & 2 \\ 2 & 0.6 & 0 & 1.6 & -3.2 \\ 1 & 0.4 & -2 & 0.4 & -0.2 \\ 4 & -0.2 & -1 & 4.2 & -0.6 \end{array}$$

$$\begin{array}{cccccc} 3 & 5 & 5 & 4 & 2 \\ 1 & 0.4 & -2 & 0.4 & -0.2 \\ 2 & 0.6 & 0 & 1.6 & -3.2 \\ 4 & -0.2 & -1 & 4.2 & 0.6 \end{array} \qquad \begin{array}{cccccc} 3 & 5 & 5 & 4 & 2 \\ 1 & 0.4 & -2 & 0.4 & -0.2 \\ 2 & 0.6 & 0 & 1.6 & -3.2 \\ 4 & -0.2 & -0.5 & 4 & -0.5 \end{array}$$

$$\begin{array}{rrrrr} 3 & 5 & 5 & 4 & 2 \\ 1 & 0.4 & -2 & 0.4 & -0.2 \\ 2 & 0.6 & 0 & 1.6 & -3.2 \\ 4 & -0.2 & 0.5 & 4 & -0.5 \end{array} \qquad \begin{array}{rrrrr} 3 & 5 & 5 & 4 & 2 \\ 1 & 0.4 & -2 & 0.4 & -0.2 \\ 4 & -0.2 & 0.5 & 4 & -0.5 \\ 2 & 0.6 & 0 & 1.6 & -3.2 \end{array}$$

$$\begin{array}{rrrrr} 3 & 5 & 5 & 4 & 2 \\ 1 & 0.4 & -2 & 0.4 & -0.2 \\ 4 & -0.2 & 0.5 & 4 & -0.5 \\ 2 & 0.6 & 0 & 0.4 & -3 \end{array}$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 & 0 & 2 & 0.6 \\ 3 & 3 & 4 & -2 \\ 5 & 5 & 4 & 2 \\ -1 & -2 & 3.4 & -1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.4 & 1 & 0 & 0 \\ -0.2 & 0.5 & 1 & 0 \\ 0.6 & 0 & 0.4 & 1 \end{pmatrix} \begin{pmatrix} 5 & 5 & 4 & 2 \\ 0 & -2 & 0.4 & -0.2 \\ 0 & 0 & 4 & -0.5 \\ 0 & 0 & 0 & -3 \end{pmatrix} \qquad \diamondsuit$$

## 2.4.3. Factorizarea Cholesky

Matricele hermitiene pozitiv definite pot fi descompuse în factori triunghiulari de două ori mai repede decât matricele generale. Algoritmul standard pentru aceasta, factorizarea Cholesky[4], este o variantă a eliminării gaussiene ce operează atât la stânga cât şi la dreapta matricei, păstrând şi exploatând simetria.

Sistemele cu matrice hermitiene pozitiv definite joacă un rol important în algebra liniară numerică şi în aplicaţii. Datorită legilor fundamentale ale fizicii, multe matrice care intervin în probleme practice sunt de acest tip.

Reamintim câteva proprietăţi ale matricelor hermitiene. Dacă $A$ este o matrice $m \times m$ hermitiană pozitiv definită şi $X$ este o matrice de tip $m \times n$ de rang maxim, cu $m \geq n$, atunci matricea $X^*AX$ este de asemenea hermitiană pozitiv definită deoarece $(X^*AX)^* = X^*A^*X = X^*AX$ şi pentru orice vector $x \neq 0$ avem $Xx \neq 0$ şi astfel $x^*\left(X^*AX\right)x = (Xx)^*A(Xx) > 0$. Alegând pe post de $X$ o matrice $m \times n$ cu un 1 în fiecare coloană şi zero în rest, putem scrie orice submatrice principală $n \times n$ a lui $A$

[4]Andre-Louis Cholesky (1875-1918) ofiţer francez, specialist în topografie şi geodezie, a activat în Creta şi Africa de nord înaintea începerii primului război mondial. A dezvoltat metoda care îi poartă numele şi a aplicat-o la calculul soluţiilor ecuaţiilor normale pentru probleme de aproximare în sensul celor mai mici pătrate care apar în geodezie. Lucrarea sa a fost publicată postum în 1924, de către camaradul său Benoît, în Bulletin Géodesique. Se pare că a luat parte la misiunea militară franceză din România în timpul primului război mondial.

sub forma $X^*AX$. De aceea, orice submatrice principală a lui $A$ trebuie să fie pozitiv definită. În particular, orice element diagonal al lui $A$ este un număr real pozitiv.

Valorile proprii ale unei matrice hermitiene pozitiv definite sunt de asemenea numere reale pozitive, Dacă $Ax = \lambda x$ pentru $x \neq 0$, avem $x^*Ax = \lambda x^*x > 0$ şi deci $\lambda > 0$ şi reciproc, dacă toate valorile proprii sunt definite, atunci $A$ este pozitiv definită. Vectorii proprii ce corespund valorilor proprii distincte ale unei matrice hermitiene sunt ortogonali. Presupunem că $Ax_1 = \lambda_1 x_1$ şi $Ax_2 = \lambda_2 x_2$ cu $\lambda_1 \neq \lambda_2$. Atunci

$$\lambda_2 x_1^* x_2 = x_1^* A x_2 = \overline{x_2^* A x_1} = \overline{\lambda_1 x_2^* x_1} = \lambda_1 x_1^* x_2,$$

aşă că $(\lambda_1 - \lambda_2) x_1^* x_2 = 0$. Deoarece $\lambda_1 \neq \lambda_2$, rezultă că $x_1^* x_2 = 0$. Matricele hermitiene sunt de asemenea normale ($AA^* = A^*A = A^2$).

O *factorizare Cholesky* a unei matrice $A$ este o descompunere de forma

$$A = R^*R, \qquad r_{jj} > 0, \tag{2.4.1}$$

unde $R$ este o matrice triunghiulară superior.

**Teorema 2.4.5** *Orice matrice hermitiană pozitiv definită $A \in \mathbb{C}^{m\times m}$ are o factorizare Cholesky* (2.4.1) *unică.*

*Demonstraţie.* (Existenţa) Deoarece $A$ este hermitiană şi pozitiv definită $a_{11} > 0$ şi putem pune $\alpha = \sqrt{a_{11}}$. Observăm că

$$\begin{aligned} A &= \begin{bmatrix} a_{11} & w^* \\ w & K \end{bmatrix} \\ &= \begin{bmatrix} \alpha & 0 \\ w/\alpha & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^*/a_{11} \end{bmatrix} \begin{bmatrix} \alpha & w^*/\alpha \\ 0 & I \end{bmatrix} = R_1^* A_1 R_1. \end{aligned} \tag{2.4.2}$$

Acesta este pasul de bază care este repetat în factorizarea Cholesky. Submatricea $K - ww^*/a_{11}$ fiind o submatrice principală de tip $(m-1)\times(m-1)$ a matricei pozitiv definite $R_1^* A R_1^{-1}$este pozitiv definită şi deci elementul ei situat în colţul din stânga sus este pozitiv. Se arată prin inducţie că toate submatricele care apar în cursul factorizării sunt pozitiv definite şi *procesul nu poate eşua.* Continuăm cu factorizarea lui $A_1 = R_2^* A_2 R_2$ şi astfel $A = R_1^* R_2^* A_2 R_2 R_1$; procesul poate continua până se ajunge la colţul din dreapta jos, obţinându-se

$$A = \underbrace{R_1^* R_2^* \dots R_m^*}_{R^*} \underbrace{R_m \dots R_2 R_1}_{R},$$

care are chiar forma dorită.

(Unicitatea) De fapt procedeul de mai sus stabileşte şi unicitatea. La fiecare pas (2.4.2), valoare $\alpha = \sqrt{a_{11}}$ este determinată din forma factorizării $R^*R$ şi odată ce $\alpha$ este determinat, prima linie a lui $R_1^*$ este de asemenea determinată. Deoarece cantităţile analoage sunt determinate la fiecare pas al reducerii, întreaga factorizare este unică. □

Când se implementează factorizarea Cholesky, este nevoie să se reprezinte explicit doar jumătate din matricea asupra căreia se operează. Această simplificare permite evitarea a jumătate din operaţiile aritmetice. Se dă mai jos una din multele posibilităţi de prezentare formală a algoritmului (algoritmul 2.2). Matricea $A$ de la intrare conţine diagonala principală şi jumătatea de deasupra diagonalei pricipale a matricei hermitiene şi pozitiv definite de tip $m \times m$ ce urmează a fi factorizată. În implementări practice se pot utiliza scheme de memorare comprimată ce evită irosirea spaţiului pentru jumătate din matricea pătratică. Matricea de ieşire reprezintă factorul triunghiular superior din factorizarea $A = R^*R$. Fiecare iteraţie externă corespunde unei singure factorizări elementare: partea triunghiulară superior a submatricei $R^*_{k:m,k:m}$ reprezintă partea supradiagonală a matricei hermitiene ce trebuie factorizată la pasul $k$.

**Algoritmul 2.2** Factorizare Cholesky

$R := A$;
**for** $k := 1$ **to** $m$ **do**
  **for** $j := k + 1$ **to** $m$ **do**
    $R_{j,j:m} := R_{j,j:m} - R_{k,j:m}R_{k,j}/R_{k,k}$
  **end for**
  $R_{k,k:m} := R_{k,k:m}/\sqrt{R_{k,k}}$
**end for**

Operaţiile aritmetice în factorizarea Cholesky (algoritmul 2.2) sunt dominate de ciclul interior. O singură execuţie a liniei

$$R_{j,j:m} := R_{j,j:m} - R_{k,j:m}R_{kj}/R_{kk}$$

necesită o împărţire, $m-j+1$ înmulţiri şi $m-j+1$ scăderi, deci un total de $\sim 2(m-j)$ flops. Acest calcul este repetat pentru fiecare $j$ de la $k+1$ la $m$ şi acest ciclu este repetat pentru fiecare $k$ de la 1 la $m$. Suma se evaluează direct

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

deci jumătate din volumul de calcule necesar pentru eliminarea gaussiană.

## 2.4.4. Descompunerea QR

**Teorema 2.4.6** *Fie $A \in \mathbb{R}^{m\times n}$, cu $m \geq n$. Atunci există o matrice ortogonală unică $Q$ de tip $m \times n$ şi o matrice triunghiulară superior unică, de tip $n \times n$ $R$ cu diagonala pozitivă ($r_{ii} > 0$) astfel încât $A = QR$.*

*Demonstraţie.* Va rezulta din algoritmul 2.3, care va fi dat în această secţiune. □

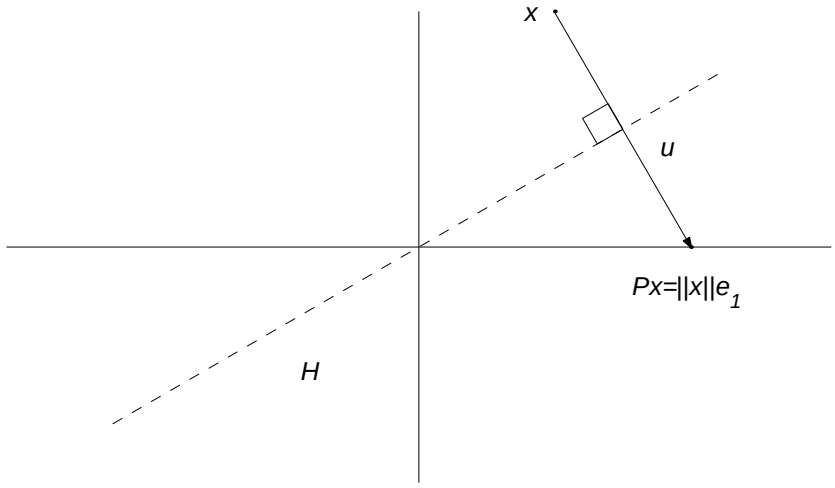



Figura 2.2: Un reflector Householder

**Transformări Householder**

O *transformare Householder* [5] (sau *reflexie*) este o matrice de forma $P = I - 2uu^T$, unde $\|u\|_2 = 1$. Se verifică uşor că $P = P^T$ şi că $PP^T = \left(I - 2uu^T\right)\left(I - 2uu^T\right) = I - 4uu^T + 4uu^Tuu^T = I$, deci $P$ este o matrice simetrică şi ortogonală. Ea se numeşte reflexie deoarece $Px$ este reflexia lui $x$ faţă de hiperplanul $H$ ce trece prin origine şi este ortogonal pe $u$ (figura 2.2).

Dându-se un vector $x$, este uşor de găsit reflexia Householder care anulează toate componentele lui $x$, exceptând prima: $Px = [c, 0, \ldots, 0]^T = ce_1$. Vom face aceasta după cum urmează. Scriem $Px = x - 2u(u^Tx) = ce_1$, deci $u = \frac{1}{2(u^Tx)}(x - ce_1)$, adică $u$ este o combinaţie liniară a lui $x$ şi $e_1$. Deoarece $\|x\|_2 = \|Px\|_2 = |c|$, $u$ trebuie să fie paralel cu vectorul $\tilde{u} = x \pm \|x\|_2 e_1$, deci $u = \tilde{u}/\|\tilde{u}\|_2$. Se poate verifica că orice alegere de semn ne conduce la un $u$ ce satisface $Px = ce_1$, atât timp cât $\tilde{u} \neq 0$. Vom utiliza $\tilde{u} = x + \mathrm{sign}(x_1)\|x\|_2 e_1$, deoarece astfel nu va apare nici o anulare flotantă la calculul

[5] Alston S. Householder (1904-1993), matematician american. Contribuţii importante în biologia matematică şi mai ales în algebra liniară numerică. Cartea sa cea mai importantă, „The theory of matrices in numerical analysis” a avut un impact deosebit asupra dezvoltării analizei numerice şi informaticii.

componentelor lui $\tilde{u}$. Dacă $x_1 = 0$, vom conveni să luam $\text{sign}(x_1) = 1$. Rezumând, vom lua

$$\tilde{u} = \begin{bmatrix} x_1 + \text{sign}(x_1)\|x\|_2 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \text{ cu } u = \frac{\tilde{u}}{\|u\|_2}.$$

Vom scrie aceasta sub forma $u = House(x)$. În practică, putem memora $\tilde{u}$ în locul lui $u$ pentru a reduce efortul de calcul al lui $u$ şi să utilizăm formula $P = I - \frac{2}{\|u\|_2^2}\tilde{u}\tilde{u}^T$ în loc de $P = I - 2uu^T$.

**Exemplul 2.4.7.** Vom arăta cum se poate calcula descompunerea QR a unei matrice $5 \times 4$ utilizând transformări Householder. Acest exemplu poate face procedeul mai uşor de înţeles în cazul general. În calculele de mai jos $P_i$ sunt matrice ortogonale $5 \times 5$, $x$ este un element generic nenul, iar $o$ o poziţie nulă.

1. Alegem $P_1$ astfel încât $A_1 \equiv P_1 A = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & x & x & x \\ o & x & x & x \\ o & x & x & x \end{bmatrix}$.

2. Alegem $P_2 = \begin{bmatrix} I_1 & 0 \\ 0 & P_2' \end{bmatrix}$ astfel încât $A_2 \equiv P_2 A_1 = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & x & x \end{bmatrix}$.

3. Alegem $P_3 = \begin{bmatrix} I_2 & 0 \\ 0 & P_3' \end{bmatrix}$ astfel încât $A_3 \equiv P_3 A_2 = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & x \end{bmatrix}$.

4. Alegem $P_4 = \begin{bmatrix} I_3 & 0 \\ 0 & P_4' \end{bmatrix}$ astfel încât $A_4 \equiv P_4 A_3 = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & o \end{bmatrix}$.

Aici am ales matricele Householder $P_i'$ pentru a anula elementele subdiagonale din coloana $i$; aceasta nu distruge zerourile deja introduse în coloanele precedente. Să notăm matricea finală $5 \times 4$ cu $\tilde{R} = A_4$. Atunci $A = P_1^T P_2^T P_3^T P_4^T \tilde{R} = QR$, unde $Q$ este

formată din primele patru coloane a lui $P_1^T P_2^T P_3^T P_4^T = P_1 P_2 P_3 P_4$ (deoarece $P_i$ sunt simetrice iar $R$ este formată din primele patru linii ale lui R. $\diamondsuit$

Algoritmul 2.3 descrie procesul de calcul pentru descompunerea QR bazată pe transformări HouseHolder.

**Algoritmul 2.3** Factorizare QR utilizând reflexii HouseHolder

1: **for** $i := 1$ **to** $\min(m-1, n)$ **do**
2: $u_i := House(A_{i:m,i})$;
3: $P_i' := I - 2u_i u_i^T$;
4: $A_{i:m,i:n} := P_i' A_{i:m,i:n}$;
5: **end for**

**Algoritmul 2.4** Calculul produsului $Q^T b$

1: **for** $i := 1$ **to** $n$ **do**
2: $\gamma := -2u_i^T b_{i:m}$;
3: $b_{i:m} := b_{i:m} + \gamma u_i$;
4: **end for**

**Algoritmul 2.5** Calculul produsului $Qx$

1: **for** $k := n$ **downto** $1$ **do**
2: $x_{k:m} := x_{k:m} - 2u_k(u_k^* x_{k:m})$;
3: **end for**

Iată câteva detalii de implementare. Nu avem niciodată nevoie să construim explicit $P_i$ ci doar să efectuăm înmulţirea

$$(I - 2u_i u_i^T) A_{i:m,i:n} = A_{i:m,i:n} - 2u_i(u_i^T A_{i:m,i:n}),$$

care este mai puţin costisitoare. Pentru a memora $P_i$ avem nevoie doar de $u_i$ sau de $\tilde{u}_i$ şi $\|\tilde{u}_i\|_2$. Acestea pot fi memorate în coloana $i$ al lui $A$; de fapt nu este nevoie ca ea să fie schimbată! Astfel descompunerea QR poate fi scrisă peste $A$, unde $Q$ este memorată în forma factorizată $P_1 \dots P_{n-1}$, iar $P_i$ este memorată prin vectorul de reflexie $\tilde{u}_i$, în coloana $i$ a lui $A$, sub diagonală. (Avem nevoie de un tablou suplimentar este memorată în forma factorizată $P_1 \dots P_{n-1}$ iar $P_i$ este memorată ca $\tilde{u}_i$ în partea subdiagonală a coloanei $i$ a lui $A$. Mai este nevoie de un tablou suplimentar de lungime $n$ pentru elementul de sus al lui $\tilde{u}_i$, deoarece diagonala este deja ocupată de elementele $R_{ii}$.

Pornind de la observaţia

$$Ax = b \Leftrightarrow QRx = b \Leftrightarrow Rx = Q^T b,$$

putem alege următoarea strategie pentru rezolvarea sistemului de ecuaţii liniare $Ax = b$:

1. Se determină factorizarea $A = QR$ a lui $A$;
2. Se calculează $y = Q^T b$;
3. Se rezolvă sistemul triunghiular superior $Rx = y$.

Calculul lui $Q^T b$ se realizează prin $Q^T b = P_n P_{n-1} \dots P_1 b$, deci avem nevoie să păstram produsul lui $b$ cu $P_1$, $P_2$, $\dots$, $P_n$ – vezi algoritmul 2.4. Similar, calculul produsului $Qx$ se poate realiza prin acelaşi proces, executat în ordine inversă (algoritmul 2.5).

Costul calculului descompunerii $A = QR$ este $2n^2m - \frac{2}{3}n^3$, costul pentru calculul lui $Q^T b$ este $O(mn)$, iar costul pentru calculul lui $Qx$ este de asemenea $O(mn)$.

Dacă dorim să calculăm matricea $Q$ explicit, putem construi $QI$ prin algoritmul 2.5 calculând coloanele sale $Qe_1$, $Qe_2$, $\dots$, $Qe_n$.

**Rotaţii Givens**

O *rotaţie Givens*

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

roteşte vectorul $x \in \mathbb{R}^2$ cu un unghi $\theta$.

De asemenea, avem nevoie de rotaţia Givens cu un unghi $\theta$ în coordonatele $i$ şi $j$:

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

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

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

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

**Exemplul 2.4.8.** Vom ilustra doi paşi intermediari din calculul descompunerii QR a unei matrice $5 \times 4$ utilizând rotaţii Givens. Pentru a trece de la

$$\begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & x & x \end{bmatrix} \quad \text{la} \quad \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & x \end{bmatrix}$$

efectuăm înmulţirile

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & c & s \\ & & & -s & c \end{bmatrix} \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & x & x \end{bmatrix} = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & o & x \end{bmatrix}$$

şi

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & c' & s' & \\ & & -s' & c' & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & x & x \\ o & o & x & x \end{bmatrix} = \begin{bmatrix} x & x & x & x \\ o & x & x & x \\ o & o & x & x \\ o & o & o & x \\ o & o & o & x \end{bmatrix}.$$

♢

Costul descompunerii QR cu rotaţii Givens este de două ori costul descompunerii cu reflexii Householder. Ele sunt utilizate în alte aplicaţii (de exemplu valori proprii).

Iată câteva detalii de implementare. Calculul sinusului şi cosinusului din matricea Givens este dat în algoritmul 2.6. El necesită 5 flops şi un radical. Nu necesită funcţii tri-

**Algoritmul 2.6** Dându-se scalarii $a$ şi $b$, calculează elementele $c = \cos(\theta)$ şi $s = \sin(\theta)$ ale unei matrice Givens

**function** $[c, s]$ =**givens**$(a, b)$
**if** $b = 0$ **then**
  $c := 1; \quad s := 0;$
**else**
  **if** $|b| > |a|$ **then**
    $\tau := -a/b;\ s := 1/\sqrt{1+\tau^2};\ c := s\tau;$
  **else**
    $\tau := -b/a;\ c := 1/\sqrt{1+\tau^2};\ s := c\tau;$
  **end if**
**end if**

gonometrice inverse şi nu dă depăşire superioară. Descompunerea QR bazată pe rotaţii

**Algoritmul 2.7** Factorizare $A = QR$ cu ajutorul matricei Givens; rezultatul $R$ se scrie peste $A$.

**Intrare:** $A \in \mathbb{R}^{m \times n}$
**Ieşire:** $R = Q^T A$, unde $R$ este triunghiulară superior.

**for** $j := 1$ **to** $n$ **do**
  **for** $i := m - 1$ **downto** $j + 1$ **do**
    $[c, s] = \mathbf{givens}(A_{i-1,j}, A_{i,j})$;
    $A_{i-1:i,j:n} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T A_{i-1:i,j:n}$;
  **end for**
**end for**

Givens este descrisă de algoritmul 2.7 Acest algoritm necesită $3n^2(m - n/3)$ flops. De notat că se pot utiliza şi alte secvenţe de rotaţii; de exemplu primele două **for**-uri din algoritmul 2.7 se pot înlocui cu

$$\begin{array}{ll} \textbf{for } i & := m \textbf{ downto } 2 \\ & \textbf{for } j := 1 \textbf{ to } \min\{i-1, n\} \end{array}$$

Astfel, zerourile vor fi introduse linie cu linie. O altă schimbare posibilă se referă la planele de rotaţie: în loc să rotim liniile $i - 1$ şi $i$ ca în algoritmul 2.7, am putea roti liniile $j$ şi $i$:

**for** $j := 1$ **to** $n$ **do**
  **for** $i := m - 1$ **downto** $j + 1$ **do**
    $[c, s] = \mathbf{givens}(A_{j,j}, A_{i,j})$;
    $A_{[j\ i],j:n} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T A_{[j\ i],j:n}$;
  **end for**
**end for**

Deoarece o rotaţie Givens anulează exact un element, trebuie să memorăm informaţii despre rotaţie pe poziţia acelui element. Vom face asta după cum urmează. Fie $s = \sin\theta$ şi $c = \cos\theta$. Dacă $|s| < |c|$ se memorează $s \cdot \mathrm{sign(c)}$; altfel se memorează $\frac{\mathrm{sign(s)}}{c}$. Pentru a recupera $s$ şi $c$ din valoarea memorată (să o numim $p$) vom proceda astfel: dacă $|p| < 1$, atunci $s := p$ şi $c = \sqrt{1 - s^2}$; altfel $c := 1/p$ şi $s := \sqrt{1 - c^2}$. Motivul pentru care nu memorăm $s$ şi nu calculăm $c = \sqrt{1 - s^2}$ este acela că dacă $s$ este apropiat de 1, $c$ va fi imprecis (datorită anulării flotante). Se poate recupera fie $s$ şi $c$, fie $-s$ şi $-c$; acest lucru este convenabil în practică.

Secvenţa de rotaţii Givens se poate aplica în aşa fel ca să se aplice mai puţine flops decât în variantele prezentate aici. Se ajunge astfel la rotaţii Givens rapide (vezi [24, capitolul 5]).

## 2.5. Rafinarea iterativă

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

$$A(\overline{x} + \Delta x_1) = b \Rightarrow A\Delta x_1 = b - A\overline{x}$$

Se rezolvă sistemul şi se obţine un nou $\overline{x}$, $\overline{x} := x + \Delta x_1$. Dacă din nou $Ax \neq b$, se calculează o nouă corecţie până când

$$\|\Delta x_i - \Delta x_{i-1}\| < \varepsilon \quad \text{sau} \quad \|Ax - b\| < \varepsilon.$$

Calculul cantităţii $r = b - Ax$, numită *reziduu*, se va efectua în dublă precizie.

## 2.6. Algoritmul lui Strassen pentru înmulţirea matricelor

Fie $A, B \in \mathbb{K}^{n\times n}$. Dorim să calculăm $C = AB$. Presupunem că $n = 2^k$. Partiţionăm $A$ şi $B$

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \quad C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

În algoritmul clasic avem $8$ înmulţiri şi $4$ adunări pentru un pas, iar timpul de execuţie $T(n) = \Theta(n^3)$, deoarece $T(n) = 8T(n/2) + \Theta(n^2)$.

Interesul este de a reduce numărul de înmulţiri. Volker Strassen a descoperit o metodă de a reduce numărul de înmulţiri pe pas la 7. Se calculează următoarele cantităţi:

$$\begin{aligned}
p_1 &= (a_{11} + a_{22})(b_{11} + b_{22}), \\
p_2 &= (a_{21} + a_{22})b_{11}, \\
p_3 &= a_{11}(b_{12} - b_{22}), \\
p_4 &= a_{22}(b_{21} - b_{11}), \\
p_5 &= (a_{11} + a_{12})b_{22}, \\
p_6 &= (a_{21} - a_{11})(b_{11} + b_{12}), \\
p_7 &= (a_{12} - a_{22})(b_{21} + b_{22}),
\end{aligned}$$

$$\begin{aligned}
c_{11} &= p_1 + p_4 - p_5 + p_7, \\
c_{12} &= p_3 + p_5, \\
c_{21} &= p_2 + p_4, \\
c_{22} &= p_1 + p_3 - p_2 + p_6.
\end{aligned}$$

Procedura este descrisă detaliat de algoritmul 2.8. Deoarece avem 7 înmulţiri şi 18

**Algoritmul 2.8** Algoritmul lui Strassen pentru înmulţirea a două matrice

Presupunem că $m = 2^q$ şi că $A, B \in \mathbb{R}^{m\times m}$. Dacă $m_{\min} = 2^d$ cu $d \leq q$, atunci acest algoritm calculează $C = AB$ aplicând recursiv procedura lui Strassen de $q - d$ ori.

**function** $C$ = **strass**$(A, B, m, m_{\min})$
**if** $m \leq m_{\min}$ **then**
  $C := AB$;
**else**
  $n := m/2;\ u := 1 : n;\ v := n + 1 : m;$
  $P_1$ := **strass**$(A(u,u) + A(v,v), B(u,u) + B(v,v), n, m_{\min})$;
  $P_2$ := **strass**$(A(v,u) + A(v,v), B(u,u), n, m_{\min})$;
  $P_3$ := **strass**$(A(u,u), B(u,v) - B(v,v), n, m_{\min})$;
  $P_4$ := **strass**$(A(v,v), B(v,u) - B(u,u), n, m_{\min})$;
  $P_5$ := **strass**$(A(u,u) + A(u,v), B(v,v), n, m_{\min})$;
  $P_6$ := **strass**$(A(v,u) - A(u,u), B(u,u) + B(u,v), n, m_{\min})$;
  $P_7$ := **strass**$(A(u,v) - A(v,v), B(v,u) + B(v,v), n, m_{\min})$;
  $C(u,u) := P_1 + P_4 - P_5 + P_7$;
  $C(u,v) := P_3 + P_5$;
  $C(v,u) := P_2 + P_4$;
  $C(v,v) := P_1 + P_3 - P_2 + P_6$;
**end if**

adunări sau scăderi pe pas, timpul de execuţie verifică relaţia de recurenţă

$$T(n) = 7T(n/2) + \Theta(n^2),$$

cu soluţia

$$T(n) = \Theta(n^{\log_2 7}) \sim 28n^{\log_2 7}.$$

Algoritmul se poate generaliza pentru matrice cu dimensiunea $n = m \cdot 2^k$.

Dacă $n$ este impar ultima coloană a lui se calculează prin metode standard şi se execută algoritmul lui Strassen pentru matrice de dimensiune $n - 1$.

$$m \cdot 2^{k+1} \to m \cdot 2^k$$

$p$-urile se pot calcula în paralel; la fel şi $c$-urile.

Accelerarea teoretică obţinută pentru înmulţirea matricelor se traduce printr-o accelerare a inversării matricelor, deci şi a rezolvării sistemelor de ecuaţii liniare. Dacă notăm cu $M(n)$ timpul de înmulţire a două matrice pătratice de ordinul $n$ şi cu $I(n)$ timpul de inversare a unei matrice de acelaşi ordin, atunci $M(n) = \Theta(n)$. Vom demonstra aceasta în două etape: arătăm că $M(n) = O(I(n))$ şi apoi că $I(n) = O(M(n))$.

**Teorema 2.6.1 (Înmulţirea nu este mai grea decât inversarea)** *Dacă putem inversa o matrice $n \times n$ în timp $I(n)$, unde $I(n) = \Omega(n^2)$ satisface condiţia de regularitate $I(3n) = O(I(n))$, atunci putem înmulţi două matrice de ordinul $n$ în timp $O(I(n))$.*

*Demonstraţie.* Fie $A$ şi $B$ două matrice $n \times n$. Vrem să calculăm $C = AB$. Definim matricea $D$ de ordinul $3n \times 3n$ astfel:

$$D = \begin{pmatrix} I_n & A & 0 \\ 0 & I_n & B \\ 0 & 0 & I_n \end{pmatrix}.$$

Inversa lui $D$ este

$$D^{-1} = \begin{pmatrix} I_n & -A & AB \\ 0 & I_n & -B \\ 0 & 0 & I_n \end{pmatrix},$$

şi, astfel, putem calcula produsul $AB$ luând submatricea de ordinul $n \times n$ din colţul din dreapta sus al matricei $D^{-1}$.

Putem calcula matricea $D$ într-un timp de ordinul $\Theta(n^2) = O(I(n))$ şi conform condiţiei de regularitate o putem inversa într-un timp $O(I(3n)) = O(I(n))$. Deci

$$M(n) = O(I(n)).$$

□

Să observăm că $I(n)$ satisface condiţia de regularitate doar dacă $I(n)$ nu are salturi mari în valoare. De exemplu, dacă $I(n) = \Theta(n^c \log^d n)$, pentru orice constante $c > 0$, $d \geq 0$, atunci $I(n)$ satisface condiţia de regularitate.

Demonstraţia că inversarea matricelor nu este mai grea decât înmulţirea lor se bazează pe proprietăţile matricelor simetrice, pozitiv definite.

**Teorema 2.6.2 (Inversarea nu este mai grea decât înmulţirea)** *Dacă putem înmulţi două matrice reale $n \times n$ în timp $M(n)$ unde $M(n) = \Omega(n^2)$ şi $M(n)$ satisface condiţiile de regularitate $M(n) = O(M(n+k))$ pentru orice $k$, $0 \leq k \leq n$ şi $M(n/2) \leq cM(n)$, pentru orice constantă $c < 1/2$, atunci putem calcula inversa unei matrice reale nesingulare de ordinul $n \times n$, în timp $O(M(n))$.*

*Demonstraţie.* Putem presupune că $n$ este multiplu de 2, deoarece

$$\begin{pmatrix} A & 0 \\ 0 & I_k \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & I_k \end{pmatrix},$$

pentru orice $k \in \mathbb{N}^*$. Aşadar, alegând pe $k$ astfel încât $n + k$ să fie o putere a lui 2, extindem matricea la o dimensiune care este puterea următoare a lui 2 şi obţinem

răspunsul dorit din răspunsul problemei pentru matricea extinsă în acest mod. Condiţiile de regularitate ne asigură că extinderea nu cauzează creşterea timpului decât cu cel mult un factor constant.

Pentru moment, să presupunem că $A$ este o matrice $n \times n$ simetrică şi pozitiv-definită; o vom partiţiona în patru submatrice de ordinul $n/2 \times n/2$ :

$$A = \begin{pmatrix} B & C^T \\ C & D \end{pmatrix}. \tag{2.6.1}$$

Dacă

$$S = D - CB^{-1}C^T \tag{2.6.2}$$

este complementul Schur al lui $A$ în raport cu $B$, atunci

$$A^{-1} = \begin{pmatrix} B^{-1} + B^{-1}C^T S^{-1} CB^{-1} & -B^{-1}C^T S^{-1} \\ -S^{-1}CB^{-1} & S^{-1} \end{pmatrix}, \tag{2.6.3}$$

relaţie care se poate verifica prin înmulţire. Matricele $B^{-1}$ şi $S^{-1}$ există, deoarece $A$ este simetrică şi pozitiv definită, deoarece atât $B$ cât şi $S$ sunt simetrice şi pozitiv definite. În plus, $B^{-1}C^T = (CB^{-1})^T$ şi $B^{-1}C^T S^{-1} = (S^{-1}CB^{-1})^T$. Putem folosi ecuaţiile (2.6.2) şi (2.6.3) pentru a specifica un algoritm recursiv în care intervin patru înmulţiri de matrice de ordinul $n/2 \times n/2$:

$$\begin{aligned} &C \cdot B^{-1}, \\ &(C \cdot B^{-1}) \cdot C^T \\ &S^{-1} \cdot (CB^{-1}) \\ &\left(CB^{-1}\right)^T \cdot \left(S^{-1}CB^{-1}\right). \end{aligned}$$

Întrucât matricele de ordinul $n/2 \times n/2$ se înmulţesc folosind un algoritm pentru matrice de ordinul $n \times n$, inversarea matricelor simetrice pozitiv-definite se poate rezolva în timpul

$$I(n) \leq 2I(n/2) + 4M(n) + O(n^2) = 2I(n/2) + O(M(n)) = O(M(n)).$$

Rămâne de studiat cazul când $A$ este inversabilă, dar nu este simetrică şi pozitiv definită. Deoarece $A^T A$ este simetrică şi pozitiv definită, problema inversării lui $A$ se reduce la problema inversării lui $A^T A$. Reducerea se bazează pe observaţia că, atunci când $A$ este o matrice nesingulară de ordinul $n \times n$, avem

$$A^{-1} = \left(A^T A\right)^{-1} A^T,$$

deoarece $((A^T A)^{-1} A^T) = (A^T A)^{-1}(A^T A) = I_n$, şi inversa unei matrice este unică. Prin urmare, putem calcula $A^{-1}$, înmulţind întâi pe pe $A^T$ cu $A$ pentru a obţine $A^T A$,

inversând apoi matricea simetrică şi pozitiv definită $A^TA$ prin algoritmul divide et impera prezentat, şi, în final înmulţind rezultatul cu $A^T$. Fiecare din aceşti paşi necesită un timp $O(M(n))$. Astfel, orice matrice nesingulară cu elemente reale poate fi inversată într-un timp de ordinul $O(M(n))$. □

Demonstraţia teoremei 2.6.2 sugerează un mod de a rezolva ecuaţii de forma $Ax = b$, cu $A$ nesingulară, fără pivotare, rezolvând ecuaţia echivalentă $(A^TA)x = b$ prin factorizare Cholesky. Dar în practică, descompunerea LUP funcţionează mai bine, iar transformarea propusă măreşte numărul de condiţionare.

## 2.7. Rezolvarea iterativă a sistemelor algebrice liniare

Dorim să calculăm soluţia sistemului liniar

$$Ax = b, \tag{2.7.1}$$

când $A$ este inversabilă. Presupunem că am găsit o matrice $T$ şi un vector $c$ astfel încât $I - T$ să fie inversabilă şi astfel încât punctul fix unic al ecuaţiei

$$x = Tx + c \tag{2.7.2}$$

să fie egal cu soluţia sistemului $Ax = b$. Fie $x^*$ soluţia lui (2.7.1) sau, echivalent, a lui (2.7.2).

Iteraţiile: se dă un $x^{(0)}$ arbitrar; se defineşte şirul $(x^{(k)})$ prin

$$x^{(k+1)} = Tx^{(k)} + c, \quad k \in \mathbb{N} \tag{2.7.3}$$

**Lema 2.7.1** *Dacă $\rho(X) < 1$, există $(I - X)^{-1}$ şi*

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

*Demonstraţie.* Fie

$$S_k = I + X + \cdots + X^k$$

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

$$\lim_{k\to\infty}(I - X)S_k = I \Rightarrow \lim_{k\to\infty} S_k = (I - X)^{-1}$$

căci $X^{k+1} \to 0 \Leftrightarrow \rho(X) < 1$. □

**Teorema 2.7.2** *Propoziţiile următoare sunt echivalente*

*(1) metoda (2.7.3) este convergentă*

*(2)* $\rho(T) < 1$

*(3)* $\|T\| < 1$ *pentru cel puţin o normă matricială.*

*Demonstraţie.*

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

(2.7.3) convergentă $\Leftrightarrow$ $I - T$ inversabilă $\Leftrightarrow$ $\rho(T) < 1 \Leftrightarrow \exists \|\cdot\|$ astfel încât $\|T\| < 1$ (din teorema 2.1.8). □

Aplicând teorema de punct fix a lui Banach se obţine:

**Teorema 2.7.3** *Dacă există* $\|\cdot\|$ *astfel încât* $\|T\| < 1$, *şirul* $(x^{(k)})$ *definit de (2.7.3) este convergent pentru orice* $x^{(0)} \in \mathbb{R}^n$ *şi are loc*

$$\|x^* - x^{(k)}\| \leq \frac{\|T\|^k}{1 - \|T\|}\|x^{(1)} - x^{(0)}\| \leq \frac{\|T\|}{1 - \|T\|}\|x^{(1)} - x^{(0)}||. \tag{2.7.4}$$

O metodă iterativă de rezolvare a unui sistem algebric liniar $Ax = b$ porneşte de la o aproximaţie iniţială $x^{(0)} \in \mathbb{R}^n(\mathbb{C}^n)$ şi generează un şir de vectori $\{x^{(k)}\}$ care converge către soluţia $x^*$ a sistemului. Aceste tehnici transformă sistemul iniţial într-un sistem echivalent de forma $x = Tx + c$, $T \in \mathbb{K}^{n\times n}$, $c \in \mathbb{K}^n$. Se generează un şir de forma $x^{(k)} = Tx^{(k-1)} + c$. Criteriul de oprire este

$$\|x^{(k)} - x^{(k-1)}\| \leq \frac{1 - \|T\|}{\|T\|}\varepsilon. \tag{2.7.5}$$

El are la bază rezultatul următor:

**Propoziţia 2.7.4** *Dacă* $x^*$ *este soluţia sistemului* (2.7.2), *cu* $\|T\| < 1$, *atunci*

$$\|x^* - x^{(k)}\| \leq \frac{\|T\|}{1 - \|T\|}\|x^{(k)} - x^{(k-1)}\|. \tag{2.7.6}$$

*Demonstraţie.* Fie $p \in \mathbb{N}^*$. Avem

$$\|x^{(k+p)} - x^{(k)}\| \leq \|x^{(k+1)} - x^{(k)}\| + \cdots + \|x^{(k+p)} - x^{(k+p-1)}\|. \tag{2.7.7}$$

Pe de altă parte, din (2.7.3), rezultă că

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

sau, pentru $k < m$

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

Aplicând aceste inegalităţi, pentru $m = \overline{k, k+p-1}$ relaţia (2.7.6) devine

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

Deoarece $\|T\| < 1$, avem

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

din care prin trecere la limită în raport cu $p$ se obţine (2.7.6). □

Dacă $\|T\| \leq 1/2$, inegalitatea (2.7.6) devine

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

iar criteriul de oprire

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

Tehnicile iterative sunt rar utilizate pentru a rezolva sisteme de mici dimensiuni, deoarece timpul necesar pentru a obţine precizia dorită depăşeşte timpul necesar pentru eliminarea gaussiană. Pentru sisteme rare (a căror matrice are multe zerouri), de mari dimensiuni, metodele iterative sunt eficiente şi din punct de vedere al spaţiului şi din punct de vedere al timpului.

Fie sistemul $Ax = b$. Presupunem că putem scrie matricea inversabilă $A$ sub forma $A = M - N$. Dacă $M$ este uşor de inversat (diagonală, triunghiulară, etc.) este mai convenabil să procedăm astfel:

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

Ultima ecuaţie are forma $x = Tx + c$, cu $T = M^{-1}N = I - M^{-1}A$. Se obţine şirul

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b, \quad k \in \mathbb{N},$$

$x^{(0)}$ vector arbitrar.

Prima descompunere pe care o considerăm este $A = D - L - U$, unde

$$(D)_{ij} = a_{ij}\delta_{ij}, \quad (-L)_{ij} = \begin{cases} a_{ij}, & i > j \\ 0, & \text{altfel} \end{cases}$$

$$(-U)_{ij} = \begin{cases} a_{ij}, & i < j \\ 0, & \text{altfel} \end{cases}$$

Se ia $M = D$, $N = L + U$. Se obţine succesiv

$$Ax = b \Leftrightarrow Dx = (L + U)x + b \Leftrightarrow x = D^{-1}(L + U)x + D^{-1}b$$

Deci $T = T_J = D^{-1}(L + U)$, $c = c_J = D^{-1}b$. Metoda se numeşte *metoda lui Jacobi* [6] ($D$ inversabilă, de ce?)

Altă descompunere este $A = D - L - U$, $M = D - L$, $N = U$. Se obţine $T_{GS} = (D - L)^{-1}U$, $c_{GS} = (D - L)^{-1}b$ – numită *metoda Gauss-Seidel* ($D - L$ inversabilă, de ce?)

Să examinăm iteraţiile Jacobi $x_i^{(k)} = \dfrac{1}{a_{ii}} \left( b_i - \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k-1)} \right)$.

La calculul lui $x_i^{(k)}$ se utilizează componentele lui $x^{(k-1)}$ (substituţia simultană). Deoarece pentru $i > 1$, $x_1^{(k)}, \ldots, x_{i-1}^{(k)}$ au fost deja calculaţi şi se presupune că sunt aproximaţii mai bune ale componentelor soluţiei decât $x_1^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$ pare rezonabil să calculăm $x_i^{(k)}$ utilizând valorile cele mai recente, adică

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

Se pot da condiţii necesare şi suficiente pentru convergenţa metodei lui Jacobi şi a metodei Gauss-Seidel

$$\rho(T_J) < 1$$

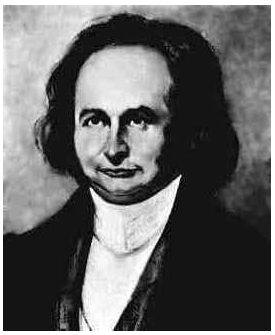
$$\rho(T_{GS}) < 1$$

şi condiţii suficiente: pentru o normă dată avem

$$\|T_J\| < 1$$

$$\|T_{GS}\| < 1.$$

6



Carl Gustav Jacob Jacobi (1804-1851) a fost contemporan al lui Gauss, şi unul dintre cei mai importanţi matematicieni germani din secolul al XIX-lea. Numele său este legat de funcţii eliptice, ecuaţiile cu derivate parţiale ale dinamicii, mecanică cerească. Matricea derivatelor parţiale poartă de asemenea numele său. Este inventatorul metodei iterative de rezolvare a sistemelor algebrice liniare numită metoda lui Jacobi şi pe care a aplicat-o în mecanica cerească.

Putem îmbunătăţi metoda Gauss-Seidel introducând un parametru $\omega$ şi alegând

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

Avem

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

iar iteraţia obţinută este

$$\left(\frac{D}{\omega} - L\right) x^{(k+1)} = \left(\frac{1-\omega}{\omega}D + U\right) x^{(k)} + b$$

Se obţine matricea

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

Metoda se numeşte *metoda relaxării* . Avem următoarele variante:

– $\omega > 1$ suprarelaxare (SOR - Successive Over Relaxation);

– $\omega < 1$ subrelaxare;

– $\omega = 1$ Gauss-Seidel.

Următoarele două teoreme se referă la convergenţa metodei relaxării. Scriem matricea metodei sub forma

$$T_\omega = (I - \omega\bar{L})^{-1}[(1 - \omega\bar{L})I + \omega\bar{U}],$$

unde $\bar{L} = D^{-1}L$ şi $\bar{U} = D^{-1}U$.

**Teorema 2.7.5 (Kahan)** *Are loc $\rho(T_\omega) \geq |\omega - 1|$. De aici rezultă condiţia necesară $0 < \omega < 2$.*

*Demonstraţie.* Scriem polinomul caracteristic al lui $T_\omega$ sub forma $p(\lambda) = \det(\lambda I - T_\omega) = \det((I - \omega\bar{L})(\lambda I - T_\omega)) = \det((\lambda + \omega - 1)I - \omega\lambda\bar{L} - \omega\bar{U})$; deci avem

$$p(0) = \pm \prod_{i=1}^{n} \lambda_i(T_\omega) = \pm \det((\omega - 1)I) = \pm(\omega - 1)^n,$$

ceea ce implică $\max_i |\lambda_i(T_\omega)| \geq |\omega - 1|$. □

**Teorema 2.7.6 (Ostrowski-Reich)** *Dacă $A$ este o matrice pozitiv definită şi $0 < \omega < 2$, SOR converge pentru orice alegere a aproximaţiei iniţiale $x^{(0)}$.*

*Demonstraţie.* Demonstraţia se face în doi paşi. Fie $M = \omega^{-1}(D - \omega L)$. Atunci:

(1) definim $Q = A^{-1}(2M - A)$ şi arătăm ca $\mathrm{Re}\lambda_i(Q) > 0$, pentru orice $i$.

(2) arătăm că $T_\omega = (Q - I)(Q + I)$, de unde rezultă că $|\lambda_i(T_\omega)| < 1$, pentru orice $i$.

Pentru primul pas, observăm că $Qx = \lambda x$ implică $(2M - A)x = \lambda Ax$. Adunând această ecuaţie cu transpusa ei conjugată se obţine $x^*(M + M^* - A)x = \mathrm{Re}\lambda(x^*Ax)$. Astfel $\mathrm{Re}\lambda = x^*(M+M^*-A)x/x*Ax = x^*(\frac{2}{\omega}-1)Dx/x^*Ax > 0$, căci $A$ şi $(\frac{2}{\omega}-1)D$ sunt pozitiv definite.

Pentru a demonstra (2), observăm că

$$(Q - I)(Q + I)^{-1} = (2A^{-1}M - 2I)(2A^{-1}M)^{-1} = I - M^{-1}A = T_\omega,$$

deci

$$|\lambda(T_\omega)| = \left|\frac{\lambda(Q) - 1}{\lambda(Q) + 1}\right| = \left|\frac{(\mathrm{Re}\lambda(Q) - 1)^2 + (\mathrm{Im}\lambda(Q))^2}{(\mathrm{Re}\lambda(Q) + 1)^2 + (\mathrm{Im}\lambda(Q))^2}\right|^{1/2}.$$

□

Teoremele 2.7.5 şi 2.7.6 ne dau o condiţie necesară şi suficientă pentru convergenţa metodei relaxării dacă matricea sistemului este simetrică şi pozitiv definită: $0 < \omega < 2$.

**Observaţia 2.7.7.** Pentru metoda Jacobi (şi Gauss-Seidel) o condiţie suficientă de convergenţă este

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \quad (A \text{ diagonal dominantă pe linii}).$$

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ji}|, \quad (A \text{ diagonal dominantă pe coloane}.) \qquad \diamondsuit$$

Valoarea optimală pentru $\omega$ este

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

Ea este valabilă doar pentru anumite tipuri de matrice (de exemplu tridiagonale pe blocuri).

**Teorema 2.7.8** *Fie $B$ o matrice pătratică şi $\|\cdot\|$ o normă matricială oarecare. Atunci*

$$\lim_{k\to\infty} \|B^k\|^{1/k} = \rho(B).$$

*Demonstraţie.* Cum $\rho(B) \leq \|B\|$ (teorema 2.1.6 şi cum $\rho(B) = (\rho(B^k))^{1/k}$ avem

$$\rho(B) \leq \|B^k\|^{1/k},\ \forall\, k \in \mathbb{N}^*.$$

Se va stabili că, pentru orice $\varepsilon > 0$ există $\ell = \ell(\varepsilon) \in \mathbb{N}$ astfel încât

$$k \geq \ell \Rightarrow \|B^k\|^{1/k} \leq \rho(B) + \varepsilon,$$

ceea ce demonstrează relaţia. Fie deci $\varepsilon > 0$ dat. Deoarece matricea

$$B_\varepsilon = \frac{B}{\rho(B) + \varepsilon}$$

verifică $\rho(B_\varepsilon) < 1$ se deduce din teorema 2.7.2 că $\lim_{k\to\infty} B_\varepsilon^k = 0$. Prin urmare, există un întreg $\ell = \ell_\varepsilon$ astfel încât

$$k \geq \ell \Rightarrow \|B_\varepsilon^k\| = \frac{\|B\|^k}{(\rho(B) + \varepsilon)^k} \leq 1$$

ceea ce este chiar relaţia căutată. □

Dăm o teoremă referitoare la matrice de forma $(I + B)$.

**Teorema 2.7.9** *(1) Fie $\|\cdot\|$ o normă matricială subordonată şi $B$ o matrice ce verifică*

$$\|B\| < 1.$$

*Atunci matricea $I + B$ este inversabilă şi $\|I + B\|^{-1} \leq \frac{1}{1-\|B\|}$.*

*(2) Dacă o matrice de forma $(I + B)$ este singulară, atunci în mod necesar*

$$\|B\| \geq 1$$

*pentru orice normă matricială subordonată sau nu.*

*Demonstraţie.* (1) Deoarece

$$(I + B)u = 0 \Rightarrow \|u\| = \|Bu\|$$

$$\|B\| < 1 \wedge u \neq 0 \Rightarrow \|Bu\| < \|u\|,$$

pentru norma vectorială corespondentă, se deduce că

$$(I + B)u = 0 \Rightarrow u = 0.$$

Matricea $I + B$ fiind inversabilă putem scrie

$$(I + B)^{-1} = I - B(I + B)^{-1}$$

de unde

$$\|(I + B)^{-1}\| \leq 1 + \|B\|\|(I + B)^{-1}\|,$$

ceea ce conduce la inegalitatea căutată.

(2) A spune că matricea $(I + B)$ este singulară revine la a spune că $-1$ este valoare proprie a lui $B$. În aceste condiţii aplicarea teoremei 2.1.6 ne arată că $\|B\| \geq \rho(B) \geq 1$. □

Cum se alege între mai multe metode iterative convergente pentru rezolvarea aceluiaşi sistem liniar $Au = b$? Pentru a fixa ideile, presupunem că $B$ este normală. Atunci

$$\|B^k e_0\|_2 \leq \|B^k\|_2 \|e_0\|_2 = (\rho(B))^k \|e_0\|_2$$

şi această inegalitate este cea mai bună posibilă, în sensul că, pentru orice întreg $k \geq 0$, există un vector $e_0(k) \neq 0$ pentru care ea devine egalitate.

În cazul matricelor normale, metoda este cu atât mai rapidă, cu cât $\rho(B)$ este mai mic, căci

$$\sup_{\|e_0\|_2=1} \|B^k e_0\|_2^{1/k} = \rho(B) \text{ pentru orice } k \geq 0.$$

În cazul general (matricea $B$ oarecare, normă vectorială oarecare), concluzia este identică: asimptotic, vectorul de eroare $e_k = B^k e_0$ se comportă ca şi $(\rho(B))^k$, cum precizează rezultatul următor.

**Teorema 2.7.10** *(1) Fie $\|\cdot\|$ o normă vectorială oarecare şi $u$ astfel încât*

$$u = Bu + c.$$

*Se consideră metoda iterativă*

$$u_{k+1} = Bu_k + c, \quad k \geq 0.$$

*Atunci*

$$\lim_{k\to\infty} \left\{ \sup_{\|u_0-u\|=1} \|u_k - u\|^{1/k} \right\} = \rho(B).$$

*(2) Fie $\|\cdot\|$ o normă vectorială oarecare şi fie $u$ astfel încât*

$$u = Bu + c = \widetilde{B}u + \widetilde{c}.$$

*Se consideră metodele iterative*

$$\widetilde{u}_{k+1} = \widetilde{B}\widetilde{u}_k + \widetilde{c}, \quad k \geq 0, \quad u_{k+1} = Bu_k + c, \quad k \geq 0$$

*cu* $\rho(B) < \rho(\widetilde{B})$, $u_0 = \widetilde{u}_0$. *Atunci, oricare ar fi numărul* $\varepsilon > 0$ *există un întreg* $\ell(\varepsilon)$ *astfel încât*

$$k \geq \ell \Rightarrow \sup_{\|u_0 - u\| = 1} \left\{ \frac{\|\widetilde{u}_k - u\|}{\|u_k - u\|} \right\}^{1/k} \geq \frac{\rho(\widetilde{B})}{\rho(B) + \varepsilon}.$$

*Demonstraţie.* Fie $\|\cdot\|$ norma matricială subordonată. Pentru orice $k$ natural putem scrie

$$(\rho(B))^k = \rho(B^k) \leq \|B^k\| = \sup_{\|e_0\|=1} \|B^k e_0\|$$

astfel ca

$$\rho(B) \leq \sup_{\|e_0\|=1} \|B^k e_0\|^{1/k} = \|B^k\|^{1/k}$$

şi aserţiunea (1) decurge din teorema 2.7.8. Conform aceleeaşi teoreme, fiind dat un $\varepsilon > 0$ există un întreg $\ell(\varepsilon)$ astfel încât

$$k \geq \ell \Rightarrow \sup_{\|e_0\|=1} \|B^k e_0\|^{1/k} \leq (\rho(B) + \varepsilon).$$

Prin urmare, pentru orice $k \geq \ell$, există un vector $e_0 = e_0(k)$ astfel încât

$$\|e_0\| = 1 \quad \text{şi} \quad \|\widetilde{B}^k e_0\|^{1/k} = \|\widetilde{B}^k\|^{1/k} \geq \rho(\widetilde{B})$$

şi (2) este demonstrată. □

Deci studiul metodelor iterative răspunde la două probleme.

(1) Fiind dată o metodă iterativă cu matricea $B$, să se determine dacă metoda este convergentă, adică dacă $\rho(B) < 1$, sau echivalent, dacă există o normă matricială astfel ca $\|B\| < 1$.

(2) Fiind date două metode iterative convergente, metoda iterativă cea mai rapidă este cea a cărei matrice are cea mai mică rază spectrală.

# CAPITOLUL 3

# Aproximarea funcţiilor

## Cuprins

Capitolul prezent este legat de aproximarea funcţiilor. Funcţiile în cauză pot să fie definite pe un continuu - de regulă un interval - sau pe o mulţime finită de puncte. Prima situaţie apare în contextul funcţiilor speciale (elementare sau transcendente) pe care dorim să le evaluăm ca parte a unei subrutine. Deoarece o astfel de evaluare trebuie să se reducă la un număr finit de operaţii aritmetice, trebuie în ultimă instanţă să aproximăm funcţiile prin intermediul polinoamelor sau funcţiilor raţionale. A doua situaţie este întâlnită în ştiinţele fizice, când măsurătorile unor cantităţi fizice se fac în funcţie de alte cantităţi (cum ar fi timpul). În ambele cazuri dorim să aproximăm o funcţie dată, cât mai bine posibil, în termeni de funcţii mai simple.

În general o schemă de aproximare poate fi descrisă după cum urmează.

Se dă o funcţie $f \in X$ ce urmează a fi aproximată, împreună cu o clasă $\Phi$ de aproximante şi o normă $\|\cdot\|$ ce măsoară mărimea funcţiilor. Căutăm o aproximare $\widehat{\varphi} \in \Phi$ a lui $f$ astfel încât

$$\|f - \widehat{\varphi}\| \leq \|f - \varphi\| \text{ pentru orice } \varphi \in \Phi. \tag{3.0.1}$$

Această problemă se numeşte *problemă de cea mai bună aproximare* a lui $f$ cu elemente din $\Phi$, iar funcţia $\widehat{\varphi}$ se numeşte *(element de) cea mai bună aproximare* a lui $f$ relativ la norma $\|\cdot\|$.

Cunoscându-se o bază $\{\pi_j\}_{j=1}^n$ a lui $\Phi$ putem scrie

$$\Phi = \Phi_n = \left\{ \varphi : \ \varphi(t) = \sum_{j=1}^{n} c_j \pi_j(t), \ c_j \in \mathbb{R} \right\}. \tag{3.0.2}$$

$\Phi$ este un spaţiu liniar finit dimensional sau o submulţime a acestuia.

**Exemplul 3.0.11.** $\Phi = \mathbb{P}_m$ - mulţimea polinoamelor de grad cel mult $m$. O bază a sa este $e_j(t) = t^j$, $j = 0, 1, \ldots, m$. Deci $\dim \mathbb{P}_m = m + 1$. Polinoamele sunt cele mai utilizate aproximante pentru funcţii pe domenii mărginite (intervale sau mulţimi finite). Motivul – teorema lui Weierstrass – orice funcţie din $C[a, b]$ poate fi aproximată oricât de bine printr-un polinom de grad suficient de mare. $\diamondsuit$

**Exemplul 3.0.12.** $\Phi = \mathbb{S}_m^k(\Delta)$ spaţiul funcţiilor spline polinomiale de grad $m$ şi 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 funcţii polinomiale pe porţiuni de grad $\leq m$, racordate în $t_1, \ldots, t_{N-1}$, astfel încât toate derivatele până la ordinul $k$ să fie continue pe $[a, b]$. Presupunem $0 \leq k < m$. Pentru $k = m$ se obţine $\mathbb{P}_m$. Dacă $k = -1$ permitem discontinuităţi în punctele de joncţiune. $\diamondsuit$

**Exemplul 3.0.13.** $\Phi = \mathbb{T}_m[0, 2\pi]$ spaţiul polinoamelor trigonometrice de grad cel mult $m$ pe $[0, 2\pi]$. Acestea sunt combinaţii liniare ale funcţiilor

$$\pi_k(t) = \cos(k-1)t \quad k = \overline{1, m+1},$$
$$\pi_{m+1-k}(t) = \sin kt \quad k = \overline{1, m}.$$

Dimensiunea spaţiului este $n = 2m + 1$. Astfel de aproximante sunt alegeri naturale dacă funcţia de aproximat este periodică de perioadă $2\pi$. (Dacă $f$ are perioada $p$ se face schimbarea de variabilă $t \to tp/2\pi$.) $\diamondsuit$

De notat că mulţimea funcţiilor raţionale

$$\Phi = \mathbb{R}_{r,s} = \{\varphi : \ \varphi = p/q, p \in \mathbb{P}_r, q \in \mathbb{P}_s\},$$

nu este spaţiu liniar.

Câteva alegeri posibile ale normei, atât pentru funcţii continue, cât şi pentru cele discrete apar în tabelul 3.1. Cazul continuu presupune un interval $[a, b]$ şi o funcţie pondere $w(t)$ (posibil şi $w(t) \equiv 1$) definită pe intervalul $[a, b]$ şi pozitivă, exceptând zerourile izolate. Cazul discret presupune o mulţime de $N$ puncte distincte $t_1, t_2, \ldots, t_N$ împreună cu ponderile $w_1, w_2, \ldots, w_N$ (posibil $w_i = 1$, $i = \overline{1, N}$). Intervalul $[a, b]$ poate fi nemărginit, dacă funcţia pondere $w$ este astfel încât integrala pe $[a, b]$ care defineşte norma să aibă sens.

| normă continuă | tip | normă discretă |
|---|---|---|
| $\|u\|_\infty = \max\limits_{a\le t\le b} \|u(t)\|$ | $L^\infty$ | $\|u\|_\infty = \max\limits_{1\le i\le N} \|u(t_i)\|$ |
| $\|u\|_1 = \int_a^b \|u(t)\|dt$ | $L^1$ | $\|u\|_1 = \sum_{i=1}^N \|u(t_i)\|$ |
| $\|u\|_{1,w} = \int_a^b \|u(t)\|w(t)dt$ | $L^1_w$ | $\|u\|_{1,w} = \sum_{i=1}^n w_i\|u(t_i)\|$ |
| $\|u\|_{2,w} = \left(\int_a^b \|u(t)\|^2 w(t)dt\right)^{1/2}$ | $L^2_w$ | $\|u\|_{2,w} = \left(\sum_{i=1}^N w_i\|u(t_i)\|^2\right)^{1/2}$ |

Tabela 3.1: Tipuri de aproximare şi normele asociate

Deci combinând normele din tabelă cu spaţiile liniare din exemple se obţine o problemă de cea mai bună aproximare (3.0.1) cu sens. În cazul continuu, funcţia dată $f$ şi funcţiile $\varphi$ din clasa $\Phi$ trebuie definită pe $[a, b]$ şi norma $\|f - \varphi\|$ să aibă sens. La fel, $f$ şi $\varphi$ trebuie definite în punctele $t_i$ în cazul discret.

De notat că dacă cea mai bună aproximantă $\widehat{\varphi}$ în cazul discret este astfel încât $\|f - \widehat{\varphi}\| = 0$, atunci $\widehat{\varphi}(t_i) = f(t_i)$, pentru $i = 1, 2, \ldots, N$. Spunem că $\widehat{\varphi}$ interpolează $f$ în punctele $t_i$ şi numim această problemă de aproximare *problemă de interpolare* .

Cele mai simple probleme de aproximare sunt problema celor mai mici pătrate (vezi secţiunea 3.1) şi problema de interpolare, iar spaţiul cel mai simplu este cel al polinoamelor.

Înainte de a începe cu problema celor mai mici pătrate, introducem un instrument notaţional care ne permite să tratăm cazul continuu şi cel discret simultan. Definim în cazul continuu

$$\lambda(t) = \begin{cases} 0, & \text{dacă} \quad t < a \ (\text{când } -\infty < a), \\ \int_a^t w(\tau)d\tau, & \text{dacă} \quad a \le t \le b, \\ \int_a^b w(\tau)d\tau, & \text{dacă} \quad t > b \ (\text{când } b < \infty). \end{cases} \tag{3.0.3}$$

Astfel putem scrie, pentru orice funcţie continuă $u$

$$\int_{\mathbb{R}} u(t)d\lambda(t) = \int_a^b u(t)w(t)dt, \tag{3.0.4}$$

căci $d\lambda(t) \equiv 0$ în afara lui $[a, b]$ şi $d\lambda(t) = w(t)dt$ în interiorul lui. Vom numi $d\lambda$ *măsură* (pozitivă) *continuă*. *Măsura discretă* (numită şi „măsura Dirac“) asociată mulţimii de puncte $\{t_1, t_2, \ldots, t_N\}$ este o măsură $d\lambda$ care este nenulă numai în punctele $t_i$ şi are aici valoarea $w_i$. Astfel în acest caz

$$\int_{\mathbb{R}} u(t)d\lambda(t) = \sum_{i=1}^{N} w_i u(t_i). \tag{3.0.5}$$

O definiţie mai precisă se poate da cu ajutorul integralei Stieltjes, dacă definim $\lambda(t)$ ca fiind o funcţie în scară cu saltul în $t_i$ egal cu $w_i$. În particular, definim norma lui $L^2$ prin

$$\|u\|_{2,d_\lambda} = \left( \int_{\mathbb{R}} |u(t)|^2 d\lambda(t) \right)^{\frac{1}{2}} \tag{3.0.6}$$

şi obţinem norma continuă sau discretă după cum $\lambda$ este ca în (3.0.3) sau o funcţie în scară ca în (3.0.5).

Vom numi *suportul* lui $d\lambda$ – notat cu $\mathrm{supp}d\lambda$ – intervalul $[a, b]$ în cazul continuu (presupunem că $w$ este pozitivă pe $[a, b]$ exceptând zerourile izolate) şi mulţimea $\{t_1, t_2, \ldots, t_N\}$ în cazul discret. Spunem că mulţimea de funcţii $\pi_j$ din (3.0.2) este liniar independentă pe $\mathrm{supp}d\lambda$ dacă

$$\forall\, t \ \in \mathrm{supp}d\lambda \quad \sum_{j=1}^{n} c_j \pi_j(t) \equiv 0 \Rightarrow c_1 = c_2 = \cdots = c_k = 0. \tag{3.0.7}$$

## 3.1. Aproximaţie prin metoda celor mai mici pătrate

Vom particulariza problema (3.0.1) luând ca normă norma din $L^2$

$$\|u\|_{2,d_\lambda} = \left( \int_{\mathbb{R}} |u(t)|^2 d\lambda(t) \right)^{\frac{1}{2}}, \tag{3.1.1}$$

unde $d\lambda$ este fie o măsură continuă (conform (3.0.3)) sau discretă (conform (3.0.5)) şi utilizând aproximanta $\varphi$ dintr-un spaţiu liniar $n$-dimensional

$$\Phi = \Phi_n = \left\{ \varphi : \ \varphi(t) = \sum_{j=1}^{n} c_j \pi_j(t), \ c_j \in \mathbb{R} \right\}. \tag{3.1.2}$$

Presupunem că funcţiile $\pi_j$ sunt liniar independente pe $\mathrm{supp} d\lambda$ şi că integrala din (3.1.1) are sens pentru $u = \pi_j$ sau $u = f$.

Problema astfel obţinută se numeşte *problemă de aproximare în sensul celor mai mici pătrate* sau *problemă de aproximare în medie pătratică*. Soluţia ei a fost dată la începutul secolului al XIX-lea de către Gauss şi Legendre [1].

Presupunem că funcţiile de bază $\pi_j$ sunt liniar independente pe $\mathrm{supp} d\lambda$. Vom presupune că integrala din (3.1.1) are sens pentru $u = \pi_j$, $j = 1, \ldots, n$ şi $u = f$.

Soluţia problemei de cea mai bună aproximare se exprimă mai uşor cu ajutorul produselor scalare.

## 3.1.1. Produse scalare

Dându-se o măsură continuă sau discretă $d\lambda$ şi două funcţii $u$ şi $v$ având norma finită putem defini produsul scalar

$$(u, v) = \int_{\mathbb{R}} u(t) v(t) d\lambda(t). \tag{3.1.3}$$

Inegalitatea Cauchy-Buniakovski-Schwarz

$$\|(u, v)\| \leq \|u\|_{2, d_\lambda} \|v\|_{2, d_\lambda}$$

ne spune că integrala în (3.1.3) este bine definită.

Produsul scalar real are următoarele proprietăţi utile:

(i) simetria $(u, v) = (v, u)$;

(ii) omogenitatea $(\alpha u, v) = \alpha (u, v)$, $\alpha \in \mathbb{R}$;

[1] Adrien Marie Legendre (1752-1833) matematician francez, bine cunoscut pentru tratatul său asupra integralelor eliptice, dar şi pentru lucrările sale de teoria numerelor şi geometrie. Este considerat, alături de Gauss, iniţiator (în 1805) al metodei celor mai mici pătrate, deşi Gauss a utilizat metoda încă din 1794, dar a publicat-o doar în 1809.

(iii) aditivitatea $(u+v,w)=(u,w)+(v,w)$;

(iv) pozitiv definirea $(u,u)\geq 0$ şi $(u,u)=0 \Leftrightarrow u\equiv 0$ pe $\mathrm{supp}d\lambda$.

Omogenitatea şi aditivitatea ne dau liniaritatea

$$(\alpha_1 u_1+\alpha_2 u_2, v)=\alpha_1(u_1,v)+\alpha_2(u_2,v) \tag{3.1.4}$$

Relaţia (3.1.4) se poate extinde la combinaţii liniare finite. De asemenea

$$\|u\|^2_{2,d_\lambda}=(u,u). \tag{3.1.5}$$

Spunem că $u$ şi $v$ sunt *ortogonale* dacă

$$(u,v)=0. \tag{3.1.6}$$

Mai general, putem considera sisteme ortogonale $\{u_k\}_{k=1}^n$:

$$(u_i,u_j)=0 \text{ dacă } i\neq j,\ u_k\neq 0 \text{ pe } \mathrm{supp}d\lambda;\ i,j=\overline{1,n},\ k=\overline{1,n}. \tag{3.1.7}$$

Pentru un astfel de sistem are loc teorema generalizată a lui Pitagora

$$\left\|\sum_{k=1}^n \alpha_k u_k\right\|^2=\sum_{k=1}^n |\alpha_k|^2\|u_k\|^2. \tag{3.1.8}$$

O consecinţă importantă a lui (3.1.8) este aceea că orice sistem ortogonal este liniar independent pe $\mathrm{supp}d\lambda$. Într-adevăr, dacă membrul stâng al lui (3.1.8) se anulează, atunci şi membrul drept se anulează şi deoarece $\|u_k\|^2>0$, din ipoteză rezultă $\alpha_1=\alpha_2=\cdots=\alpha_n=0$.

## 3.1.2. Ecuaţiile normale

Suntem acum în măsură să rezolvăm problema de aproximare în sensul celor mai mici pătrate. Din (3.1.5) 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 din (3.1.2) se obţine

$$E^2[\varphi]=\int_{\mathbb{R}}\left(\sum_{j=1}^n c_j\pi_j(t)\right)^2 d\lambda(t)-2\int_{\mathbb{R}}\left(\sum_{j=1}^n c_j\pi_j(t)\right)f(t)d\lambda(t)+ \tag{3.1.9}$$
$$+\int_{\mathbb{R}} f^2(t)d\lambda(t).$$

Pătratul erorii din $L^2$ este o funcţie cuadratică de coeficienţii $c_1, \ldots, c_n$ ai lui $\varphi$. Problema celei mai bune aproximaţii în $L^2$ revine la a minimiza această funcţie pătratică; ea se rezolvă anulând derivatele parţiale. Se obţine

$$\frac{\partial}{\partial c_i} E^2[\varphi] = 2 \int_{\mathbb{R}} \left( \sum_{j=1}^{n} c_j \pi_j(t) \right) \pi_i(t) d\lambda(t) - 2 \int_{\mathbb{R}} \pi_i(t) f(t) d\lambda(t) = 0$$

adică

$$\sum_{j=1}^{n} (\pi_i, \pi_j) c_j = (\pi_i, f), \quad i = 1, 2, \ldots, n. \tag{3.1.10}$$

Aceste ecuaţii se numesc *ecuaţii normale* pentru problema celor mai mici pătrate. Ele formează un sistem liniar de forma

$$Ac = b \tag{3.1.11}$$

unde matricea $A$ şi vectorul $b$ au elementele

$$A = [a_{ij}], \quad a_{ij} = (\pi_i, \pi_j), \quad b = [b_i], \quad b_i = (\pi_i, f). \tag{3.1.12}$$

Datorită simetriei produsului scalar, $A$ este o matrice simetrică. Mai mult, $A$ este pozitiv definită, adică

$$x^T A x = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j > 0 \text{ dacă } x \neq [0, 0, \ldots, 0]^T. \tag{3.1.13}$$

Funcţia (3.1.13) se numeşte *formă pătratică* (deoarece este omogenă de grad 2). Pozitiv definirea lui $A$ ne spune că forma pătratică ai cărei coeficienţi sunt elementele lui $A$ este întotdeauna nenegativă şi zero numai dacă variabilele $x_i$ se anulează.

Pentru a demonstra (3.1.13) să inserăm definiţia lui $a_{ij}$ şi să utilizăm proprietăţile (i)-(iv) ale produsului scalar

$$x^T A x = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j (\pi_i, \pi_j) = \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i \pi_i, x_j \pi_j) = \left\| \sum_{i=1}^{n} x_i \pi_i \right\|^2 .$$

Aceasta este evident nenegativă. Ea este zero numai dacă $\sum_{i=1}^{n} x_i \pi_i \equiv 0$ pe supp$d\lambda$, care pe baza liniar independenţei lui $\pi_i$ implică $x_1 = x_2 = \cdots = x_n = 0$.

Este un rezultat cunoscut din algebra liniară că o matrice $A$ simetrică pozitiv definită este nesingulară. Într-adevăr, determinantul său, precum şi minorii principali sunt strict pozitivi. Rezultă că sistemul de ecuaţii normale (3.1.10) are soluţie unică. Corespunde această soluţie minimului lui $E[\varphi]$ în (3.1.9)? 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 funcţie cuadratică. De aceea, $H$, ca şi $A$, este într-adevăr pozitiv definită şi soluţia ecuaţiilor normale ne dă

minimul dorit. Problema de aproximare în sensul celor mai mici pătrate are o soluţie unică, dată de

$$\widehat{\varphi}(t) = \sum_{j=1}^{n} \widehat{c}_j \pi_j(t) \tag{3.1.14}$$

unde $\widehat{c} = [\widehat{c}_1, \widehat{c}_2, \ldots, \widehat{c}_n]^T$ este vectorul soluţie al ecuaţiilor normale (3.1.10). Aceasta rezolvă problema de aproximare în sensul celor mai mici pătrate complet în teorie, dar nu şi în practică. Referitor la o mulţime generală de funcţii de bază liniar independente, pot apărea următoarele dificultăţi:

(1) Sistemul de ecuaţii normale (3.1.10) poate fi prost condiţionat. Un exemplu simplu este următorul: $\mathrm{supp} d\lambda = [0,1]$, $d\lambda(t) = dt$ pe $[0,1]$ şi $\pi_j(t) = t^{j-1}$, $j = 1, 2, \ldots, n$. Atunci

$$(\pi_i, \pi_j) = \int_0^1 t^{i+j-2} dt = \frac{1}{i+j-1}, \quad i, j = 1, 2, \ldots, n,$$

adică matricea $A$ este matricea Hilbert. Prost condiţionarea ecuaţiilor normale se datorează alegerii neinspirate a funcţiilor de bază. Acestea devin aproape liniar dependente când exponentul creşte. 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 funcţie discontinuă. Un polinom care oscilează mai rapid pe $[0,1]$ ar fi de preferat, căci ar angaja mai viguros funcţia $f$.

(2) Al doilea dezavantaj este faptul că toţi coeficienţii $\widehat{c}_j$ din (3.1.14) depind de $n$, adică $\widehat{c}_j = \widehat{c}_j^{(n)}$, $j = 1, 2, \ldots, n$. Mărirea lui $n$ ne dă un nou sistem de ecuaţii mai mare şi cu o soluţie complet diferită. Acest fenomen se numeşte *nepermanenţa coeficienţilor* $\widehat{c}_j$.

Amândouă neajunsurile (1) şi (2) pot fi eliminate (sau măcar atenuate) alegând ca funcţii de bază un sistem ortogonal,

$$(\pi_i, \pi_j) = 0 \text{ dacă } i \neq j \quad (\pi_i, \pi_j) = \|\pi_j\|^2 > 0 \tag{3.1.15}$$

Atunci sistemul de ecuaţii normale devine diagonal şi poate fi rezolvat imediat cu formula

$$\widehat{c}_j = \frac{(\pi_j, f)}{(\pi_i, \pi_j)}, \quad j = 1, 2, \ldots, n. \tag{3.1.16}$$

Evident, aceşti coeficienţi $\widehat{c}_j$ sunt independenţi de $n$ şi odată calculaţi rămân la fel pentru orice $n$ mai mare. Avem acum proprietatea de permanenţă a coeficienţilor. De asemenea nu trebuie să rezolvăm sistemul de ecuaţii normale, ci putem aplica direct (3.1.16). Aceasta nu înseamnă că nu sunt probleme numerice în (3.1.16). Într-adevăr, numitorul $\|\pi_j\|^2$ creşte odată cu $j$, în timp ce integrala de la numărător (sau termenii individuali în cazul discret) au acelaşi ordin de mărime ca şi $f$. De aceea ne aşteptăm ca valorile $\widehat{c}_j$ ale coeficienţilor să descrească rapid. Din acest motiv pot să apară erori de anulare atunci

când se calculează produsul scalar de la numărător. Problemele de anulare pot fi ocolite într-o oarecare măsură calculând $\widehat{c}_j$ în forma alternativă

$$\widehat{c}_j = \frac{1}{(\pi_j, \pi_j)} \left( f - \sum_{k=1}^{j-1} c_k \pi_k, \pi_j \right), \quad j =, 1, 2, \ldots, n, \tag{3.1.17}$$

unde suma vidă (când $j = 1$) se ia egală cu zero. Evident din ortogonalitatea lui $\pi_j$, (3.1.17) este echivalentă cu (3.1.16) din punct de vedere matematic, dar nu neapărat şi numeric.

Dăm un algoritm care calculează $\widehat{c}_j$ cu (3.1.17), dar în acelaşi timp şi $\widehat{\varphi}(t)$:

$s_0 := 0$;
**for** $j = 1, 2, \ldots, n$ **do**
  $\widehat{c}_j := \dfrac{1}{|\pi_j|^2}(f - s_{j-1}, \pi_j)$;
  $s_j := s_{j-1} + \widehat{c}_j \pi_j(t)$.
**end for**

Orice sistem $\{\widehat{\pi}_j\}$ care este liniar independent pe $\mathrm{supp} d\lambda$ poate fi ortogonalizat (în raport cu măsura $d\lambda$) prin *procedeul Gram-Schmidt.* Se ia

$$\pi_1 = \widehat{\pi}_1$$

şi apoi, pentru $j = 2, 3, \ldots$ se calculează recursiv

$$\pi_j = \widehat{\pi}_j - \sum_{k=1}^{j-1} c_k \pi_k, \quad c_k = \frac{(\widehat{\pi}_j, \pi_k)}{(\pi_k, \pi_k)}, \quad k = \overline{1, j-1}.$$

Atunci fiecare $\pi_j$ astfel determinat este ortogonal pe toate funcţiile precedente.

## 3.1.3. Eroarea în metoda celor mai mici pătrate. Convergenţa

Am văzut că dacă $\Phi = \Phi_n$ constă din $n$ funcţii $\pi_j$, $j = 1, 2, \ldots, n$ care sunt liniar independente pe $\mathrm{supp} d\lambda$, atunci problema de aproximare în sensul celor mai mici pătrate pentru $d\lambda$

$$\min_{\varphi \in \phi_n} \|f - \varphi\|_{2, d_\lambda} = \|f - \widehat{\varphi}\|_{2, d_\lambda} \tag{3.1.18}$$

are o soluţie unică $\widehat{\varphi} = \widehat{\varphi}_n$, dată de (3.1.14). Există multe moduri de a selecta baza $\{\pi_j\}$ a lui $\Phi_n$ şi de aceea mai multe moduri de a reprezenta soluţia, care conduc totuşi la aceeaşi funcţie. Eroarea în sensul celor mai mici pătrate – cantitatea din dreapta relaţiei (3.1.18) – este independentă de alegerea funcţiilor de bază (deşi calculul soluţiei, aşa cum s-a menţionat 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 (3.1.16)

$$\widehat{\varphi}_n(t) = \sum_{j=1}^{n} \widehat{c}_j \pi_j(t), \quad \widehat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}. \tag{3.1.19}$$

Observăm întâi că eroarea $f - \varphi_n$ este ortogonală pe $\Phi_n$, adică

$$(f - \widehat{\varphi_n}, \varphi) = 0, \ \forall\, \varphi \in \Phi_n \tag{3.1.20}$$

unde produsul scalar este cel din (3.1.3). Deoarece $\varphi$ este o combinaţie liniară de $\pi_k$, este suficient să arătăm (3.1.20) pentru fiecare $\varphi = \pi_k$, $k = 1, 2, \ldots, n$. Înlocuind $\hat{\varphi}_n$ cu expresia sa din (3.1.19) în (3.1.20), găsim

$$(f - \widehat{\varphi}_n, \pi_k) = \left( f - \sum_{j=1}^{n} \widehat{c}_j \pi_k, \pi_k \right) = (f, \pi_k) - \widehat{c}_k(\pi_k, \pi_k) = 0,$$

ultima ecuaţie rezultând din formula pentru $\widehat{c}_k$ din (3.1.19). Rezultatul din (3.1.20) are o interpretare geometrică simplă. Dacă reprezentăm funcţiile ca vectori şi spaţiul $\Phi_n$ ca un plan, atunci pentru orice funcţie $f$ care înţeapă planul $\Phi_n$, aproximanta în sensul celor mai mici pătrate $\widehat{\varphi}_n$ este *proiecţia ortogonală* a lui $f$ pe $\Phi_n$, vezi figura 3.1.

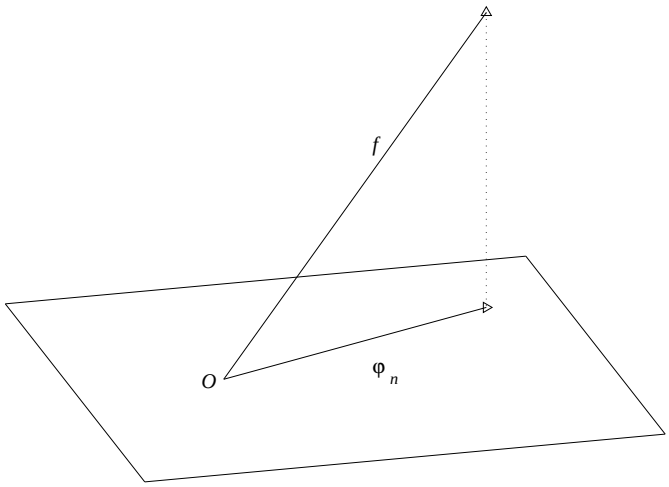



Figura 3.1: Aproximaţia în sensul celor mai mici pătrate ca proiecţie ortogonală

În particular, alegând $\varphi = \widehat{\varphi}_n$ în (3.1.20) obţinem

$$(f - \widehat{\varphi}_n, \widehat{\varphi}_n) = 0$$

şi de aceea, deoarece $f = (f - \widehat{\varphi}) + \widehat{\varphi}$, conform teoremei lui Pitagora şi generalizării sale (3.1.8)

$$\|f\|^2 = \|f - \widehat{\varphi}\|^2 + \|\widehat{\varphi}\|^2 = \|f - \widehat{\varphi}_n\|^2 + \left\|\sum_{j=1}^{n} \widehat{c}_j \pi_j\right\|^2$$
$$= \|f - \widehat{\varphi}_n\|^2 + \sum_{j=1}^{n} |\widehat{c}_j|^2 \|\pi_j\|^2.$$

Exprimând primul termen din dreapta obţinem

$$\|f - \widehat{\varphi}_n\| = \left\{ \|f\|^2 - \sum_{j=1}^{n} |\widehat{c}_j| \|\pi_j\|^2 \right\}^{1/2}, \quad \widehat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}. \tag{3.1.21}$$

De notat că expresia dintre acolade trebuie să fie nenegativă.

Formula (3.1.21) este de interes teoretic, dar de utilitate practică limitată. De notat, într-adevăr, că pe măsură ce eroarea se aproprie de nivelul $\mathrm{eps}$ al preciziei maşinii, calculul erorii din membrul drept al lui (3.1.21) nu poate produce ceva mai mic decât $\sqrt{\mathrm{eps}}$ datorită erorilor comise în timpul scăderilor sub radical (astfel se poate obţine chiar un rezultat negativ sub radical). Utilizând în loc definiţia

$$\|f - \widehat{\varphi}_n\| = \left\{ \int_{\mathbb{R}} [f(t) - \widehat{\varphi}_n(t)]^2 d\lambda(t) \right\}^{\frac{1}{2}},$$

împreună, probabil, cu o regulă de cuadratură (pozitivă) potrivită se garantează că se produce un rezultat nenegativ, potenţial la fel de mic ca $O(\mathrm{eps})$.

Dacă se dă acum un şir de spaţii liniare $\Phi_n$, $n = 1, 2, 3, \ldots$, avem evident

$$\|f - \widehat{\varphi}_1\| \geq \|f - \widehat{\varphi}_2\| \geq \|f - \widehat{\varphi}_3\| \geq \ldots,$$

care rezultă nu numai din (3.1.21), dar mai direct din faptul că

$$\Phi_1 \subset \Phi_2 \subset \Phi_3 \subset \ldots.$$

Deoarece există o infinitate de astfel de spaţii, atunci secvenţa 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 când $n \to \infty$. Este evident din (3.1.21) că o condiţie necesară şi suficientă pentru aceasta este

$$\sum_{j=1}^{\infty} |\widehat{c}_j|^2 \|\pi_j\|^2 = \|f\|^2. \tag{3.1.22}$$

Un mod echivalent de a formula convergenţa este următorul: dându-se $f$ cu $\|f\| < \infty$, adică $\forall\ f \in L^2_{d\lambda}$ şi dându-se un $\varepsilon > 0$ arbitrar de mic, există un întreg $n = n_\varepsilon$ şi o funcţie $\varphi^* \in \Phi_n$ astfel încât $\|f - \varphi^*\| \leq \varepsilon$. O clasă de spaţii $\Phi_n$ având această proprietate se numeşte completă în raport cu norma $\|\cdot\| = \|\cdot\|_{2,d\lambda}$. Vom numi relaţia (3.1.22) relaţia de completitudine. Pentru un interval finit $[a, b]$ putem defini completitudinea lui $\{\varphi_n\}$ de asemenea pentru norma uniformă $\|\cdot\|_\infty$ pe $[a, b]$. Se poate presupune că $f \in C[a, b]$ şi $\pi_j \in C[a, b]$ şi numim $\{\varphi_n\}$ completă în norma $\|\cdot\|_\infty$ dacă pentru orice $f \in C[a, b]$ şi orice $\varepsilon > 0$ există $n = n_\varepsilon$ şi un $\varphi^* \in \Phi_n$ astfel încât $\|f - \varphi^*\|_\infty \leq \varepsilon$. Este uşor de văzut că completitudinea lui $\{\varphi_n\}$ în norma $\|\cdot\|_\infty$ pe $(a, b)$ implică completitudinea lui $\{\varphi_n\}$ în norma din $L^2 \quad \|\cdot\|_{2,d\lambda}$, unde $\mathrm{supp} d\lambda = [a, b]$ şi deci convergenţa procesului de aproximare prin metoda celor mai mici pătrate. Într-adevăr, fie $\varepsilon > 0$ arbitrar şi fie $n$ şi $\varphi^* \in \Phi_n$ astfel încât

$$\|f - \varphi^*\|_\infty \leq \frac{\varepsilon}{\left(\int_{\mathbb{R}} d\lambda(t)\right)^{1/2}}.$$

Aceasta este posibil din ipoteză. Atunci

$$\begin{aligned}
\|f - \varphi^*\|_{2,d\lambda} &= \left(\int_{\mathbb{R}} [f(t) - \varphi^*(t)]^2 d\lambda(t)\right)^{1/2} \leq \\
&\leq \|f - \varphi^*\|_\infty \left(\int_{\mathbb{R}} d\lambda(t)\right)^{1/2} \leq \\
&\leq \frac{\varepsilon}{\left(\int_{\mathbb{R}} d\lambda(t)\right)^{1/2}} \left(\int_{\mathbb{R}} d\lambda(t)\right)^{1/2} = \varepsilon,
\end{aligned}$$

aşa cum s-a afirmat.

**Exemplul 3.1.1.** $\Phi_n = \mathbb{P}_{n-1}$. Aici completitudinea lui $\{\varphi_n\}$ în norma $\|\cdot\|_\infty$ (pe un interval finit $[a, b]$) este o consecinţă a teoremei de aproximare a lui Weierstrass. Astfel aproximaţia polinomială în sensul celor mai mici pătrate pe un interval finit converge întotdeauna (în medie). $\diamondsuit$

# 3.2. Exemple de sisteme ortogonale

Unul dintre cele mai utilizate sisteme este sistemul trigonometric cunoscut din analiza Fourier. Un alt sistem larg utilizat este cel al polinoamelor ortogonale.

(1) **Sistemul trigonometric** este format din funcţiile:

$$1, \cos t, \cos 2t, \cos 3t, \ldots, \sin t, \sin 2t, \sin 3t, \ldots$$

El este ortogonal pe $[0, 2\pi]$ în raport cu măsura

$$d\lambda(t) = \begin{cases} dt & \text{pe } [0, 2\pi] \\ 0 & \text{în rest} \end{cases}$$

Avem

$$\int_0^{2\pi} \sin kt \sin \ell t dt = \begin{cases} 0, & \text{pentru} \quad k \neq \ell \\ \pi, & \text{pentru} \quad k = \ell \end{cases} \qquad k, \ell = 1, 2, 3, \ldots$$

$$\int_0^{2\pi} \cos kt \cos \ell t dt = \begin{cases} 0, & k \neq \ell \\ 2\pi, & k = \ell = 0 \\ \pi, & k = \ell > 0 \end{cases} \qquad k, \ell = 0, 1, 2 \ldots$$

$$\int_0^{2\pi} \sin kt \cos \ell t dt = 0, \quad k = 1, 2, 3, \ldots, \quad \ell = 0, 1, 2, \ldots$$

Aproximarea are forma

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt). \tag{3.2.1}$$

Utilizând (3.1.16) obţinem

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(t) \cos kt dt, \quad k = 1, 2, \ldots$$

$$b_k = \frac{1}{\pi} \int_0^{2\pi} f(t) \sin kt dt, \quad k = 1, 2, \ldots \tag{3.2.2}$$

numiţi *coeficienţi Fourier* ai lui $f$. Ei sunt coeficienţii (3.1.16) pentru sistemul trigonometric. Prin extensie coeficienţii (3.1.16) pentru orice sistem ortogonal $(\pi_j)$ se vor numi coeficienţii Fourier ai lui $f$ relativ la acest sistem. În particular, recunoaştem în seria Fourier trunchiată pentru $k = m$ cea mai bună aproximare a lui $f$ din clasa polinoamelor trigonometrice de grad $\leq n$ relativ la norma

$$\|u\|_2 = \left( \int_0^{2\pi} |u(t)|^2 dt \right)^{1/2}.$$

(2) **Polinoame ortogonale**. Dându-se o măsură $d\lambda$, ştim că orice sistem finit de puteri $1, t, t^2, \ldots$ este liniar independent pe $[a, b]$, dacă $\text{supp} d\lambda = [a, b]$, iar $1, t, \ldots, t^{n-1}$ sunt liniar independente pe $\text{supp} d\lambda = \{t_1, t_2, \ldots, t_N\}$. Deoarece o mulţime de vectori liniar independenţi a unui spaţiu liniar poate fi ortogonalizată prin procedeul Gram-Schmidt, orice măsură $d\lambda$ de tipul considerat generează o mulţime unică de polinoame monice $\pi_j(t, d\lambda)$, $j = 0, 1, 2, \ldots$ ce satisfac

$$\begin{aligned} &\text{grad} \pi_j = j, \quad j = 0, 1, 2, \ldots \\ &\int_{\mathbb{R}} \pi_k(t) \pi_\ell(t) d\lambda(t) = 0, \text{ dacă } k \neq \ell. \end{aligned} \tag{3.2.3}$$

Aceste polinoame se numesc *polinoame ortogonale* relativ la măsura $d\lambda$. Vom permite indicilor să meargă de la 0. Mulţimea $\{\pi_j\}$ este infinită dacă $\mathrm{supp}d\lambda = [a,b]$ şi constă din exact $N$ polinoame $\pi_0, \pi_1, \ldots, \pi_{N-1}$ dacă $\mathrm{supp}d\lambda = \{t_1, \ldots, t_N\}$. În ultimul caz polinoamele se numesc *polinoame ortogonale discrete*.

Între trei polinoame ortogonale monice[2] consecutive există o relaţie liniară. Mai exact, există constantele reale $\alpha_k = \alpha_k(d\lambda)$ şi $\beta_k = \beta_k(d\lambda) > 0$ (depinzând de măsura $d\lambda$) astfel încât

$$\begin{aligned} &\pi_{k+1}(t) = (t-\alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0,1,2,\ldots \\ &\pi_{-1}(t) = 0, \quad \pi_0(t) = 1. \end{aligned} \tag{3.2.4}$$

(Se subînţelege că (3.2.4) are loc pentru orice $k \in \mathbb{N}$ dacă $\mathrm{supp}d\lambda = [a,b]$ şi numai pentru $k = \overline{0, N-2}$ dacă $\mathrm{supp}d\lambda = \{t_1, t_2, \ldots, t_N\}$).

Pentru a demonstra (3.2.4) şi a obţine expresiile coeficienţilor să observăm că

$$\pi_{k+1}(t) - t\pi_k(t)$$

este un polinom de grad $\leq k$, şi deci poate fi exprimat ca o combinaţie liniară a lui $\pi_0, \pi_1, \ldots, \pi_k$. Scriem această combinaţie sub forma

$$\pi_{k+1} - t\pi_k(t) = -\alpha_k\pi_k(t) - \beta_k\pi_{k-1}(t) + \sum_{j=0}^{k-2}\gamma_{k,j}\pi_j(t) \tag{3.2.5}$$

(sumele vide se consideră nule). Înmulţim scalar ambii membri ai relaţiei anterioare cu $\pi_k$ şi obţinem

$$(-t\pi_k, \pi_k) = -\alpha_k(\pi_k, \pi_k)$$

adică

$$\alpha_k = \frac{(t\pi_k, \pi_k)}{(\pi_k, \pi_k)}, \quad k = 0,1,2,\ldots \tag{3.2.6}$$

La fel, înmulţind scalar cu $\pi_{k-1}$ obţinem

$$(-t\pi_k, \pi_{k-1}) = -\beta_k(\pi_{k-1}, \pi_{k-1}).$$

Deoarece $(t\pi_k, \pi_{k-1}) = (\pi_k, t\pi_{k-1})$ şi $t\pi_{k-1}$ diferă de $\pi_k$ printr-un polinom de grad $< k$ se obţine prin ortogonalitate $(t\pi_k, \pi_{k-1}) = (\pi_k, \pi_k)$, deci

$$\beta_k = \frac{(\pi_k, \pi_k)}{(\pi_{k-1}, \pi_{k-1})}, \quad k = 1,2,\ldots \tag{3.2.7}$$

Înmulţind (3.2.5) cu $\pi_\ell$, $\ell < k-1$, se obţine

$$\gamma_{k,\ell} = 0, \quad \ell = 0,1,\ldots,k-1. \tag{3.2.8}$$

[2]Un polinom se numeşte *monic* dacă coeficientul său dominant este 1.

Formula de recurenţă (3.2.4) ne dă o modalitate practică de determinare a polinoamelor ortogonale. Deoarece $\pi_0 = 1$, putem calcula $\alpha_0$ cu (3.2.6) pentru $k = 0$. Se continuuă apoi cu $\pi_1$, utilizând (3.2.4) pentru $k = 0$. Mai departe, utilizând alternativ (3.2.6), (3.2.7) şi (3.2.4) putem calcula câte polinoame ortogonale dorim. Procedeul – numit procedura lui Stieltjes [3] – 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 complică lucrurile. Din fericire, pentru multe măsuri speciale importante, coeficienţii se cunosc explicit. Cazul special când măsura este simetrică (adică $d\lambda(t) = w(t)$ cu $w(-t) = w(t)$ şi $\mathrm{supp} d\lambda$ simetrică faţă de origine) merită o atenţie specială deoarece în acest caz $\alpha_k = 0,\ \forall\, k \in \mathbb{N}$, conform lui (3.2.1) căci

$$(t\pi_k, \pi_k) = \int_{\mathbb{R}} w(t) t\pi_k^2(t)dt = \int_a^b w(t) t\pi_k^2(t)dt = 0,$$

deoarece avem o integrală dintr-o funcţie impară pe un domeniu simetric faţa de origine.

## 3.2.1. Exemple de polinoame ortogonale

### Polinoamele lui Legendre

Se definesc prin aşa-numita formulă a lui Rodrigues

$$\pi_k(t) = \frac{k!}{(2k)!}\frac{d^k}{dt^k}(t^2-1)^k. \tag{3.2.9}$$

Verificăm întâi ortogonalitatea pe $[-1, 1]$ în raport cu măsura $d\lambda(t) = dt$. Pentru orice $0 \le \ell < k$, prin integrare repetată prin părţi se obţine:

$$\int_{-1}^{1} \frac{d^k}{dt^k}(t^2-1)^k = \sum_{m=0}^{\ell} \ell(\ell-1)\dots(\ell-m+1)t^{\ell-m}\frac{d^{k-m-1}}{dt^{k-m-1}}(t^2-1)^k\Big|_{-1}^{1} = 0, \tag{3.2.10}$$

[3] Thomas Jan Stieltjes (1856-1894), matematician olandez, a studiat la Institutul Tehnic din Delft, dar nu şi-a luat niciodată licenţa datorită aversiunii pe care o avea pentru examene. A lucrat la Observatorul astronomic din Leyda, în calitate de ,,calculator asistent pentru calcule astronomice". Lucrările sale timpurii au atras atenţia lui Hermite, care i-a asigurat un post universitar la Toulouse. Prietenia lor a fost exemplară. Stieltjes este cunoscut pentru lucrările sale despre fracţii continue şi problema momentelor, în care printre altele a introdus integrala care îi poartă numele. A murit de tuberculoză la 38 de ani.

ultima relaţie având loc deoarece $0 \leq k-m-1 < k$. Deci,

$$(\pi_k, p) = 0, \qquad \forall p \in \mathbb{P}_{k-1},$$

demonstrându-se astfel ortogonalitatea. Datorită simetriei, putem scrie

$$\pi_k(t) = t^k + \mu_k t^{k-2} + \ldots, \quad k \geq 2$$

şi observând (din nou datorită simetriei) că relaţia de recurenţă are forma

$$\pi_{k+1}(t) = t\pi_k(t) - \beta_k \pi_{k-1}(t),$$

obţinem

$$\beta_k = \frac{t\pi_k(t) - \pi_{k+1}(t)}{\pi_{k-1}(t)},$$

care este valabilă pentru orice $t$. Făcând $t \to \infty$,

$$\beta_k = \lim_{t\to\infty} \frac{t\pi_k(t) - \pi_{k+1}(t)}{\pi_{k-1}(t)} = \lim_{t\to\infty} \frac{(\mu_k - \mu_{k+1})t^{k-1} + \ldots}{t^{k-1} + \ldots} = \mu_k - \mu_{k+1}.$$

(Dacă $k = 1$, punem $\mu_1 = 0$.) Din formula lui Rodrigues rezultă

$$\begin{aligned}\pi_k(t) &= \frac{k!}{(2k)!}\frac{d^k}{dt^k}\left(t^{2k} - kt^{2k-2} + \ldots\right) \\ &= \frac{k!}{(2k)!}\left(2k(2k-1)\ldots(k+1)t^k - k(2k-2)(2k-3)\ldots(k-1)t^{k-1} + \ldots\right) \\ &= t^k - \frac{k(k-1)}{2(2k-1)}t^{k-2} + \ldots,\end{aligned}$$

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 $\mu_1 = 0$,

$$\beta_k = \frac{1}{4 - k^{-2}}, \quad k \geq 1. \tag{3.2.11}$$

**Polinoamele Cebîşev de speţa I**

Polinoamele lui Cebîşev [4] de speţa I se pot defini prin relaţia

$$T_n(x) = \cos(n \arccos x), \quad n \in \mathbb{N}. \tag{3.2.12}$$

Din identitatea trigonometrică

$$\cos(k+1)\theta + \cos(k-1)\theta = 2\cos\theta\cos k\theta$$

şi din (3.2.12), punând $\theta = \arccos x$ se obţine

$$\begin{aligned} T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x) \quad k = 1, 2, 3, \ldots \\ T_0(x) &= 1, \quad T_1(x) = x. \end{aligned} \tag{3.2.13}$$

De exemplu

$$\begin{aligned} T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1 \end{aligned}$$

ş.a.m.d.

Din relaţia (3.2.13) se obţine pentru coeficientul dominant al lui $T_n$ valoarea $2^{n-1}$ (dacă $n \geq 1$), deci polinomul Cebîşev de speţa I monic este

$$\overset{\circ}{T}_n(x) = \frac{1}{2^{n-1}} T_n(x), \quad n \geq 0, \quad \overset{\circ}{T}_0 = T_0. \tag{3.2.14}$$

Din (3.2.12) se pot obţine rădăcinile lui $T_n$

$$x_k^{(n)} = \cos\theta_k^{(n)}, \quad \theta_k^{(n)} = \frac{2k-1}{2n}\pi, \quad k = \overline{1, n}. \tag{3.2.15}$$

Ele sunt proiecţiile pe axa reală ale punctelor de pe cercul unitate de argument $\theta_k^{(n)}$; figura 3.2 ilustrează acest lucru pentru n=4.

Pe intervalul $[-1, 1]$ $T_n$ oscilează de la +1 la -1, atingând aceste valori extreme în punctele

$$y_k^{(n)} = \cos\eta_k^{(n)}, \quad \eta_k^{(n)} = \frac{k\pi}{n}, \quad k = \overline{0, n}.$$

În figura 3.3 apar graficele unor polinoame Cebîşev de speţa I.

[4] Pafnuti Levovici Cebîşev (1821-1894), matematician rus, cel mai important reprezentant al şcolii matematice din Sankt Petersburg. A avut contribuţii de pionierat în domeniul teoriei numerelor, calculului probabilităţilor şi teoriei aproximării. Este considerat fondatorul teoriei constructive a funcţiilor, dar a lucrat şi în mecanică şi în balistică.

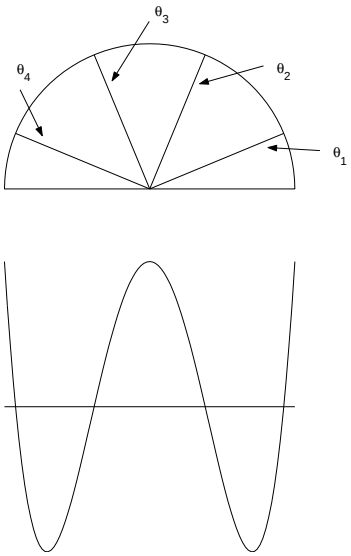



Figura 3.2: Polinomul Cebîşev $T_4$ şi rădăcinile sale

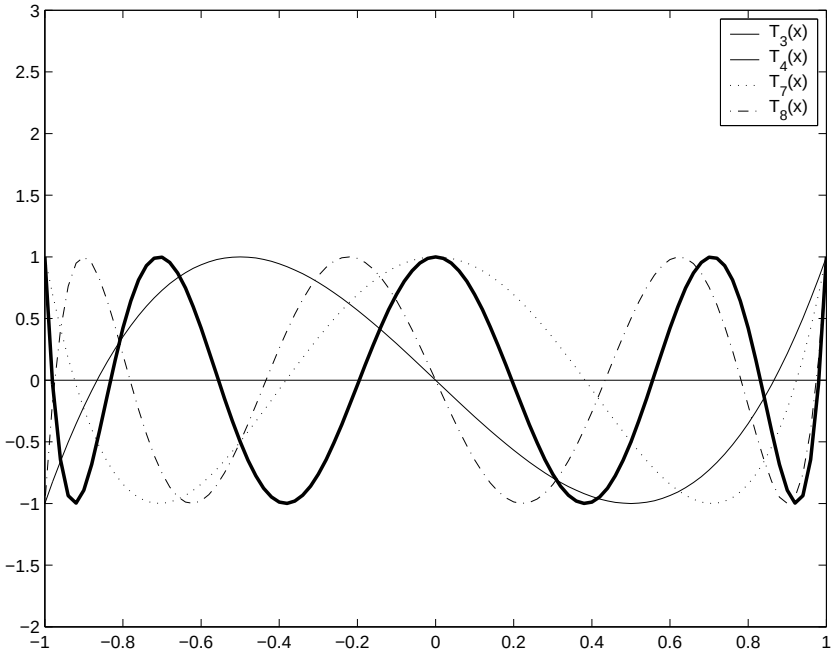



Figura 3.3: Polinoamele Cebîşev $T_3$, $T_4$, $T_7$, $T_8$ pe [-1,1]

Polinoamele Cebîşev de speţa I sunt ortogonale în raport cu măsura

$$d\lambda(x) = \frac{dx}{\sqrt{1-x^2}}, \quad \text{pe } [-1,1].$$

Se verifică uşor din (3.2.12) că

$$\int_{-1}^{1} T_k(x)T_\ell(x)\frac{dx}{\sqrt{1-x^2}} = \int_0^\pi T_k(\cos\theta)T_\ell(\cos\theta)d\theta$$
$$= \int_0^\pi \cos k\theta \cos \ell\theta d\theta = \begin{cases} 0, & \text{dacă } k \neq \ell, \\ \pi, & \text{dacă } k = \ell = 0, \\ \pi/2, & \text{dacă } k = \ell \neq 0. \end{cases} \tag{3.2.16}$$

Dezvoltarea în serie Fourier de polinoame Cebîşev este dată de

$$f(x) = {\sum_{j=0}^{\infty}}' c_j T_j(x) := \frac{1}{2}c_0 + \sum_{j=1}^{\infty} c_j T_j(x), \tag{3.2.17}$$

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 (3.2.17) numai termenii de grad cel mult $n$ se obţine o aproximare polinomială utilă de grad $n$

$$\tau_n(x) = {\sum_{j=0}^{n}}' c_j T_j(x), \tag{3.2.18}$$

având eroarea

$$f(x) - \tau_n(x) = \sum_{j=n+1}^{\infty} c_j T_j(x) \approx c_{n+1}T_{n+1}(x). \tag{3.2.19}$$

Aproximanta din (3.2.18) este cu atât mai bună cu cât coeficienţii din extremitatea dreaptă tind mai repede către zero. Eroarea (3.2.19) oscilează în esenţă între $+c_{n+1}$ şi $-c_{n+1}$ şi este deci de mărime „uniformă“. Acest lucru contrastează puternic cu dezvoltarea Taylor în jurul lui $x = 0$, unde polinomul de grad $n$ are eroarea proporţională cu $x^{n+1}$ pe $[-1,1]$.

Dintre toate polinoamele monice de grad $n$ $\overset{\circ}{T}_n$ are norma uniformă cea mai mică.

**Teorema 3.2.1 (Cebîşev)** *Pentru orice polinom monic $\overset{\circ}{p}_n$ de grad $n$ are loc*

$$\max_{-1\le x\le 1}\left|\overset{\circ}{p}_n(x)\right| \ge \max_{-1\le x\le 1}\left|\overset{\circ}{T}_n(x)\right| = \frac{1}{2^{n-1}}, \quad n \ge 1, \tag{3.2.20}$$

*unde $\overset{\circ}{T}_n(x)$ este dat de (3.2.14).*

*Demonstraţie.* Se face prin reducere la absurd. Presupunem că

$$\max_{-1\le x\le 1}\left|\overset{\circ}{p}_n(x)\right| < \frac{1}{2^{n-1}}. \tag{3.2.21}$$

Atunci polinomul $d_n(x) = \overset{\circ}{T}_n(x) - \overset{\circ}{p}_n(x)$ (de grad $\le n-1$) satisface

$$d_n\left(y_0^{(n)}\right) > 0,\ d_n\left(y_1^{(n)}\right) < 0,\ d_n\left(y_2^{(n)}\right) > 0, \ldots, (-1)^n d_n\left(y_n^{(n)}\right) > 0. \tag{3.2.22}$$

Deoarece $d_n$ are $n$ schimbări de semn, el este identic nul; aceasta contrazice (3.2.22) şi astfel (3.2.21) nu poate fi adevărată. □

Rezultatul (3.2.20) 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 $\overset{\circ}{T}_n$ luaţi cu semnul minus. Din teoria aproximaţiilor uniforme se ştie că cea mai bună aproximare polinomială uniformă este unică. Deci, egalitatea în (3.2.20) poate avea loc numai dacă $\overset{\circ}{p}_n(x) = \overset{\circ}{T}_n(x)$.

**Polinoamele lui Cebîşev de speţa a doua**

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 $d\lambda(t) = w(t)dt$, $w(t) = \sqrt{1-t^2}$. Relaţia de recurenţă este

$$Q_{n+1}(t) = 2tQ_n(t) - Q_{n-1}(t), \quad Q_0(t) = 1, \quad Q_1(t) = 2t.$$

**Polinoamele lui Laguerre**

Polinoamele lui Laguerre [5] sunt ortogonale pe $[0,\infty)$ în raport cu ponderea $w(t) = t^\alpha e^{-t}$. Se definesc prin

$$l_n^\alpha(t) = \frac{e^t t^{-\alpha}}{n!}\frac{d^n}{dt^n}(t^{n+\alpha}e^{-t}) \text{ pentru } \alpha > 1.$$

[5] Edmond Laguerre (1834-1886) matematician francez, activ în Paris, cu contribuţii esenţiale în geometrie, algebră şi analiză.

Relaţia de recurenţă este

$$nl_n^\alpha(t) - (2n - 1 + \alpha - t)l_{n-1}^\alpha(t) + (n - 1 - \alpha)l_{n-2}^\alpha(t) = 0.$$

**Polinoamele lui Hermite**

Polinoamele lui Hermite 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 $w(t) = e^{-t^2}$ şi verifică relaţia de recurenţă

$$H_{n+1}(t) = 2tH_n(t) - 2nH_{n-1}(t),$$

$$H_0(t) = 1, \quad H_1(t) = 2t.$$

**Polinoamele lui Jacobi**

Sunt ortogonale pe $[-1, 1]$ în raport cu ponderea

$$w(t) = (1 - t)^\alpha (1 + t)^\beta.$$

Pentru detalii privind aspectele algoritmice legate de polinoame ortogonale a se vedea [21].

## 3.3. Interpolare polinomială

### 3.3.1. Spaţiul $H^n[a, b]$

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

$$H^n[a, b] = \{f : [a, b] \to \mathbb{R} : f \in C^{n-1}[a, b], f^{(n-1)} \text{ absolut continuă pe } [a, b]\}. \tag{3.3.1}$$

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

$$f(x) = \sum_{k=0}^{n-1} \frac{(x - a)^k}{k!} f^{(k)}(a) + \int_a^x \frac{(x - t)^{n-1}}{(n - 1)!} f^{(n)}(t)dt. \tag{3.3.2}$$

$H^n[a, b]$ este un spaţiu liniar.

**Observaţia 3.3.1.** Funcţia $f : I \to \mathbb{R}$, $I$ interval, se numeşte 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 următoare, datorată lui Peano [6], de o importanţă deosebită în analiza numerică, este o teoremă de reprezentare a funcţionalelor liniare reale, definite pe $H^n[a, b]$.

**Teorema 3.3.2 (Peano)** *Fie $L$ o funcţională reală, continuă, definită pe $H^n[a, b]$. Dacă $KerL = \mathbb{P}_{n-1}$ atunci*

$$Lf = \int_a^b K(t) f^{(n)}(t) dt, \tag{3.3.3}$$

*unde*

$$K(t) = \frac{1}{(n-1)!} L[(\cdot - t)_+^{n-1}] \quad \textit{(nucleul lui Peano).} \tag{3.3.4}$$

**Observaţia 3.3.3.** Funcţia

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

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

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

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

unde

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

---

[6] Giuseppe Peano (1858-1932), matematician italian activ la Torino, cu contribuţii fundamentale în logica matematică, teoria mulţimilor, şi fundamentele matematicii. Teoremele generale de existenţă din domeniul teoriei ecuaţiilor diferenţiale îi poartă numele.

Aplicând $L$ obţinem

$$Lf = \underbrace{LT_{n-1}}_{0} + LR_{n-1} \Rightarrow Lf = \frac{1}{(n-1)!} L\left(\int_a^b (\cdot - t)_+^{n-1} f^{(n)}(t)dt\right) =$$

$$\overset{cont}{=} \frac{1}{(n-1)!}\int_a^b L(\cdot - t)_+^{n-1} f^{(n)}(t)dt.$$

□

**Observaţia 3.3.4.** Concluzia teoremei rămâne valabilă şi dacă $L$ nu este continuă, ci are forma

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

unde $BV[a,b]$ este mulţimea funcţiilor cu variaţie mărginită pe $[a,b]$. ♢

**Corolarul 3.3.5** *Dacă $K$ păstrează semn constant pe $[a,b]$ şi $f^{(n)}$ este continuă pe $[a,b]$, atunci există $\xi \in [a,b]$ astfel încât*

$$Lf = \frac{1}{n!} f^{(n)}(\xi) Le_n, \tag{3.3.5}$$

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

*Demonstraţie.* Deoarece $K$ păstrează u{a} semn constant putem aplica în (3.3.3) teorema de medie

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

Luând $f = e_n$ se obţine chiar (3.3.5). □

## 3.3.2. Interpolare Lagrange

Fie intervalul închis $[a,b] \subset \mathbb{R}$, $f : [a,b] \to \mathbb{R}$ şi o mulţime de $m+1$ puncte distincte $\{x_0, x_1, \ldots, x_m\} \subset [a,b]$.

**Teorema 3.3.6** *Există un polinom şi numai unul $L_m f \in \mathbb{P}_m$ astfel încât*

$$\forall\, i = 0, 1, \ldots, m, \quad (L_m f)(x_i) = f(x_i); \tag{3.3.6}$$

*acest polinom se scrie sub forma*

$$(L_m f)(x) = \sum_{i=0}^{m} f(x_i)\ell_i(x), \tag{3.3.7}$$

*unde*

$$\ell_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{m} \frac{x - x_j}{x_i - x_j}. \tag{3.3.8}$$

**Definiţia 3.3.7** *Polinomul $L_m f$ definit astfel se numeşte* polinom de interpolare Lagrange [7] *a lui $f$ relativ la punctele $x_0, x_1, \ldots, x_m$, iar funcţiile $\ell_i(x)$, $i = \overline{0, m}$, se numesc* polinoame de bază (fundamentale) Lagrange *asociate acelor puncte.*

*Demonstraţie.* Se verifică imediat că $\ell_i \in \mathbb{P}_i$ şi că $\ell_i(x_j) = \delta_{ij}$ (simbolul lui Kronecker); rezultă că polinomul $L_m f$ definit de (3.3.6) este de grad cel mult $m$ şi verifică (3.3.7). Presupunem că există un alt polinom $p_m^* \in \mathbb{P}_m$ care verifică (3.3.7) şi punem $q_m = L_m - p_m^*$; avem $q_m \in \mathbb{P}_m$ şi $\forall\ i = 0, 1, \ldots, m$, $q_m(x_i) = 0$; deci $q_m$ având $(m+1)$ rădăcini distincte este identic nul, de unde unicitatea lui $L_m$. □

**Observaţia 3.3.8.** Polinomul fundamental $\ell_i$ este deci unicul polinom care verifică

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

Punând

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

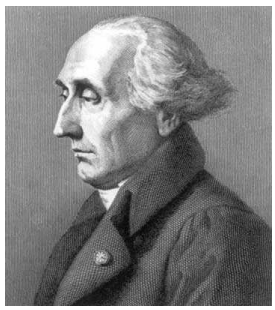
din (3.3.8) se deduce că $\forall\ x \neq x_i, \quad \ell_i(x) = \frac{u(x)}{(x-x_i)u'(x_i)}$. ♢

Demonstrând teorema 3.3.6 am demonstrat de fapt existenţa şi unicitatea soluţiei problemei generale de interpolare Lagrange:

(PGIL) Fiind date $b_0, b_1, \ldots, b_m \in \mathbb{R}$, să se determine

$$p_m \in \mathbb{P}_m \text{ astfel încât } \forall\ i = 0, 1, \ldots, n, \quad p_m(x_i) = b_i. \tag{3.3.9}$$

[7] Joseph Louis Lagrange (1736-1813), protejat al lui Euler. Clairaut scria despre tânărul Lagrange: „...un tânăr nu mai puţin remarcabil prin talent decât prin modestie; temperamentul său este blând şi melancolic; nu cunoaşte altă plăcere decât studiul." Lagrange a avut contribuţii fundamentale în calculul variaţional, teoria numerelor şi analiză matematică. Este cunoscut şi pentru reprezentarea pe care a dat-o restului din formula lui Taylor. A dat formula de interpolare în 1794. Lucrarea sa *Mécanique Analytique*, publicată în 1788, l-a făcut unul din fondatorii mecanicii analitice.



Problema (3.3.9) conduce la un sistem liniar de $(m+1)$ ecuaţii cu $(m+1)$ necunoscute (coeficienţii lui $p_m$).

Din teoria sistemelor liniare se ştie că

$$\{\text{existenţa unei soluţii } \forall\, b_0, b_1, \ldots, b_m\} \Leftrightarrow \{\text{unicitatea soluţiei}\} \Leftrightarrow$$

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

Punem $p_m = a_0 + a_1 x + \cdots + a_m x^m$

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

şi notăm cu $V = (v_{ij})$ matricea pătratică de ordin $m+1$ cu elementele $v_{ij} = x_i^j$. Ecuaţia (3.3.9) se scrie sub forma

$$Va = b.$$

Matricea $V$ este inversabilă (determinantul ei este Vandermonde); se arată uşor că $V^{-1} = U^T$, unde $U = (u_{ij})$ cu $\ell_i(x) = \sum_{k=0}^{m} u_{ik} x^k$; se obţine în acest mod un procedeu puţin costisitor de inversare a matricei Vandermonde şi prin urmare şi de rezolvare a sistemului (3.3.9).

**Exemplul 3.3.9.** Polinomul de interpolare Lagrange corespunzător unei funcţii $f$ şi nodurilor $x_0$ şi $x_1$ este

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

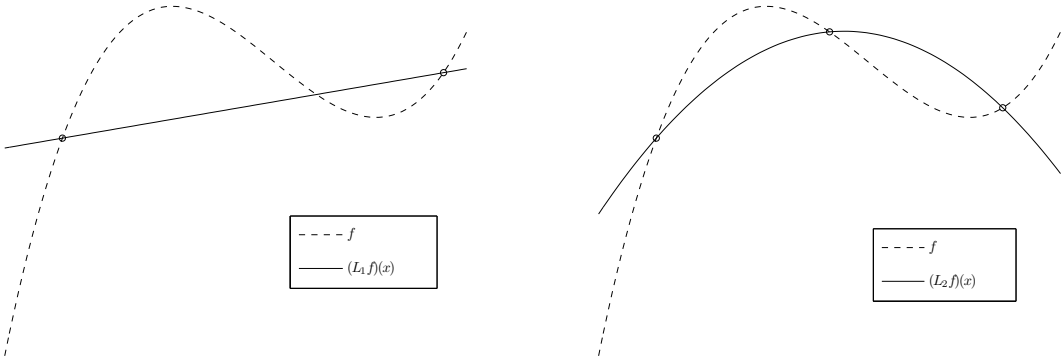
adică dreapta care trece prin punctele $(x_0, f(x_0))$ şi $(x_1, f(x_1))$. Analog, polinomul de interpolare Lagrange corespunzător unei funcţii $f$ şi nodurilor $x_0$, $x_1$ şi $x_2$ este

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

adică parabola care trece prin punctele $(x_0, f(x_0))$, $(x_1, f(x_1))$ şi $(x_2, f(x_2))$. Interpretarea lor geometrică apare în figura 3.4. ♢

## 3.3.3. Interpolare Hermite

În loc să facem să coincidă $f$ şi polinomul de interpolare în punctele $x_i$ din $[a, b]$, am putea face ca $f$ şi polinomul de interpolare să coincidă împreună cu derivatele lor până la ordinul $r_i$ în punctele $x_i$. Se obţine:




(a) $(L_1 f)$ (b) $(L_2 f)$

Figura 3.4: Interpretarea geometrică a lui $L_1 f$ (stânga) şi $L_2 f$

**Teorema 3.3.10** *Fiind date $(m+1)$ puncte distincte $x_0, x_1, \ldots, x_m$ din $[a,b]$ şi $(m+1)$ numere naturale $r_0, r_1, \ldots, r_m$, punem $n = m + r_0 + r_1 + \cdots + r_m$. Atunci, fiind dată o funcţie $f$, definită pe $[a,b]$ şi admiţând derivate de ordin $r_i$ în punctele $x_i$ există un singur polinom şi numai unul $H_n f$ de grad $\leq n$ astfel încât*

$$\forall\, (i,\ell),\ 0 \leq i \leq m,\ 0 \leq \ell \leq r_i \qquad (H_n f)^{(\ell)}(x_i) = f^{(\ell)}(x_i), \tag{3.3.10}$$

*unde $f^{(\ell)}(x_i)$ este derivata de ordinul $\ell$ a lui $f$ în $x_i$.*

**Definiţia 3.3.11** *Polinomul definit în acest mod se numeşte* polinom de interpolare al lui Hermite [8] *al funcţiei $f$ relativ la punctele $x_0, x_1, \ldots, x_m$ şi la întregii $r_0, r_1, \ldots, r_m$.*

*Demonstraţie.* Ecuaţia (3.3.10) conduce la un sistem liniar de $(n+1)$ ecuaţii cu $(n+1)$ necunoscute (coeficienţii lui $H_n f$), deci este suficient să arătăm că sistemul omogen corespunzător admite doar soluţia nulă, adică relaţiile

$$H_n f \in \mathbb{P}_n \text{ şi } \forall\, (i,\ell),\ 0 \leq i \leq k,\ 0 \leq \ell \leq r_i,\ (H_n f)^{(\ell)}(x_i) = 0$$

[8] Charles Hermite (1822-1901) matematician francez de frunte, membru al Academiei Franceze, cunoscut pentru lucrările sale în domeniul teoriei numerelor, algebră şi analiză. A devenit faimos după ce a dat, în 1873, demonstraţia transcendenţei numărului $e$.

ne asigură că pentru orice $i = 0, 1, \ldots, m$ $x_i$ este rădăcină de ordinul $r_i + 1$ a lui $H_n f$; prin urmare $H_n f$ are forma

$$(H_n f)(x) = q(x) \prod_{i=0}^{m} (x - x_i)^{r_i+1},$$

unde $q$ este un polinom. Cum $\sum_{i=0}^{m}(r_i + 1) = n + 1$, acest lucru nu este compatibil cu apartenenţa lui $H_n$ la $\mathbb{P}_n$, decât dacă $q \equiv 0$ şi deci $H_n \equiv 0$. □

**Observaţia 3.3.12.** 1) Dându-se numerele reale $b_{i\ell}$ pentru orice pereche $(i, \ell)$ astfel încât $0 \leq i \leq k$ şi $0 \leq \ell \leq r_i$, am arătat că problema generală de interpolare Hermite

$$\begin{aligned} &\text{să se determine } p_n \in \mathbb{P}_n \text{ a.î. } \forall\ (i, \ell) \text{ cu } 0 \leq i \leq m \text{ şi} \\ &0 \leq \ell \leq r_i,\ p_n^{(\ell)}(x_i) = b_{i\ell} \end{aligned} \tag{3.3.11}$$

admite o soluţie şi numai una. În particular, dacă alegem pentru o pereche $(i, \ell)$ dată $b_{i\ell} = 1$ şi $b_{jn} = 0$, $\forall\ (j, m) \neq (i, \ell)$ se obţine un polinom de bază (fundamental) de interpolare Hermite relativ la punctele $x_0, x_1, \ldots, x_m$ şi la întregii $r_0, r_1, \ldots, r_m$. Polinomul de interpolare Hermite definit prin (3.3.10) se obţine cu ajutorul polinoamelor de bază(fundamentale) cu formula

$$(H_n f)(x) = \sum_{i=0}^{m} \sum_{\ell=0}^{r_i} f^{(l)}(x) h_{i\ell}(x). \tag{3.3.12}$$

Punând

$$q_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{k} \left( \frac{x - x_j}{x_i - x_j} \right)^{r_{j+1}},$$

se verifică că polinoamele de bază $h_{i\ell}$ sunt definite prin relaţiile de recurenţă

$$h_{ir_i}(x) = \frac{(x - x_i)^{r_i}}{r_i!} q_i(x)$$

şi pentru $\ell = r_{i-1}, r_{i-2}, \ldots, 1, 0$

$$h_{i\ell}(x) = \frac{(x - x_i)^{\ell}}{\ell!} q_i(x) - \sum_{j=l+1}^{r_i} \binom{j}{\ell} q_i^{(j-l)}(x_i) h_{ij}(x).$$

2) Matricea $V$ asociată sistemului liniar (3.3.11) se numeşte matrice Vandermonde generalizată; ea este inversabilă, iar elementele matricei ei inverse sunt coeficienţii polinoamelor $h_{i\ell}$.

3) Interpolarea Lagrange este un caz particular al interpolării Hermite (pentru $r_i = 0$, $i = 0, 1, \ldots, m$); polinomul Taylor este un caz particular pentru $m = 0$ şi $r_0 = n$. ♢

Vom prezenta o expresie mai convenabilă a polinoamelor fundamentale Hermite, obţinută de Dimitrie D. Stancu în 1957. Ele verifică relaţiile

$$\begin{aligned} h_{kj}^{(p)}(x_\nu) &= 0, \quad \nu \neq k,\ p = \overline{0, r_\nu} \\ h_{kj}^{(p)}(x_k) &= \delta_{jp}, \quad p = \overline{0, r_k} \end{aligned} \tag{3.3.13}$$

pentru $j = \overline{0, r_k}$ şi $\nu, k = \overline{0, m}$. Introducând notaţiile

$$u(x) = \prod_{k=0}^{m} (x - x_k)^{r_k+1}$$

şi

$$u_k(x) = \frac{u(x)}{(x - x_k)^{r_k+1}},$$

din (3.3.13) rezultă că $h_{kj}$ are forma

$$h_{kj}(x) = u_k(x)(x - x_k)^j g_{kj}(x), \qquad g_{kj} \in \mathbb{P}_{r_k - j}. \tag{3.3.14}$$

Dezvoltând $g_{kj}$ cu formula lui Taylor, avem

$$g_{kj}(x) = \sum_{\nu=0}^{r_k - j} \frac{(x - x_k)^\nu}{\nu!} g_{kj}^\nu(x_k); \tag{3.3.15}$$

mai rămân de determinat valorile lui $g_{kj}^\nu(x_k)$, $\nu = \overline{0, r_k - j}$. Scriind (3.3.14) sub forma

$$(x - x_k)^j g_{kj}(x) = h_{kj}(x) \frac{1}{u_k(x)},$$

şi aplicând formula lui Lebniz pentru derivata de ordinul $j + \nu$ a produsului se obţine

$$\sum_{s=0}^{j+\nu} \binom{j+\nu}{s} \left[(x - x_k)^j\right]^{(j+\nu-s)} g_{kj}^{(s)}(x) = \sum_{s=0}^{j+\nu} \binom{j+\nu}{s} h_{kj}^{(j+\nu-s)}(x) \left[\frac{1}{u_k(x)}\right]^{(s)}.$$

Lând $x = x_k$, toţi termenii din ambii membri se vor anula, cu excepţia celor corespunzători lui $s = \nu$. Avem deci

$$\binom{j+\nu}{\nu} j! g_{kj}^{(\nu)}(x_k) = \binom{j+\nu}{\nu} \left[\frac{1}{u_k(x)}\right]_{x=x_k}^{(\nu)}, \qquad \nu = \overline{0, r_k - j}.$$

Am obţinut

$$g_{kj}^{(\nu)}(x_k) = \frac{1}{j!}\left[\frac{1}{u_k(x)}\right]^{(\nu)}_{x=x_k},$$

iar din (3.3.15) şi (3.3.14) avem în final

$$h_{kj}(x) = \frac{(x-x_k)^j}{j!}u_k(x)\sum_{\nu=0}^{r_k-j}\frac{(x-x_k)^\nu}{\nu!}\left[\frac{1}{u_k(x)}\right]^{(\nu)}_{x=x_k}.$$

**Propoziţia 3.3.13** *Operatorul $H_n$ este proiector, adică*

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

*Demonstraţie.* Liniaritatea rezultă imediat din formula (3.3.12). Datorită unicităţii polinomului de interpolare Hermite $H_n(H_nf)-H_nf$ este identic nul, deci $H_n(H_nf) = H_nf$ şi am arătat idempotenţa. □

**Exemplul 3.3.14.** Polinomul de interpolare Hermite corespunzător unei funcţii $f$ şi nodurilor duble 0 şi 1 are expresia

$$(H_3f)(x) = h_{00}(x)f(0) + h_{10}(x)f(1) + h_{01}(x)f'(0) + h_{11}(x)f'(1),$$

unde

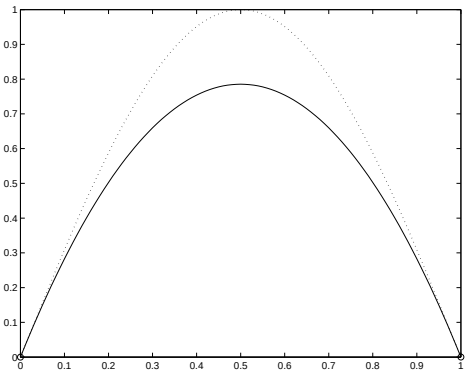
$$\begin{aligned} h_{00}(x) &= (x-1)^2(2x+1),\\ h_{01}(x) &= x(x-1)^2,\\ h_{10}(x) &= x^2(3-2x),\\ h_{11}(x) &= x^2(x-1). \end{aligned}$$

Dacă se adaugă nodul $x = \frac{1}{2}$, calitatea aproximării creşte (vezi figura 3.5). ♢

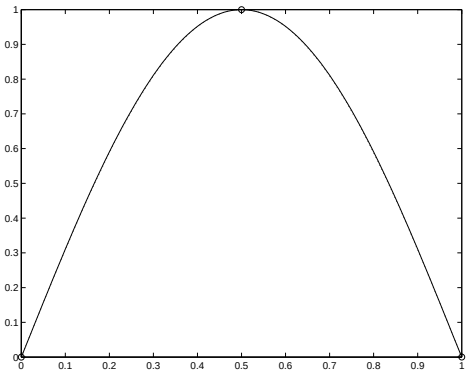
## 3.3.4. Expresia erorii de interpolare

Reamintim că norma unui operator liniar $P_n$ se poate defini prin

$$\|P_n\| = \max_{f\in C[a,b]}\frac{\|P_nf\|}{\|f\|}, \tag{3.3.16}$$



(a) $(H_3 f)$



(b) $(H_3 f)$

Figura 3.5: Polinoamele de interpolare Hermite $(H_3 f)$ (—) corespunzător funcţiei $f : [0,1] \to \mathbb{R}$ , $f(x) = \sin \pi x$ şi nodurilor duble $x_0 = 0$ şi $x_1 = 1$ $(\cdots)$(stânga) şi $(H_5 f)$ (—) corespunzător funcţiei $f : [0,1] \to \mathbb{R}$ , $f(x) = \sin \pi x$ $(\cdots)$ şi nodurilor duble $x_0 = 0$, $x_1 = \frac{1}{2}$ şi $x_2 = 1$

unde în membrul drept se ia o normă convenabilă pentru funcţii. Luând norma $L^\infty$, din formula lui Lagrange se obţine

$$\begin{aligned} \|(L_m f)(.)\|_\infty &= \max_{a \le x \le b} \left| \sum_{i=0}^{m} f(x_i) \ell_i(x) \right| \\ &\le \|f\|_\infty \max_{a \le x \le b} \sum_{i=0}^{m} |\ell_i(x)|. \end{aligned} \tag{3.3.17}$$

Fie $\|\lambda_m\|_\infty = \lambda_m(x_\infty)$. Egalitatea are loc pentru o funcţie $\varphi \in C[a,b]$, liniară pe porţiuni şi care verifică $\varphi(x_i) = \operatorname{sgn} \ell_i(x_\infty)$, $i = \overline{0,m}$. Deci,

$$\|P_n\|_\infty = \Lambda_m, \tag{3.3.18}$$

unde

$$\Lambda_m = \|\lambda_m\|_\infty, \qquad \lambda_m(x) = \sum_{i=0}^{m} |\ell_i(x)|. \tag{3.3.19}$$

Funcţia $\lambda_m(x)$ şi maximul său $\Lambda_m$ se numesc *funcţia lui Lebesgue* [9] şi respectiv *constanta lui Lebesgue* pentru interpolarea Lagrange. Ele furnizează o primă estimare a erorii de interpolare: fie $\mathcal{E}_m(f)$ eroarea în cea mai bună aproximare a lui $f$ prin polinoame de grad $\leq m$,

$$\mathcal{E}_m(f) = \min_{p \in \mathbb{P}_m} \|f - p\|_\infty = \|f - \hat{p}_n\|_\infty, \tag{3.3.20}$$

unde $\hat{p}_n$ este polinomul de grad $m$ de cea mai bună aproximare a lui $f$. Utilizând faptul că operatorul $L_m$ este proiector şi formulele (3.3.17) şi (3.3.19), se găseşte

$$\begin{aligned} \|f - L_m f\| &= \|f - \hat{p}_m - L_m(f - \hat{p}_m)\|_\infty \\ &\leq \|f - \hat{p}_m\|_\infty + \Lambda_m \|f - \hat{p}_m\|_\infty; \end{aligned}$$

adică,

$$\|f - L_m f\|_\infty \leq (1 + \Lambda_m)\mathcal{E}_m(f). \tag{3.3.21}$$

Astfel, cu cât $f$ poate fi aproximată mai bine prin polinoame de grad $\leq m$, cu atât este mai mică eroarea de interpolare. Din păcate, $\Lambda_m$ nu este uniform mărginită: indiferent de cum se aleg nodurile $x_i = x_i^{(m)}$, $i = \overline{0, m}$ se poate arăta că $\Lambda_m > O(\log m)$ când $m \to \infty$. Totuşi, nu este posibil să tragem, pe baza teoremei de aproximare a lui Weierstrass (adică din $\mathcal{E}_m \to 0$, $m \to \infty$), concluzia că interpolarea Lagrange converge uniform pentru orice funcţie $f$, nici chiar pentru noduri judicios alese; se ştie că, de fapt convergenţa nu are loc.

Dacă dorim să utilizăm polinomul de interpolare Lagrange sau Hermite pentru a aproxima funcţia $f$ într-un punct $x \in [a, b]$, distinct de nodurile de interpolare $(x_0, \ldots, x_m)$, trebuie să estimăm eroarea comisă $(R_n f)(x) = f(x) - (H_n f)(x)$. Dacă nu posedăm nici o informaţie referitoare la $f$ în afara punctelor $x_i$, este clar că nu putem spune nimic despre $(R_n f)(x)$; într-adevăr este posibil să schimbăm $f$ în afara punctelor $x_i$ fără a modifica $(H_n f)(x)$. Trebuie deci să facem ipoteze suplimentare, care vor fi ipoteze de regularitate asupra lui $f$. Să notăm cu $C^m[a, b]$ spaţiul funcţiilor reale de $m$ ori continuu diferenţiabile pe $[a, b]$. Avem următoarea teoremă referitoare la estimarea erorii în interpolarea Hermite.

[9] Henry Lebesgue (1875-1941), matematician francez, cunoscut pentru lucrările sale fundamentale în domeniul teoriei funcţiilor reale, şi în special pentru introducerea măsurii şi integralei care îi poartă numele.

**Teorema 3.3.15** *Presupunem că* $f \in C^n[\alpha,\beta]$ *şi există* $f^{(n+1)}$ *pe* $(\alpha,\beta)$*, unde* $\alpha = \min\{x, x_0, \ldots, x_m\}$ *şi* $\beta = \max\{x, x_0, \ldots, x_m\}$*; atunci, pentru orice* $x \in [\alpha,\beta]$*, există un* $\xi_x \in (\alpha,\beta)$ *astfel încât*

$$(R_n f)(x) = \frac{1}{(n+1)!} u_n(x) f^{(n+1)}(\xi_x), \tag{3.3.22}$$

*unde*

$$u_n(x) = \prod_{i=0}^{m} (x - x_i)^{r_{i+1}}.$$

*Demonstraţie.* Dacă $x = x_i$, $(R_n f)(x) = 0$ şi (3.3.22) se verifică trivial. Presupunem că $x$ este distinct de $x_i$ şi considerăm, pentru $x$ fixat, funcţia auxiliară

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

Se observă că $F \in C^n[\alpha,\beta]$, $\exists\ F^{(n+1)}$ pe $(\alpha,\beta)$, $F(x) = 0$ şi $F^{(j)}(x_k) = 0$ pentru $k = \overline{0,m}$, $j = \overline{0,r_k}$. Deci, $F$ are $(n+2)$ zerouri, luând în considerare şi ordinele de multiplicitate. Aplicând succesiv teorema lui Rolle generalizată, rezultă că există cel puţin un $\xi \in (\alpha,\beta)$ astfel încât $F^{(n+1)}(\xi) = 0$, adică

$$F^{(m+1)}(\xi) = \begin{vmatrix} (n+1)! & f^{(n+1)}(\xi) \\ u_n(x) & (R_n f)(x) \end{vmatrix} = 0, \tag{3.3.23}$$

unde s-a ţinut cont că $(R_n f)^{(n+1)} = f^{(n+1)} - (H_n f)^{(n+1)} = f^{(n+1)}$. Exprimând $(R_n f)(x)$ din (3.3.23) se obţine (3.3.22). □

**Corolarul 3.3.16** *Punem* $M_{n+1} = \max\limits_{x\in[a,b]} |f^{(n+1)}(x)|$*; o margine superioară a erorii de interpolare* $(R_n f)(x) = f(x) - (H_n f)(x)$ *este dată prin*

$$|(R_n f)(x)| \le \frac{M_{n+1}}{(n+1)!} |u_n(x)|.$$

Deoarece $H_n$ este proiector, rezultă că $R_n$ este de asemenea proiector; în plus $\mathrm{Ker} R_n = \mathbb{P}_n$, deoarece $R_n f = f - H_n f = f - f = 0$, $\forall f \in \mathbb{P}_n$. Deci, putem aplica lui $R_n$ teorema lui Peano.

**Teorema 3.3.17** *Dacă* $f \in C^{n+1}[a,b]$*, atunci*

$$(R_n f)(x) = \int_a^b K_n(x;t) f^{(n+1)}(t)\,dt, \tag{3.3.24}$$

*unde*

$$K_n(x;t) = \frac{1}{n!} \left\{ (x-t)_+^n - \sum_{k=0}^{m} \sum_{j=0}^{r_k} h_{kj}(x) \left[ (x_k - t)_+^n \right]^{(j)} \right\}. \tag{3.3.25}$$

*Demonstraţie.* Aplicând teorema lui Peano, avem

$$(R_n f)(x) = \int_a^b K_n(x;t) f^{(n+1)}(t) dt$$

şi ţinând cont că

$$K_n(x;t) = R_n\left[\frac{(x-t)_+^n}{n!}\right] = \frac{(x-t)_+^n}{n!} - H_n\left[\frac{(x-t)_+^n}{n!}\right],$$

teorema rezultă imediat. □

Deoarece interpolarea Lagrange este un caz particular al interpolării Hermite pentru $r_i = 0$, $i = 0, 1, \ldots, m$ din teorema 3.3.15 se obţine:

**Corolarul 3.3.18** *Presupunem că $f \in C^m[\alpha, \beta]$ şi există $f^{(m+1)}$ pe $(\alpha, \beta)$, unde $\alpha = \min\{x, x_0, \ldots, x_m\}$ şi $\beta = \max\{x, x_0, \ldots, x_m\}$; atunci, pentru orice $x \in [\alpha, \beta]$, există un $\xi_x \in (\alpha, \beta)$ astfel încât*

$$(R_m f)(x) = \frac{1}{(m+1)!} u_m(x) f^{(m+1)}(\xi_x), \tag{3.3.26}$$

*unde*

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

De asemenea, din teorema 3.3.17 avem:

**Corolarul 3.3.19** *Dacă $f \in C^{m+1}[a, b]$, atunci*

$$(R_m f)(x) = \int_a^b K_m(x;t) f^{(m+1)}(t) dt \tag{3.3.27}$$

*unde*

$$K_m(x;t) = \frac{1}{m!}\left[(x-t)_+^m - \sum_{k=0}^{m} \ell_k(x)(x_k - t)_+^m\right]. \tag{3.3.28}$$

**Exemplul 3.3.20.** Pentru polinoamele de interpolare din exemplul 3.3.9 resturile corespunzătoare sunt

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

şi respectiv

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

**Exemplul 3.3.21.** Restul din formula de interpolare Hermite cu nodurile duble 0 şi 1 pentru $f \in C^4[\alpha, \beta]$ este

$$(R_3 f)(x) = \frac{x^2(x-1)^2}{6!} f^{(4)}(\xi). \qquad \diamondsuit$$

**Exemplul 3.3.22.** Luăm $f(x) = e^x$. Avem pentru $x \in [a, b]$, $M_{n+1} = e^b$ şi oricum am alege punctele $x_i$, $|u_n(x)| \leq (b-a)^{n+1}$, de unde

$$\max_{x\in[a,b]} |(R_n f)(x)| \leq \frac{(b-a)^{n+1}}{(n+1)!} e^b.$$

Se deduce că

$$\lim_{n\to\infty} \left\{ \max_{x\in[a,b]} |(R_n f)(x)| \right\} = \lim_{n\to\infty} \|R_n f\| = 0,$$

adică $H_n f$ converge uniform către $f$ pe $[a, b]$ când $n$ tinde la $\infty$. De fapt se poate demonstra un rezultat analog pentru orice funcţie dezvoltabilă în serie întreagă în jurul punctului $x = \frac{a+b}{2}$ cu raza de convergenţă $r > \frac{3}{2}(b-a)$. $\diamondsuit$

## 3.3.5. Convergenţa interpolării Lagrange

Să definim ce înţelegem prin convergenţă. Presupunem că se dă un tablou triunghiular de noduri de interpolare $x_i = x_i^{(m)}$, având exact $m+1$ noduri distincte pentru orice $m = 0, 1, 2, \ldots$.

$$\begin{array}{ccccccc} x_0^{(0)} & & & & & & \\ x_0^{(1)} & x_1^{(1)} & & & & & \\ x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ x_0^{(m)} & x_1^{(m)} & x_2^{(m)} & \ldots & x_m^{(m)} & & \\ \vdots & \vdots & \vdots & & \vdots & & \end{array} \tag{3.3.29}$$

Presupunem că toate nodurile sunt conţinute într-un interval finit $[a, b]$. Atunci pentru orice $m$ definim

$$P_m(x) = L_m(f; x_0^{(m)}, x_1^{(m)}, \ldots, x_m^{(m)}; x), \quad x \in [a, b]. \tag{3.3.30}$$

Spunem că interpolarea Lagrange bazată pe tabelul de noduri (3.3.29) converge dacă

$$p_m(x) \rightrightarrows f(x), \text{ când } n \to \infty \text{ pe } [a, b]. \tag{3.3.31}$$

Convergenţa depinde evident de comportarea derivatei de ordinul $k$ a lui $f$, $f^{(k)}$, când $k \to \infty$. Presupunem că $f \in C^\infty[a,b]$ şi că

$$|f^{(k)}(x)| \leq M_k \text{ pentru } a \leq x \leq b,\ k = 0, 1, 2, \ldots \tag{3.3.32}$$

Deoarece $|x_i - x_i^{(m)}| \leq b - a$ când $x \in [a,b]$ şi $x_i^{(n)} \in [a,b]$ avem

$$\left|(x - x_0^{(m)}) \ldots (x - x_m^{(m)})\right| < (b-a)^{m+1}, \tag{3.3.33}$$

deci

$$|f(x) - (L_m f)(x)| \leq (b-a)^{m+1} \frac{M_{m+1}}{(m+1)!}, \quad x \in [a,b]. \tag{3.3.34}$$

Deci avem convergenţă dacă

$$\lim_{k\to\infty} \frac{(b-a)^k}{k!} M_k = 0. \tag{3.3.35}$$

Să arătăm că (3.3.35) este adevărată dacă $f$ este analitică într-o vecinătate suficient de mare din $\mathbb{C}$ ce conţine intervalul $[a,b]$. Mai concret fie $C_r$ discul circular (închis) cu centrul în mijlocul intervalului $[a,b]$ şi de rază $r$ şi presupunem că $r > \frac{1}{2}(b-a)$, astfel că $[a,b] \subset C_r$. Presupunem că $f$ este analitică în $C_r$. Atunci putem estima derivata în (3.3.32) cu formula lui Cauchy

$$f^{(k)}(x) = \frac{k!}{2\pi i} \int_{C_r} \frac{f(z)}{(z-x)^{k+1}} dz, \quad x \in [a,b]. \tag{3.3.36}$$

Observând că $|z - x| \geq r - \frac{1}{2}(b-a)$ (vezi figura 3.6) obţinem

$$|f^{(k)}(x)| \leq \frac{k!}{2\pi} \frac{\max\limits_{z\in\partial C_r} |f(z)|}{\left[r - \frac{1}{2}(b-a)\right]^{k+1}} \cdot 2\pi r;$$

putem lua pentru $M_k$ în (3.3.32)

$$M_k = \frac{r}{r - \frac{1}{2}(b-a)} \max_{z\in\partial C_r} |f(z)| \frac{k!}{\left[r - \frac{1}{2}(b-a)\right]^k} \tag{3.3.37}$$

şi (3.3.35) are loc dacă

$$\left(\frac{b-a}{r - \frac{1}{2}(b-a)}\right)^k \to 0 \text{ când } k \to \infty,$$

adică, dacă $b - a < r - \frac{1}{2}(b-a)$ sau echivalent

$$r > \frac{3}{2}(b-a). \tag{3.3.38}$$

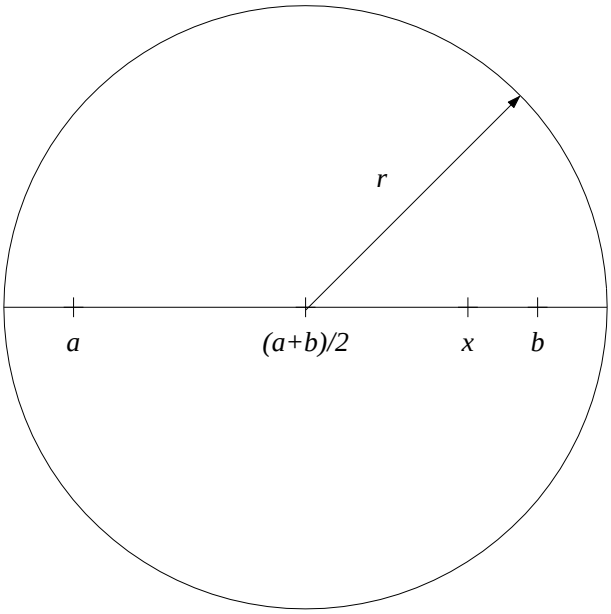



Figura 3.6: Discul circular $C_r$

Am arătat că interpolarea Lagrange converge (uniform pe $[a, b]$) pentru o mulţime arbitrară de noduri (3.3.29) (toate conţinute în $[a, b]$) dacă $f$ este analitică în discul circular $C_r$ centrat în $(a + b)/2$ şi având raza suficient de mare astfel ca (3.3.29) să aibă loc.

Deoarece acest rezultat utilizează o estimare grosieră (în particular (3.3.33), domeniul cerut de analiticitate pentru $f$ nu este prea îngust. Utilizând metode mai rafinate, se poate arăta următorul lucru. Fie $d\mu(t)$ distribuţia limită a nodurilor de interpolare, adică

$$\int_a^x d\mu(t), \quad a < x \leq b,$$

raportul dintre numărul de noduri $x_i^{(m)}$ din $[a, x]$ şi numărul total de noduri, asimptotic când $n \to \infty$. (Când nodurile sunt uniform distribuite pe intervalul $[a, b]$, atunci $d\mu(t) = \frac{dt}{b-a}$). O curbă cu potenţial logaritmic constant este locul geometric al punctelor $z \in \mathbb{C}$ cu proprietatea

$$u(z) = \int_a^b \ln \frac{1}{|z - t|} d\mu(t) = \gamma,$$

unde $\gamma$ este o constantă. Pentru $\gamma$ negativ foarte mare, aceste curbe arată ca nişte cercuri cu raza foarte mare şi centrul în $(a+b)/2$. Pe măsură ce $\gamma$ creşte, curbele se „comprimă”

spre intervalul $[a, b]$. Domeniul important (ce înlocuieşte $C_r$) este domeniul

$$C_\Gamma = \{z \in \mathbb{C} : \ u(z) \geq \Gamma\},$$

în sensul că dacă $f$ este analitică în orice domeniu $C$ ce conţine $C_r$ în interiorul său (nu contează cât de strâns $C$ acoperă pe $C_r$), atunci

$$|f(z) - (L_m f)(z)| \to 0, \text{ când } n \to \infty, \tag{3.3.39}$$

uniform pentru $z \in C_\Gamma$.

**Exemplul 3.3.23.** Noduri echidistante: $d\mu(t) = dt/(b-a)$, $a \leq t \leq b$. În acest caz $C_\Gamma$ este un domeniu în formă de lentilă, aşa cum se arată în figura 3.7. Astfel, avem convergenţa uniformă în $C_\Gamma$ (nu doar pe $[a, b]$ ca mai sus) cu condiţia ca $f$ să fie analitică într-o regiune puţin mai largă decât $C_\Gamma$. Fie $\Gamma = \sup \gamma$, unde supremumul se ia pentru toate curbele $u(z) = \gamma$ ce conţin pe $[a, b]$ în interior.

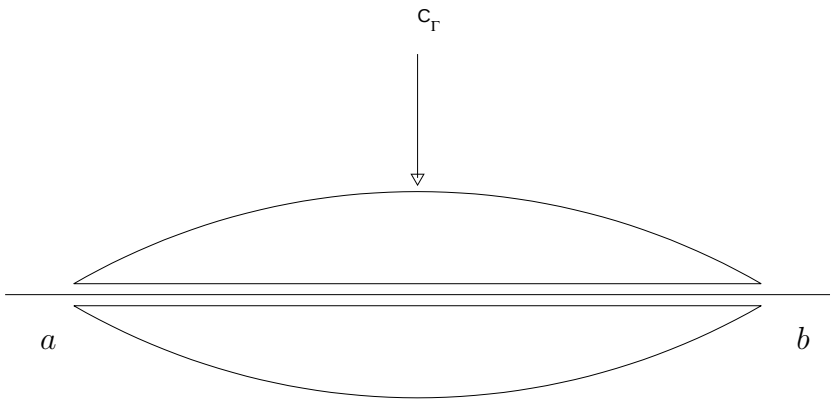



Figura 3.7: Domeniul $C_\Gamma$ pentru noduri uniform distribuite

**Exemplul 3.3.24.** Distribuţia arcsin pe $[-1, 1]$ :

$$d\mu(t) = \frac{1}{\pi} \frac{dt}{\sqrt{1-t^2}}.$$

Nodurile sunt în acest caz rădăcinile polinomului Cebîşev de speţa I. Ele sunt mai dens distribuite în apropierea capetelor intervalului $[-1, 1]$. În acest caz $C_\Gamma = [-1, 1]$, aşa că interpolarea Lagrange converge uniform pe $[-1, 1]$ dacă $f$ este analitică pe $[-1, 1]$, adică în orice regiune al cărei interior conţine intervalul $[-1, 1]$. $\diamondsuit$

Care este semnificaţia polinoamelor Cebîşev pentru interpolare?

Reamintim că eroarea de interpolare (pe [-1,1] pentru o funcţie de clasă $C^{m+1}[-1,1]$) este dată de

$$f(x)-(L_mf)(x)=\frac{f^{(m+1)}(\xi(x))}{(m+1)!}\prod_{i=0}^{m}(x-x_i). \tag{3.3.40}$$

Primul factor este independent de alegerea nodurilor $x_i$. Punctul intermediar $\xi(x)$ depinde de $x_i$, dar de obicei majorăm $|f^{(m+1)}|$ prin $||f^{(m+1)}||_\infty$, ceea ce înlătură această dependenţă. Pe de altă parte, produsul din al doilea factor, inclusiv norma sa

$$\left\|\prod_{i=0}^{m}(.-x_i)\right\|_\infty, \tag{3.3.41}$$

depinde puternic de $x_i$. Are sens, deci, să încercăm să minimizăm (3.3.41) după toţi $x_i\in[-1,1]$. Deoarece produsul din (3.3.41) este un polinom monic de grad $m+1$, din teorema 3.2.1 rezultă că nodurile optimale $x_i=\widehat{x_i}^{(m)}$ din (3.3.40) sunt rădăcinile lui $T_{m+1}$, adică

$$\widehat{x_i}^{(m)}=\cos\frac{2i+1}{2m+2}\pi,\qquad i=\overline{0,m}. \tag{3.3.42}$$

Pentru aceste noduri, avem conform lui (3.2.20)

$$\|f(.)-(L_mf)(.)\|_\infty\le\frac{||f^{(m+1)}||_\infty}{(m+1)!}\cdot\frac{1}{2^m}. \tag{3.3.43}$$

Se cuvine a compara acest factor cu marginea mai brută dată în (3.3.34) care în acest caz pe intervalul $[-1,1]$ este $2^{m+1}/(m+1)!$.

Deoarece conform (3.3.40) curba de eroare $y=f-L_mf$ pentru punctele Cebîşev (3.3.42) este în esenţă *echilibrată* (modulo variaţia factorului $f^{(m+1)}$) şi astfel liberă de oscilaţiile violente pe care le-am văzut pentru puncte echidistante ne vom aştepta la proprietăţi mai favorabile pentru tablouri triunghiulare (3.3.29) formate din noduri Cebîşev. Se poate arăta că dacă $f\in C^1[-1,1]$, atunci

$$(L_mf)\left(x;\widehat{x_0}^{(m)},\widehat{x_1}^{(m)},\dots,\widehat{x_m}^{(m)}\right)\rightrightarrows f(x),\quad n\to\infty, \tag{3.3.44}$$

pe $[-1,1]$. Astfel, nu avem nevoie de analiticitatea lui $f$ pentru ca (3.3.44) să aibă loc.

**Exemplul 3.3.25 (Exemplul lui Runge).** Considerăm funcţia

$$f(x)=\frac{1}{1+x^2},\quad x\in[-5,5]$$

şi nodurile

$$x_k^{(m)} = -5 + 10\frac{k}{m}, \quad k = \overline{0, m}. \tag{3.3.45}$$

Nodurile sunt echidistante pe $[-5, 5]$, deci asimptotic uniform distribuite. Observăm că $f$ are doi poli în $z = \pm i$. Aceşti poli sunt aşezaţi în interiorul regiunii $C_\Gamma$ din figura 3.7 pentru intervalul $[-5, 5]$, deci $f$ nu este analitică în $C_\Gamma$. Din acest motiv nu ne aşteptăm să avem convergenţă pe întreg intervalul $[-5, 5]$. Se poate demonstra că

$$\lim_{m\to\infty} |f(x) - p_m(f; x)| = \begin{cases} 0 & \text{dacă} \quad |x| < 3.633\ldots \\ \infty & \text{dacă} \quad |x| > 3.633\ldots \end{cases} \tag{3.3.46}$$

Având în minte figura 3.7 acest rezultat nu este surprinzător. Graficul pentru $m = 10, 13, 16$ apare în figura 3.8. ♢

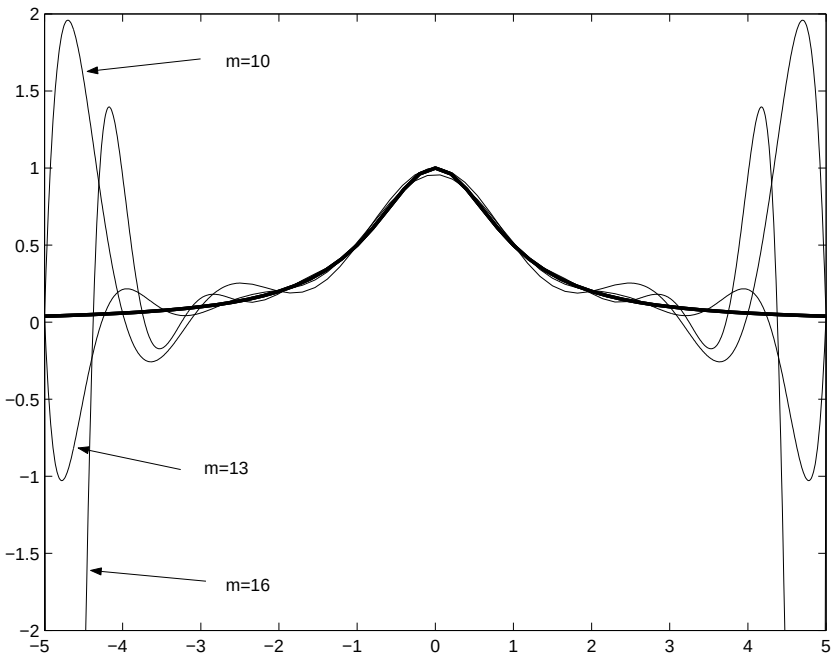



Figura 3.8: O ilustrare grafică a contraexemplului lui Runge

**Exemplul 3.3.26 (Exemplul lui Bernstein).** Luăm funcţia

$$f(x) = |x|, \quad x \in [-1, 1]$$

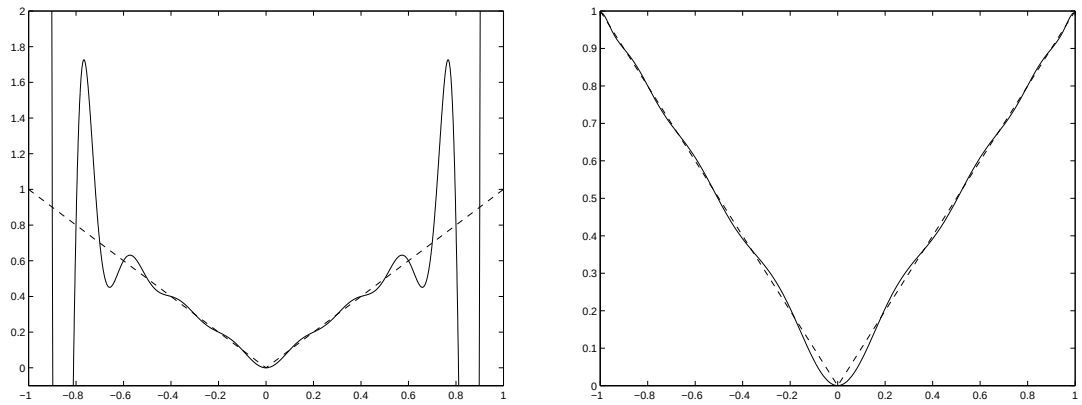
şi nodurile echidistante.

$$x_k^{(m)} = -1 + \frac{2k}{m}, \quad k = 0, 1, 2, \ldots, m. \tag{3.3.47}$$

Problema analiticităţii nu se pune, deoarece $f$ nu este derivabilă în $x = 0$. Se obţine că

$$\lim_{m\to\infty} |f(x) - L_m(f; x)| = \infty \qquad \forall x \in [-1, 1]$$

exceptând punctele $x = -1$, $x = 0$ şi $x = 1$. Vezi figura 3.9(a), pentru $m = 20$. Convergenţa în $x = \pm 1$ este trivială deoarece acestea sunt noduri de interpolare şi deci eroarea în aceste puncte este 0. Acelaşi lucru este adevărat pentru $x = 0$, când $n$ este impar, dar nu şi când $n$ este par. Eşecul convergenţei pentru aceste noduri se explică doar parţial prin insuficienţa regularităţii a lui $f$. Un alt motiv este distribuţia uniformă a nodurilor. Există exemple mai bune de distribuţii ale nodurilor, cum ar fi distribuţia *arcsin* din exemplul 3.3.24. În figura 3.9(b) se dă graficul pentru $m = 17$. ♢



(a) Noduri echidistante, $m = 20$ (b) Noduri Cebîşev, $m = 17$

Figura 3.9: Comportarea interpolării Lagrange pentru $f : [-1, 1] \to \mathbb{R}$, $f(x) = |x|$.

# 3.4. Calculul eficient al polinoamelor de interpolare

## 3.4.1. Metode de tip Aitken

În multe situaţii 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 cunoaştem $\|f^{(m+1)}\|_\infty$. Vom nota cu $P_{m_1,m_2,\ldots,m_k}$ polinomul de interpolare Lagrange având nodurile $x_{m_1}, \ldots, x_{m_k}$.

**Propoziţia 3.4.1** *Dacă $f$ este definită în $x_0, \ldots, x_k$, $x_j \neq x_i$, $0 \leq i, j \leq k$, atunci*

$$P_{0,1,\ldots,k} = \frac{(x - x_j)P_{0,1,\ldots,j-1,j+1,\ldots,k}(x) - (x - x_i)P_{0,1,\ldots,i-1,i+1,\ldots,k}(x)}{x_i - x_j} = \\ = \frac{1}{x_i - x_j} \begin{vmatrix} x - x_j & P_{0,1,\ldots,i-1,i+1,\ldots,k}(x) \\ x - x_i & P_{0,1,\ldots,j-1,j+1,\ldots,k}(x) \end{vmatrix} \tag{3.4.1}$$

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

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

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

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

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

şi

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

deci $P = P_{0,1,\ldots,k}$. □

În acest mod am stabilit o relaţie de recurenţă între un polinom de interpolare Lagrange de gradul $k$ şi două polinoame de interpolare Lagrange de gradul $k-1$. Calculele pot fi aşezate în formă tabelară

$$\begin{array}{llllll} x_0 & P_0 & & & & \\ x_1 & P_1 & P_{0,1} & & & \\ x_2 & P_2 & P_{1,2} & P_{0,1,2} & & \\ x_3 & P_3 & P_{2,3} & P_{1,2,3} & P_{0,1,2,3} & \\ x_4 & P_4 & P_{3,4} & P_{2,3,4} & P_{1,2,3,4} & P_{0,1,2,3,4} \end{array}$$

Să presupunem că în acest moment $P_{0,1,2,3,4}$ nu ne asigură precizia dorită. Se poate selecta un nou nod şi adăuga o nouă linie tabelei

$$x_5 \quad P_5 \quad P_{4,5} \quad P_{3,4,5} \quad P_{2,3,4,5} \quad P_{1,2,3,4,5} \quad P_{0,1,2,3,4,5}$$

iar elementele vecine de pe linie, coloană sau diagonală se pot compara pentru a vedea dacă s-a obţinut precizia dorită.

Metoda de mai sus se numeşte *metoda lui Neville* .
Notaţiile pot fi simplificate

$$Q_{i,j} := P_{i-j,i-j+1,\ldots,i-1,i},$$

$$Q_{i,j-1} = P_{i-j+1,\ldots,i-1,i},$$

$$Q_{i-1,j-1} := P_{i-j,i-j+1,\ldots,i-1}.$$

Din (3.4.1) rezultă

$$Q_{i,j} = \frac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{x_i - x_{i-j}},$$

pentru $j = 1, 2, 3, \ldots$, $i = j + 1, j + 2, \ldots$
În plus, $Q_{i,0} = f(x_i)$. Obţinem tabelul

$$\begin{array}{llllll} x_0 & Q_{0,0} & & & & \\ x_1 & Q_{1,0} & Q_{1,1} & & & \\ x_2 & Q_{2,0} & Q_{2,1} & Q_{2,2} & & \\ x_3 & Q_{3,0} & Q_{3,1} & Q_{3,2} & Q_{3,3} & \end{array}$$

Dacă procedeul de interpolare converge, atunci şirul $Q_{i,i}$ converge şi el şi 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. Ea construieşte tabelul

$$\begin{array}{llllll} x_0 & P_0 & & & & \\ x_1 & P_1 & P_{0,1} & & & \\ x_2 & P_2 & P_{0,2} & P_{0,1,2} & & \\ x_3 & P_3 & P_{0,3} & P_{0,1,3} & P_{0,1,2,3} & \\ x_4 & P_4 & P_{0,4} & P_{0,1,4} & P_{0,1,2,4} & P_{0,1,2,3,4} \end{array}$$

Pentru a calcula o nouă valoare se utilizează valoarea din vârful coloanei precedente şi valoarea din aceeaşi linie, coloana precedentă.

## 3.4.2. Metoda diferenţelor divizate

Vom nota cu $L_k f$ polinomul de interpolare Lagrange cu nodurile $x_0, x_1, \ldots, x_k$ pentru $k = 0, 1, \ldots, n$. Vom construi $L_m$ prin recurenţă. 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 $x_0, x_1, \ldots, x_k$ şi deci este de forma:

$$(L_k f)(x) = (L_{k-1} f)(x) + f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\ldots(x - x_{k-1}), \quad (3.4.2)$$

unde $f[x_0, x_1, \ldots, x_k]$ desemnează coeficientul lui $x^k$ din $(L_k f)(x)$. Se deduce expresia polinomului de interpolare $L_m f$ cu nodurile $x_0, x_1, \ldots, x_n$

$$(L_m f)(x) = f(x_0) + \sum_{k=1}^{m} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\ldots(x - x_{k-1}), \quad (3.4.3)$$

numită forma Newton [10] a polinomului de interpolare Lagrange.

Formula (3.4.3) reduce calculul prin recurenţă al lui $L_m f$ la cel al coeficienţilor $f[x_0, x_1, \ldots, x_k]$, $k = \overline{0, m}$.

Are loc

**Lema 3.4.2**

$$\forall\, k \geq 1 \quad f[x_0, x_1, \ldots, x_k] = \frac{f[x_1, x_2, \ldots, x_k] - f[x_0, x_1, \ldots, x_{k-1}]}{x_k - x_0} \quad (3.4.4)$$

*şi*

$$f[x_i] = f(x_i), \quad i = 0, 1, \ldots, k.$$

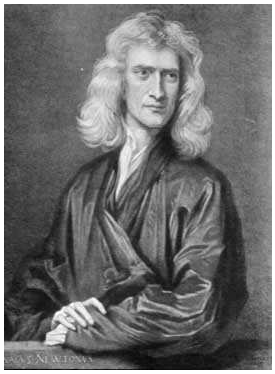
*Demonstraţie.* Notăm, pentru $k \geq 1$ cu $L^*_{k-1} f$ polinomul de interpolare pentru $f$ de grad $k - 1$ şi cu nodurile $x_1, x_2, \ldots, x_k$; coeficientul lui $x^{k-1}$ este $f[x_1, x_2, \ldots, x_k]$. Polinomul $q_k$ de grad $k$ definit prin

$$q_k(x) = \frac{(x - x_0)(L^*_{k-1} f)(x) - (x - x_k)(L_{k-1} f)(x)}{x_k - x_0}$$

coincide cu $f$ în punctele $x_0, x_1, \ldots, x_k$ şi deci $q_k(x) \equiv (L_k f)(x)$. Formula (3.4.4) se obţine identificând coeficientul lui $x^k$ din cei doi membri. □

---

[10] Sir Isaac Newton (1643 - 1727) a fost una dintre cele mai remarcabile figuri ale matematicii şi fizicii din vremea sa. Nu numai că dat legile fundamentale ale fizicii moderne, dar a fost şi unul dintre inventatorii calculului diferenţial şi integral (alături de Leibniz, cu care a intrat într-o polemică de o viată privind prioritatea). Lucrarea sa care a avut cea mai mare influenţă a fost *Principia*, care conţine ideile sale asupra interpolării şi utilizării ei la integrare.



**Definiţia 3.4.3** *Cantitatea* $f[x_0, x_1, \ldots, x_k]$ *se numeşte* diferenţă divizată *de ordinul* $k$ *a lui* $f$ *în punctele* $x_0, x_1, \ldots, x_k$.

Altă notaţie utilizată este $[x_0, \ldots, x_k; f]$.

Din definiţie rezultă că $f[x_0, x_1, \ldots, x_k]$ este independentă de ordinea punctelor $x_i$ şi ea poate fi calculată în funcţie de $f(x_0), \ldots, f(x_m)$. Într-adevăr polinumul de interpolare Lagrange de grad $\leq m$ relativ la punctele $x_0, \ldots, x_m$ se scrie

$$(L_m f)(x) = \sum_{i=0}^{m} l_i f(x_i)$$

şi coeficientul lui $x^m$ este

$$f[x_0, \ldots, x_m] = \sum_{i=0}^{m} \frac{f(x_i)}{\prod\limits_{\substack{j=0 \\ j \neq i}}^{m} (x_i - x_j)}. \tag{3.4.5}$$

Diferenţele divizate se pot obţine prin algoritmul tabelar următor, bazat pe formula (3.4.4), care este mai flexibil şi mai puţin costisitor decât aplicarea formulei (3.4.5)

$$\begin{array}{llllllll}
x_0 & f[x_0] & \rightarrow & f[x_0, x_1] & \rightarrow & f[x_0, x_1, x_2] & \rightarrow & f[x_0, x_1, x_2, x_3] \\
 & & \nearrow & & \nearrow & & \nearrow & \\
x_1 & f[x_1] & \rightarrow & f[x_1, x_2] & \rightarrow & f[x_1, x_2, x_3] & & \\
 & & \nearrow & & \nearrow & & & \\
x_2 & f[x_2] & \rightarrow & f[x_2, x_3] & & & & \\
 & & \nearrow & & & & & \\
x_3 & f[x_3] & & & & & & \\
\vdots & & & & & & &
\end{array}$$

Prima coloană conţine valorile lui $f$, a doua valorile diferenţelor divizate de ordinul I ş.a.m.d; se trece de la o coloană la următoarea utilizând formula (3.4.4): fiecare element este diferenţa dintre elementul situat în stânga şi dedesubtul lui şi elementul situat imediat în stânga lui, împărţită la diferenţa dintre valoarea lui $x$ determinată megând pe diagonală în jos şi valoarea lui $x$ situată la stânga pe orizontală. Diferenţele divizate care apar în formula lui Newton (3.4.3) sunt cele $m + 1$ elemente de pe prima linie a tabelei diferenţelor divizate. Calculul lor necesită $n(n + 1)$ adunări şi $\frac{1}{2}n(n + 1)$ împărţiri. Adăugarea unui nou punct $(x_{m+1}, f[x_{m+1}])$ necesită generarea diagonalei următoare. $L_{m+1}f$ poate fi obţinut din $L_m f$ adăugând termenul $f[x_0, \ldots, x_{m+1}](x - x_0) \ldots (x - x_{m+1})$.

**Observaţia 3.4.4.** Eroarea de interpolare este dată de

$$f(x) - (L_m f)(x) = u_m(x) f[x_0, x_1, \ldots, x_m, x]. \tag{3.4.6}$$

Într-adevăr, este suficient să observăm că

$$(L_m f)(t) + u_m(t) f[x_0, \ldots, x_m; x]$$

este conform lui (3.4.3) polinomul de interpolare (în $t$) al lui $f$ în punctele $x_0$, $x_1$, $\ldots$, $x_m$, $x$. Se deduce din teorema referitoare la restul formulei de interpolare Lagrange (3.3.22) că există $\xi \in (a, b)$ astfel încât

$$f[x_0, x_1, \ldots, x_m] = \frac{1}{m!} f^{(m)}(\xi) \tag{3.4.7}$$

(formula de medie pentru diferenţe divizate). ♢

Diferenţa divizată se poate scrie sub forma unui cât a doi determinanţi.

**Teorema 3.4.5** *Are loc*

$$f[x_0, \ldots, x_m] = \frac{(Wf)(x_0, \ldots, x_m)}{V(x_0, \ldots, x_m)} \tag{3.4.8}$$

*unde*

$$(Wf)(x_0, \ldots, x_n) = \begin{vmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^{m-1} & f(x_0) \\ 1 & x_1 & x_1^2 & \ldots & x_1^{m-1} & f(x_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_m & x_m^2 & \ldots & x_m^{m-1} & f(x_m) \end{vmatrix}, \tag{3.4.9}$$

*iar $V(x_0, \ldots, x_m)$ este determinantul Vandermonde.*

*Demonstraţie.* Se dezvoltă $(Wf)(x_0, \ldots, x_m)$ după elementele ultimei coloane şi ţinând cont că fiecare complement algebric este un determinant Vandermonde, se obţine

$$f[x_0, \ldots, x_m] = \frac{1}{V(x_0, \ldots, x_m)} \sum_{i=0}^{m} V(x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m) f(x_i) =$$

$$= \sum_{i=0}^{m} (-1)^{m-i} \frac{f(x_i)}{(x_i - x_0) \ldots (x_i - x_{i-1})(x_i - x_{i+1}) \ldots (x_n - x_i)},$$

din care după schimbarea semnelor ultimilor $m - i$ termeni rezultă (3.4.5). □

### 3.4.3. Diferenţe finite: formula lui Newton progresivă şi regresivă

În cazul când punctele de interpolare $x_i$ nu sunt echidistante se utilizează algoritmul lui Newton descris anterior; în cazul când punctele sunt echidistante se poate construi un algoritm mai simplu şi mai puţin costisitor. Istoric aceşti algoritmi au avut o mare importanţă pentru interpolarea funcţiilor ale căror valori erau tabelate; apariţia calculatoarelor moderne a diminuat acest interes, dar noile (co)procesoare flotante i-au readus în actualitate.

Presupunem că funcţia $f$ este cunoscută în punctele $x_i$ cu pasul $h$

$$x_i = x_0 + ih, \quad i = 0, 1, \ldots$$

Se definesc *diferenţele progresive* prin

$$\Delta^0 f_i = f(x_i) = f_i,$$

şi $\forall\, k \geq 0$

$$\Delta^{k+1} f_i = \Delta(\Delta^k f_i) = \Delta^k f_{i+1} - \Delta^k f_i.$$

Cu ajutorul formulei (3.4.5) se verifică imediat că

$$\Delta^k f_i = f[x_i, x_{i+1}, \ldots, x_{i+k}]k!h^k.$$

Forma polinomului Newton $L_m f$ de gradul $m$ a lui $f$ în punctele $x_0$, $x_1$, …, $x_m$ se simplifică. Dacă $s = (x - x_0)/h$,

$$(x-x_0)(x-x_1)\ldots(x-x_{k-1})f[x_0, x_1, \ldots, x_k] = \frac{s(s-1)\ldots(s-k+1)}{k!}\Delta^k f_0$$

şi deci

$$\begin{aligned}(L_m f)(x) = f_0 + \frac{s}{1!}\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + \ldots \\ + \frac{s(s-1)\ldots(s-m+1)}{m!}\Delta^m f_0.\end{aligned}$$

Utilizând pentru $s \in \mathbb{R}$, $k \in \mathbb{N}$ notaţia

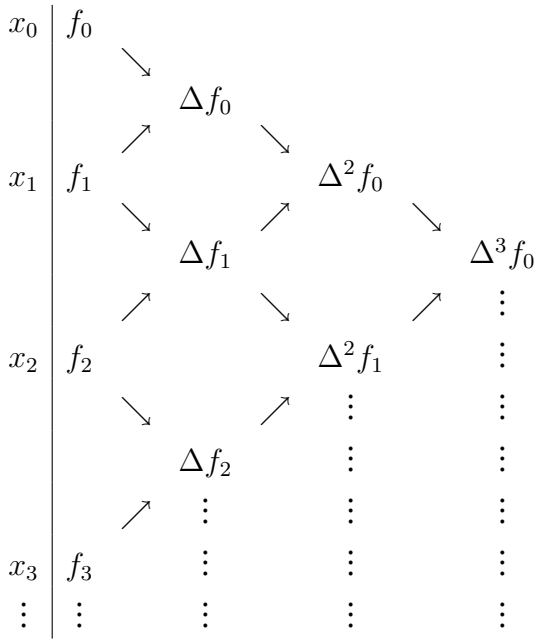
$$\binom{s}{k} = \frac{s(s-1)\ldots(s-k+1)}{k!}$$

(coeficient binomial generalizat), se obţine *formula lui Newton progresivă*

$$(L_m f)(x) = f_0 + \binom{s}{1}\Delta f_0 + \binom{s}{2}\Delta^2 f_0 + \cdots + \binom{s}{n}\Delta^n f_0, \tag{3.4.10}$$

unde $s = (x - x_0)/h$.

În practică această formulă serveşte la calculul lui $(L_m f)(x)$ în puncte apropiate de începutul tabelei. Diferenţele finite succesive se obţin din tabloul triunghiular




unde $\begin{matrix} a \\ \searrow \\ \quad c \\ \nearrow \\ b \end{matrix}$ înseamnă $c = b - a$.

Eroarea comisă după $m$ paşi, se scrie pentru $x = x_0 + sh$

$$f(x) - (L_m f)(x) = h^{m+1} \binom{s}{m+1} f^{(m+1)}(\xi_x), \tag{3.4.11}$$

unde $\xi_x$ aparţine celui mai mic interval ce conţine $x_0, x_m$ şi $x$.

Analog se introduce operatorul de *diferenţă finită regresivă* $\nabla$ prin

$$\begin{aligned} &\nabla^0 f_i = f_i \\ &\nabla^1 f_i = f_i - f_{i-1} \\ &\nabla^{k+1} f_i = \nabla(\nabla^k f_i) = \nabla^k f_i - \nabla^k f_{i-1} \end{aligned}$$

Efectuând schimbarea de variabilă $s = \dfrac{x - x_n}{m}$ se obţine

$$\begin{aligned} (L_m f)(x) = f_m + \frac{s}{1!}\nabla f_m + \frac{s(s+1)}{2!}\nabla^2 f_m + \dots \\ + \frac{s(s+1)\dots(s+n-1)}{m!}\nabla^m f_m, \end{aligned}$$

care se mai poate scrie

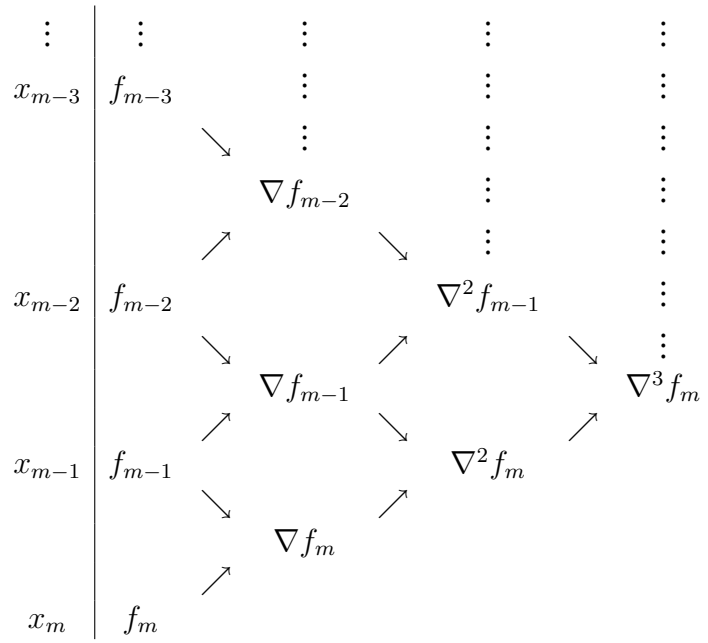
$$(L_m f)(x) = f_m + \binom{s}{1}\nabla f_m + \binom{s+1}{2}\nabla^2 f_m + \cdots + \binom{s+m-1}{m}\nabla^m f_m, \quad (3.4.12)$$

numită *formula lui Newton regresivă* .

Eroarea de interpolare se scrie sub forma

$$f(x) - (L_m f)(x) = h^{m+1}\binom{s+m}{m+1} f^{(m+1)}(\xi_x).$$

unde $\xi_x$ aparţine celui mai mic interval ce conţine $x_0, x_m$ şi $x$. Formula (3.4.12) serveşte pentru calculul polinomului de interpolare în valori apropiate de capătul tabelei. Diferenţele regresive se pot calcula cu tabela




unde $\begin{matrix} a \\ \searrow \\ \quad c \\ \nearrow \\ b \end{matrix}$ înseamnă $c = b - a$.

## 3.4.4. Diferenţe divizate cu noduri multiple

Formula (3.4.8) serveşte ca bază pentru introducerea diferenţei divizate cu noduri multiple: dacă $f \in C^m[a,b]$ şi $\alpha \in [a,b]$, atunci

$$\lim_{x_0,\ldots,x_n \to \alpha} [x_0, \ldots, x_n; f] = \lim_{\xi \to \alpha} \frac{f^m(\xi)}{m!} = \frac{f^{(m)}(\alpha)}{m!}$$

Aceasta justifică relaţia

$$[\underbrace{\alpha,\ldots,\alpha}_{m+1};f]=\frac{1}{m!}f^{(m)}(\alpha).$$

Reprezentând aceasta ca pe un cât de doi determinanţi se obţine

$$(Wf)\left(\underbrace{\alpha,\ldots,\alpha}_{m+1}\right)=\begin{vmatrix} 1 & \alpha & \alpha^2 & \ldots & \alpha^{m-1} & f(\alpha)\\ 0 & 1 & 2\alpha & \ldots & (m-1)\alpha^{m-2} & f'(\alpha)\\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & \ldots & (m-1)! & f^{(m-1)}(\alpha)\end{vmatrix}$$

şi

$$V\left(\underbrace{\alpha,\ldots,\alpha}_{m+1}\right)=\begin{vmatrix} 1 & \alpha & \alpha^2 & \ldots & \alpha^{m}\\ 0 & 1 & 2\alpha & \ldots & m\alpha^{m-1}\\ \ldots & \ldots & \ldots & \ldots & \ldots\\ 0 & 0 & 0 & \ldots & m!\end{vmatrix},$$

adică cei doi determinanţi sunt constituiţi din linia relativă la nodul $\alpha$ şi derivatele succesive ale acesteia până la ordinul $m$ în raport cu $\alpha$.

Generalizarea pentru mai multe noduri este următoarea:

**Definiţia 3.4.6** *Fie $r_k\in\mathbb{N}$, $k=\overline{0,m}$, $n=r_0+\cdots+r_m$. Presupunem că există $f^{(j)}(x_k)$, $k=\overline{0,m}$, $j=\overline{0,r_k-1}$. Mărimea*

$$[\underbrace{x_0,\ldots,x_0}_{r_0},\underbrace{x_1,\ldots,x_1}_{r_1},\ldots,\underbrace{x_m,\ldots,x_m}_{r_m};f]=\frac{(Wf)(x_0,\ldots,x_0,\ldots,x_m,\ldots,x_m)}{V(x_0,\ldots,x_0,\ldots,x_m,\ldots,x_m)},$$

*unde*

$$(Wf)(x_0,\ldots,x_0,\ldots,x_m,\ldots,x_m)=$$

$$=\begin{vmatrix} 1 & x_0 & \ldots & x_0^{r_0-1} & \ldots & x_0^{n-1} & f(x_0)\\ 0 & 1 & \ldots & (r_0-1)x_0^{r_0-2} & \ldots & (n-1)x_0^{n-2} & f'(x_0)\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \ldots & (r_0-1)! & \ldots & \prod_{p=1}^{r_0-1}(n-p)x_0^{n-r_0} & f^{(r_0-1)}(x_0)\\ 1 & x_m & \ldots & x_m^{r_m-1} & \ldots & x_m^{n-1} & f(x_m)\\ 0 & 1 & \ldots & (r_m-1)x_m^{r_m-2} & \ldots & (n-1)x_m^{n-2} & f'(x_m)\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \ldots & (r_m-1)! & \ldots & \prod_{p=1}^{r_{m-1}}(n-p)x_m^{n-r_m} & f^{(r_n-1)}(x_n)\end{vmatrix},$$

*iar $V(x_0,\ldots,x_0,\ldots,x_m,\ldots,x_m)$ este ca mai sus, exceptând ultima coloană care este*

$$(x_0^n,nx_0^{n-1},\ldots,\prod_{p=0}^{r_0-2}(n-p)x_0^{n-r_0+1},\ldots,x_m^n,nx_m^{n-1},\ldots,\prod_{p=0}^{r_m-2}x_m^{n-r_m+1})^T$$

$$
\begin{array}{lll}
z_0 = x_0 & f[z_0] & \\
 & f[z_0, z_1] = f'(x_0) & \\
z_1 = x_0 & f[z_1] & f[z_0, z_1, z_2] = \frac{f[z_1,z_2]-f[z_0,z_1]}{z_2-z_0} \\
 & f[z_1, z_2] = \frac{f(z_2)-f(z_1)}{z_2-z_1} & \\
z_2 = x_1 & f[z_2] & f[x_1, z_2, z_3] = \frac{f[z_3,z_2]-f[z_2,z_1]}{z_3-z_1} \\
 & f[z_2, z_3] = f'(x_1) & \\
z_3 = x_1 & f[z_3] & f[z_2, z_3, z_4] = \frac{f[z_4,z_3]-f[z_3,z_2]}{z_4-z_2} \\
 & f[z_3, z_4] = \frac{f(z_4)-f(z_3)}{z_4-z_3} & \\
z_4 = x_2 & f[z_4] & f[z_3, z_4, z_5] = \frac{f[z_5,z_4]-f[z_4,z_3]}{z_5-z_3} \\
 & f[z_4, z_5] = f'(x_2) & \\
z_5 = x_2 & f[z_5] &
\end{array}
$$

Tabela 3.2: Tabelă de diferenţe divizate pentru noduri duble

*se numeşte* diferenţă divizată cu nodurile multiple $x_k$, $k = \overline{0, m}$ *şi ordinele de multiplicitate* $r_k$, $k = \overline{0, m}$.

Generalizând forma Newton a polinomului de interpolare Lagrange se obţine o metodă bazată pe diferenţele divizate cu noduri multiple pentru polinomul de interpolare Hermite.

Presupunem că se dau nodurile $x_i$, $i = \overline{0, m}$ şi valorile $f(x_i)$, $f'(x_i)$. Definim secvenţa de noduri $z_0, z_1, \ldots, z_{2n+1}$ prin $z_{2i} = z_{2i+1} = x_i$, $i = \overline{0, m}$. Construim acum tabela diferenţelor 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 diferenţă divizată cu nod dublu şi este egală cu $f'(x_i)$, deci vom utiliza $f'(x_0), f'(x_1), \ldots, f'(x_m)$ în locul diferenţelor divizate de ordinul I

$$f[z_0, z_1], f[z_2, z_3], \ldots, f[z_{2m}, z_{2m+1}].$$

Restul diferenţelor se obţin în manieră obişnuită, aşa cum se arată în tabelul 3.2. Ideea poate fi extinsă şi pentru alte interpolări Hermite. Se pare că metoda este datorată lui Powell.

## 3.5. Interpolare spline

Fie $\Delta$ o diviziune a lui $[a, b]$

$$\Delta : \ a = x_1 < x_2 < \cdots < x_{n-1} < x_n = b \tag{3.5.1}$$

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 funcţiile pot fi aproximate arbitrar de bine prin polinoame de grad mic, chiar 0 sau 1.

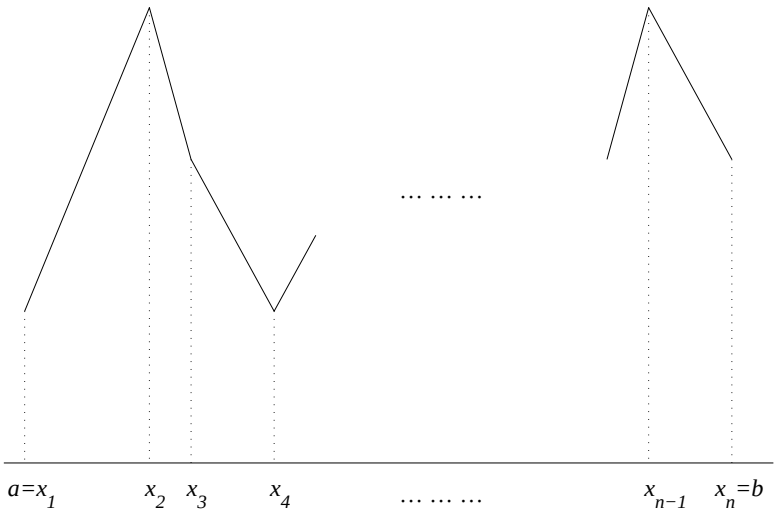



Figura 3.10: Spline liniare

Am introdus în exemplul 3.0.12 spaţiul

$$\mathbb{S}_m^k(\Delta) = \{s:\ s \in C^k[a,b],\ s|_{[x_i,x_{i+1}]} \in \mathbb{P}_m,\ i = 1,2,\ldots,n-1\} \tag{3.5.2}$$

$m \geq 0,\ k \in \mathbb{N} \cup \{-1\}$, numit spaţiul funcţiilor spline polinomiale de grad $m$ şi clasă de netezime $k$. Dacă $k = m$, atunci funcţiile $s \in \mathbb{S}_m^m(\Delta)$ sunt polinoame.

Pentru $m = 1$ şi $k = 0$ se obţin *spline liniare*.

Dorim să găsim $s \in \mathbb{S}_1^0(\Delta)$ astfel încât

$$s(x_i) = f_i, \text{ unde } f_i = f(x_i), \quad i = 1,2,\ldots,n.$$

Soluţia este trivială, vezi figura 3.10. Pe intervalul $[x_i, x_{i+1}]$

$$s(f;x) = f_i + (x - x_i) f[x_i, x_{i+1}], \tag{3.5.3}$$

iar

$$|f(x) - s(f(x))| \leq \frac{(\Delta x_i)^2}{8} \max_{x \in [x_i,x_{i+1}]} |f''(x)|. \tag{3.5.4}$$

Rezultă că

$$\|f(\cdot) - s(f,\cdot)\|_\infty \leq \frac{1}{8}|\Delta|^2 \|f''\|_\infty. \tag{3.5.5}$$

Dimensiunea lui $\mathbb{S}_1^0(\Delta)$ se calculează astfel: deoarece avem $n-1$ porţiuni şi pe fiecare 2 coeficienţi (2 grade de libertate) şi fiecare condiţie reduce numărul de grade de

libertate cu 1, avem în final

$$\dim \mathbb{S}_1^0(\Delta) = 2n - 2 - (n-2) = n.$$

O bază a spaţiului este dată de aşa-numitele funcţii *B-spline*:
Punem $x_0 = x_1,\ x_{n+1} = x_n$, pentru $i = \overline{1,n}$

$$B_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}}, & \text{pentru } x_{i-1} \le x \le x_i \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i}, & \text{pentru } x_i \le x \le x_{i+1} \\ 0, & \text{în rest} \end{cases} \tag{3.5.6}$$

Pentru $i = 1$ prima şi pentru $i = n$ a doua ecuaţie se ignoră.
Funcţia $B_i$ se numeşte *pălărie chinezească.* Graficul funcţiilor $B_i$ apare în figura 3.11.

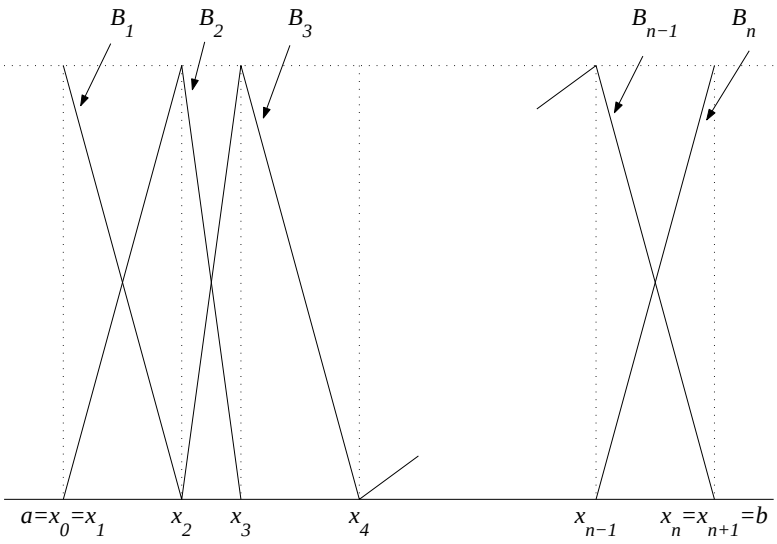



Figura 3.11: Funcţii B-spline de grad 1

Ele au proprietatea

$$B_i(x_j) = \delta_{ij},$$

sunt liniar independente, deoarece

$$s(x) = \sum_{i=1}^{n} c_i B_i(x) = 0 \wedge x \neq x_j \Rightarrow c_j = 0.$$

şi

$$\langle B_i\rangle_{i=\overline{1,n}} = S_1^0(\Delta),$$

$B_i$ joacă acelaşi rol ca polinoamele fundamentale Lagrange $l_i$.

### 3.5.1. Interpolarea cu spline cubice

Funcţiile spline cubice sunt cele mai utilizate.

Vom discuta întâi problema interpolării pentru $s \in \mathbb{S}_3^1(\Delta)$. Continuitatea derivatei de ordinul I pentru $s_3(f;\cdot)$ se poate realiza impunând valorile primei derivate în fiecare punct $x_i$, $i = 1, 2, \ldots, n$. Astfel fie $m_1, m_2, \ldots, m_n$ numere arbitrare date şi notăm

$$s_3(f;\cdot)|_{[x_i,x_{i+1}]} = p_i(x), \quad i = 1, 2, \ldots, n-1 \tag{3.5.7}$$

Realizăm $s_3(f; x_i) = m_i$, $i = \overline{1,n}$, luând fiecare bucată ca soluţie unică a problemei de interpolare Hermite, şi anume

$$\begin{aligned} &p_i(x_i) = f_i, \quad p_i(x_{i+1}) = f_{i+1}, \quad i = \overline{1, n-1}, \\ &p_i'(x_i) = m_i, \quad p_i'(x_{i+1}) = m_{i+1} \end{aligned} \tag{3.5.8}$$

Vom rezolva problema folosind interpolarea Newton. Diferenţele divizate sunt

$$\begin{array}{cccccc} x_i & f_i & & & & \\ & & m_i & & & \\ x_i & f_i & & \frac{f[x_i,x_{i+1}]-m_i}{\Delta x_i} & & \\ & & f[x_i,x_{i+1}] & & \frac{m_{i+1}+m_i-2f[x_i,x_{i+1}]}{(\Delta x_i)^2} & \\ x_{i+1} & f_{i+1} & & \frac{m_{i+1}-f[x_i,x_{i+1}]}{\Delta x_i} & & \\ & & m_{i+1} & & & \\ x_{i+1} & f_{i+1} & & & & \end{array}$$

şi deci forma Newton a polinomului de interpolare Hermite este

$$p_i(x) = f_i + (x-x_i)m_i + (x-x_i)^2\frac{f[x_i,x_{i+1}]-m_i}{\Delta x_i} +$$

$$+(x-x_i)^2(x-x_{i+1})\frac{m_{i+1}+m_i-2f[x_i,x_{i+1}]}{(\Delta x_i)^2}.$$

Forma Taylor a lui $p_i$ pentru $x_i \le x \le x_{i+1}$ este

$$p_i(x) = c_{i,0} + c_{i,1}(x-x_i) + c_{i,2}(x-x_i)^2 + c_{i,3}(x-x_i)^3 \tag{3.5.9}$$

şi deoarece $x - x_{i+1} = x - x_i - \Delta x_i$, prin identificare avem

$$\begin{aligned} c_{i,0} &= f_i \\ c_{i,1} &= m_i \\ c_{i,2} &= \frac{f[x_i,x_{i+1}]-m_i}{\Delta x_i} - c_{i,3}\Delta x_i \\ c_{i,3} &= \frac{m_{i+1}+m_i-2f[x_i,x_{i+1}]}{(\Delta x_i)^2} \end{aligned} \tag{3.5.10}$$

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 coeficienţii cu (3.5.10) şi să evaluăm spline-ul cu (3.5.9). Vom discuta câteva alegeri posibile pentru $m_1, m_2, \ldots, m_n$.

**Interpolare Hermite cubică pe porţiuni**

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 cealaltă. Mai mult, eroarea este

$$|f(x) - p_i(x)| \le \left(\frac{1}{2}\Delta x_i\right)^4 \max_{x\in[x_i,x_{i+1}]} \frac{|f^{(4)}(x)|}{4!}, \quad x_i \le x \le x_{i+1}. \tag{3.5.11}$$

Deci

$$\|f(\cdot) - s_3(f;\cdot)\|_\infty \le \frac{1}{384}|\Delta|^4 \|f^{(4)}\|_\infty. \tag{3.5.12}$$

Pentru puncte echidistante

$$|\Delta| = (b-a)/(n-1)$$

şi deci

$$\|f(\cdot) - s_3(f;\cdot)\|_\infty = O(n^{-4}), \quad n \to \infty. \tag{3.5.13}$$

**Interpolare cu spline cubice**

Cerem ca $s_3(f;\cdot) \in \mathbb{S}_3^2(\Delta)$, adică continuitatea derivatelor de ordinul al II-lea. Aceasta înseamnă cu notaţia (3.5.7)

$$p''_{i-1}(x_i) = p''_i(x_i), \quad i = \overline{2, n-1}, \tag{3.5.14}$$

care convertită în coeficienţi Taylor (3.5.9) 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 (3.5.10) pentru coeficienţi se ajunge la sistemul liniar

$$\Delta x_i m_{i-1} + 2(\Delta x_{i-1} + \Delta x_i)m_i + (\Delta x_{i-1})m_{i+1} = b_i, \quad i = \overline{2, n-1} \tag{3.5.15}$$

unde

$$b_i = 3\{\Delta x_i f[x_{i-1}, x_i] + \Delta x_{i-1} f[x_i, x_{i+1}]\} \tag{3.5.16}$$

Avem un sistem de $n-2$ ecuaţii liniare cu $n$ necunoscute $m_1, m_2, \ldots, m_n$. Odată alese $m_1$ şi $m_n$, sistemul devine tridiagonal şi se poate rezolva eficient prin eliminare gaussiană, prin factorizare sau cu o metodă iterativă.

Se dau în continuare câteva alegeri posibile pentru $m_1$ şi $m_n$.

**Spline complete(racordate, limitate).** Luăm $m_1 = f'(a)$, $m_n = f'(b)$. Se ştie că pentru acest tip de spline, dacă $f \in C^4[a,b]$

$$\|f^{(r)}(\cdot) - s^{(r)}(f;\cdot)\|_\infty \leq c_r |\Delta|^{4-r} \|f^{(n)}\|_\infty, \quad r = 0,1,2,3 \tag{3.5.17}$$

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 condiţiile $s_3''(f;a) = f''(a)$; $s_3''(f;b) = f''(b)$. Aceste condiţii conduc la două ecuaţii suplimentare

$$\begin{aligned} 2m_1 + m_2 &= 3f[x_1,x_2] - \tfrac{1}{2} f''(a)\Delta x_1 \\ m_{n-1} + 2m_n &= 3f[x_{n-1},x_n] + \tfrac{1}{2} f''(b)\Delta x_{n-1} \end{aligned} \tag{3.5.18}$$

Prima ecuaţie se pune la începutul sistemului (3.5.15), iar a doua la sfârşitul lui, păstrându-se astfel structura tridiagonală a sistemului.

**Spline cubice naturale.** Impunând $s''(f;a) = s''(f;b) = 0$, se obţin două ecuaţii noi din (3.5.18) luând $f''(a) = f''(b) = 0$.

Avantajul – este nevoie numai de valori ale lui $f$, nu şi ale derivatelor, dar preţul 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). Cerem ca $p_1(x) \equiv p_2(x)$ şi $p_{n-2}(x) \equiv p_{n-1}(x)$; adică primele două părţi şi respectiv ultimele două trebuie să coincidă. Într-adevăr, aceasta înseamnă că primul punct interior $x_2$ şi ultimul $x_{n-1}$ sunt ambele inactive. Se obţin încă două ecuaţii suplimentare exprimând continuitatea lui $s_3'''(f;x)$ în $x = x_2$ şi $x = x_{n-1}$. Condţia de continuitate a lui $s_3(f,.)$ în $x_2$ şi $x_{n-1}$ revine la egalitatea coeficienţilor dominanţi $c_{1,3} = c_{2,3}$ şi $c_{n-2,3} = c_{n-1,3}$. De aici se obţin ecuaţiile

$$\begin{aligned} (\Delta x_2)^2 m_1 + [(\Delta x_2)^2 - (\Delta x_1)^2] m_2 - (\Delta x_1)^2 m_3 &= \beta_1 \\ (\Delta x_2)^2 m_{n-2} + [(\Delta x_2)^2 - (\Delta x_1)^2] m_{n-1} - (\Delta x_1)^2 m_n &= \beta_2, \end{aligned}$$

unde

$$\begin{aligned} \beta_1 &= 2\{(\Delta x_2)^2 f[x_1,x_2] - (\Delta x_1)^2 f[x_2,x_3]\} \\ \beta_2 &= 2\{(\Delta x_{n-1})^2 f[x_{n-2},x_{n-1}] - (\Delta x_{n-2})^2 f[x_{n-1},x_n]\}. \end{aligned}$$

Prima ecuaţie se adaugă pe prima poziţie iar a doua pe ultima poziţie a sistemului format din cele $n-2$ ecuaţii date de (3.5.15) şi (3.5.16). Sistemul obţinut nu mai este tridiagonal, dar el se poate transforma în unul tridiagonal combinând ecuaţiile 1 cu 2 şi $n-1$ cu $n$. După aceste transformări prima şi ultima ecuaţie devin

$$\Delta x_2 m_1 + (\Delta x_2 + \Delta x_1) m_2 = \gamma_1 \tag{3.5.19}$$

$$(\Delta x_{n-1} + \Delta x_{n-2}) m_{n-1} + \Delta x_{n-2} m_n = \gamma_2, \tag{3.5.20}$$

unde

$$\begin{aligned}\gamma_1 &= \frac{1}{\Delta x_2+\Delta x_1}\left\{f[x_1,x_2]\Delta x_2[\Delta x_1+2(\Delta x_1+\Delta x_2)]+(\Delta x_1)^2 f[x_2,x_3]\right\}\\ \gamma_2 &= \frac{1}{\Delta x_{n-1}+\Delta x_{n-2}}\Big\{\Delta x_{n-1}^2 f[x_{n-2},x_{n-1}]+\\ &\quad [2(\Delta x_{n-1}+\Delta x_{n-2})+\Delta x_{n-1}]\Delta x_{n-2} f[x_{n-1},x_n]\Big\}.\end{aligned}$$

## 3.5.2. Proprietăţi de minimalitate ale funcţiilor spline cubice

Funcţiile spline cubice complete şi naturale au proprietăţi interesante de optimalitate. Pentru a le formula, este convenabil să considerăm nu numai subdiviziunea $\Delta$ ci şi

$$\Delta':\ a=x_0=x_1<x_2<x_3<\cdots<x_{n-1}<x_n=x_{n+1}=b, \tag{3.5.21}$$

în care capetele sunt noduri duble. Aceasta înseamnă că ori de câte ori interpolăm pe $\Delta'$, interpolăm valorile funcţiei pe punctele interioare, iar la capete valorile funcţiei şi ale derivatei. Prima teoremă se referă la funcţii spline cubice complete $s_{compl}(f;\cdot)$.

**Teorema 3.5.1** *Pentru orice funcţie $g\in C^2[a,b]$ care interpolează $f$ pe $\Delta'$, are loc*

$$\int_a^b [g''(x)]^2dx \geq \int_a^b [s''_{compl}(f;x)]^2dx, \tag{3.5.22}$$

*cu egalitate dacă şi numai dacă $g(\cdot)=s_{compl}(f;\cdot)$.*

**Observaţia 3.5.2.** $s_{compl}(f;\cdot)$ din teorema 3.5.1 interpolează $f$ pe $\Delta'$ şi dintre toţi interpolanţii de acest tip, derivata sa de ordinul II are norma minimă. $\diamondsuit$

*Demonstraţie.* Folosim notaţia prescurtată $s_{compl}=s$. Teorema rezultă imediat, dacă arătăm că

$$\int_a^b [g''(x)]^2dx=\int_a^b [g''(x)-s''(x)]^2dx+\int_a^b [s''(x)]^2dx. \tag{3.5.23}$$

Aceasta implică imediat (3.5.22) şi faptul că egalitatea în (3.5.22) are loc dacă şi numai dacă $g''(x)-s''(x)\equiv 0$, din care integrând de două ori de la $a$ la $x$ şi utilizând proprietăţile de interpolare ale lui $s$ şi $g$ în $x=a$ se obţine $g(x)=s(x)$. Relaţia (3.5.23) este echivalentă cu

$$\int_a^b s''(x)[g''(x)-s''(x)]dx=0. \tag{3.5.24}$$

Integrând prin părţi şi ţinând cont că $s'(b)=g'(b)=f'(b)$ şi $s'(a)=g'(a)=f'(a)$ se obţine

$$\int_a^b s''(x)[g''(x)-s''(x)]dx= \tag{3.5.25}$$

$$= s''(x)[g'(x) - s'(x)]\Big|_a^b - \int_a^b s'''(x)[g'(x) - s'(x)]dx =$$

$$= -\int_a^b s'''(x)[g'(x) - s'(x)]dx.$$

Deoarece $s'''$ este constantă pe porţiuni

$$\int_a^b s'''(x)[g'(x) - s'(x)]dx = \sum_{\nu-1}^{n-1} s'''(x_\nu + 0) \int_{x_\nu}^{x_{\nu+1}} [g'(x) - s'(x)]dx =$$

$$= \sum_{\nu=1}^{n-1} s'''(x_{\nu+0})[g(x_{\nu+1}) - s(x_{\nu+1}) - (g(x_\nu) - s(x_\nu))] = 0$$

căci atât $s$ cât şi $g$ interpolează $f$ pe $\Delta$. Aceasta demonstrează (3.5.24) şi deci şi teorema. □

Pentru interpolarea pe $\Delta$, calitatea de a fi optimal revine funcţiilor spline naturale de interpolare $s_{nat}(f;\cdot)$.

**Teorema 3.5.3** *Pentru orice funcţie $g \in C^2[a,b]$ ce interpolează $f$ pe $\Delta$, are loc*

$$\int_a^b [g''(x)]^2 dx \geq \int_a^b [s''_{nat}(f;x)]^2 dx \tag{3.5.26}$$

*cu egalitate dacă şi numai dacă $g(\cdot) = s_{nat}(f;\cdot)$.*

Demonstraţia este analoagă cu a teoremei 3.5.1, deoarece (3.5.23) are loc din nou căci $s''(b) = s''(a) = 0$.

Punând $g(\cdot) = s_{compl}(f;\cdot)$ în teorema 3.5.3 se obţine

$$\int_a^b [s''_{compl}(f;x)]^2 dx \geq \int_a^b [s''_{nat}(f;x)]^2 dx. \tag{3.5.27}$$

Deci, într-un anumit sens, spline-ul cubic natural este cel mai neted interpolant.

Proprietatea exprimată în teorema 3.5.3 stă la originea numelui de spline. Un spline este o vergea flexibilă folosită pentru a desena curbe. Dacă forma sa este dată de ecuaţia $y = g(x)$, $x \in [a,b]$ şi dacă spline-ul trebuie să treacă prin punctele $(x_i, g_i)$, atunci se presupune că spline-ul are o formă ce minimizează energia potenţială

$$\int_a^b \frac{[g''(x)]^2 dx}{(1 + [g'(x)]^2)^3},$$

pentru toate funcţiile $g$ supuse aceloraşi restricţii. Pentru variaţii lente ale lui $g$ ($\|g'\|_\infty \ll 1$) aceasta aproximează bine proprietatea de minim din teorema 3.5.3.

# CAPITOLUL 4

# Aproximare uniformă

## Cuprins



Aproximarea uniformă a funcţiilor are două aspecte fundamentale:

- aproximarea unei funcţii $f$ prin funcţii ce converg uniform către $f$ pe domeniul considerat;
- cea mai bună aproximare uniformă a lui $f$ prin funcţii dintr-o mulţime dată.

Vom pune accentul pe studiul primei probleme.

Fie $\mathcal{B}$ o clasă de funcţii definite pe un interval $[a, b] \subset \mathbb{R}$ şi $\mathcal{A} \subset \mathcal{B}$. Pentru $f \in \mathcal{B}$ şi $\varepsilon \in \mathbb{R}_+$ date, se pune problema determinării unei funcţii $g \in \mathcal{A}$ astfel încât

$$|f(x) - g(x)| < \varepsilon, \qquad x \in [a, b].$$

Această problemă işi are originea în binecunoscuta teoremă de aproximare uniformă a lui Weierstrass [1].

**Teorema 4.0.4 (Weierstrass 1885)** *Fie* $(C[a,b], \|\cdot\|)$. *Subspaţiul* $\mathbb{P} \leq C[a,b]$ *al polinoamelor de grad arbitrar este dens în* $C[a,b]$ $(\bar{\mathbb{P}} = C[a,b])$*, adică* $\forall f \in C[a,b]$ $\forall \varepsilon \in \mathbb{R}_+$ $\exists p \in \mathbb{P}$ *astfel încât* $|f(x) - p(x)| < \varepsilon$*,* $\forall x \in [a,b]$*. (Echivalent* $\exists (p_n) \subset \mathbb{P}$ *astfel încât* $p_n \rightrightarrows f$*.)*

O importanţă deosebită în studiul aproximării uniforme a funcţiilor a avut problema pusă de E. Borel în 1905. Deoarece şirul polinoamelor de interpolare Lagrange $(L_m f)$ nu converge către $f$, el nu poate fi folosit la demonstraţia teoremei lui Weierstrass. Din acest motiv Borel a propus căutarea unor procedee de interpolare mai generale, care să permită contruirea unor şiruri de polinoame $(P_n f)$ ce converg uniform către $f$ dacă $f \in C[a,b]$. Astfel de procedee ar permite să se dea demonstraţii constructive ale teoremei lui Weierstrass. Una dintre primele soluţii pentru problema lui Borel a fost dată de Sergi N. Bernstein în 1912 prin introducerea polinoamelor care în poartă numele.

## 4.1. Polinoamele lui Bernstein

**Definiţia 4.1.1** *Fie* $f : [0,1] \to \mathbb{R}$. *Operatorul* $B_m$ *definit prin relaţia*

$$(B_m f)(x) = \sum_{k=0}^{m} p_{mk}(x) f\left(\frac{k}{m}\right), \tag{4.1.1}$$

*unde*

(4.1.2)

$$p_{m,k}(x) = \binom{m}{k} x^k (1-x)^{m-k} \tag{4.1.3}$$

*se numeşte* operatorul lui Bernstein*, iar polinomul* $B_m f$ *se numeşte* polinomul lui Bernstein.

În figura 4.1 apar polinoamele de bază Bernstein pentru $k = 3$, iar în figura 4.2 polinoamele Bernstein pentru funcţia $f(x) = \sin 2\pi x$ şi $k = 10, 15, 100$.

[1] Karl Theodor Wilhelm Weierstrass (1813-1897), matematician german, considerat a fi unul dintre părinţii analizei moderne. A avut contribuţii importante în teoria funcţiilor de variabile reale, funcţii eliptice, calcul variaţional, studiul formelor biliniare şi pătratice.

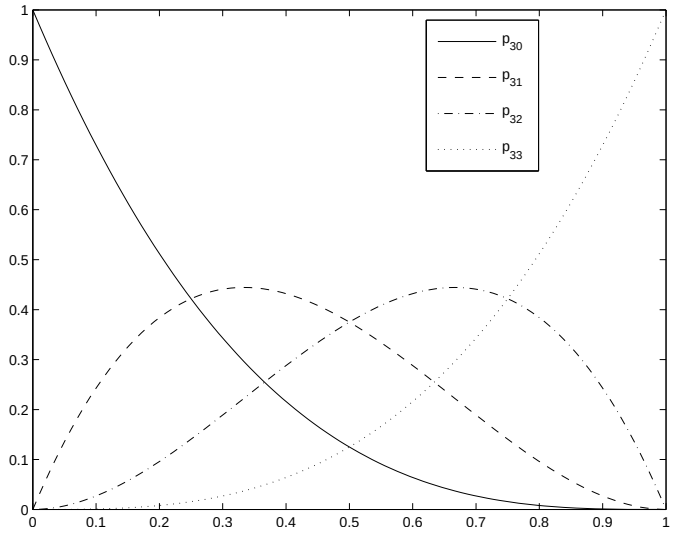



Figura 4.1: Polinoamele de bază Bernstein pentru $k = 3$

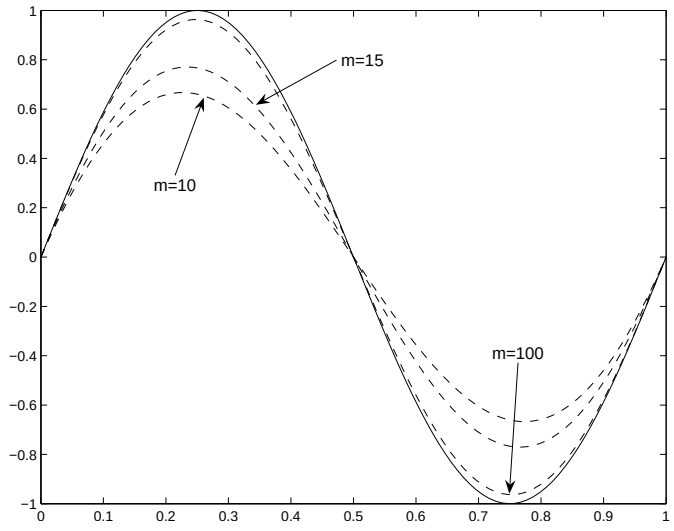



Figura 4.2: Polinoamele Bernstein pentru funcţia $f(x) = \sin 2\pi x$ (linie continuă) şi $k = 10, 15, 100$

Polinomul lui Bernstein poate fi obţinut şi pe cale probabilistică astfel: dacă $f$ este mărginită pe $[0,1]$, se consideră variabila aleatoare $X$ cu distribuţia

$$X : \begin{pmatrix} f(\frac{k}{m}) \\ p_{m,k}(x) \end{pmatrix},$$

a cărei valoare medie $M(x) = \sum_{k=0}^{m} p_{m,k}(x) f(\frac{k}{m})$ este chiar $(B_m f)(x)$.

## Proprietăţi

**Teorema 4.1.2** *1. $B_m$ este un operator liniar.*

*2. $B_m$ este un operator pozitiv, adică*

$$f(x) \geq 0 \Rightarrow (B_m f)(x) \geq 0, \forall x \in [0,1].$$

*3. $B_m$ reproduce polinoamele până la gradul 1 inclusiv, adică*

$$(B_m e_k)(x) = e_k(x), \qquad k = 0, 1$$

*şi în plus*

$$(B_m e_2)(x) = e_2(x) + \frac{x(1-x)}{m}.$$

*4. Dacă $m \leq f \leq M$ pe $[0,1]$, atunci $m \leq B_m f \leq M$ pe [0,1].*

*5. Dacă $f \in C[0,1]$, atunci $B_m f \rightrightarrows f$ pe [0,1] când $m \to \infty$.*

*Demonstraţie.* 1. Rezultă imediat din definiţie.

2. Imediată ţinând cont că $\forall x \in [0,1]\ p_m(x) \geq 0$.

3. Se foloseşte binomul lui Newton

$$(u+v)^m = \sum_{k=0}^{m} \binom{m}{k} u^k v^{m-k}. \tag{4.1.4}$$

Diferenţiem succesiv în raport cu $u$ egalitatea de mai sus

$$u(u+v)^{m-1} = \sum_{k=0}^{m} \frac{k}{m} u^k v^{m-k} \binom{m}{k} \tag{4.1.5}$$

$$\left(1 - \frac{1}{m}\right) u^2 (u+v)^{m-2} = \sum_{k=0}^{m} \left[\frac{k^2}{m^2} - \frac{k}{m^2}\right] \binom{m}{k} u^k v^{m-k}. \tag{4.1.6}$$

Punând în ultimele trei relaţii $u = x$ şi $v = 1 - x$ vom obţine

$$\sum_{k=0}^{m} p_{mk}(x) = 1, \qquad \sum_{k=0}^{m} \frac{k}{m} p_{mk}(x) = x,$$
$$\sum_{k=0}^{m} \left(\frac{k}{m}\right)^2 p_{mk}(x) = x^2 + \frac{x(1-x)}{m}.$$

4. Din $f - m \geq 0$, prin pozitivitate, rezultă $B_m(f - m) \geq 0$. Analog şi cealată inegalitate.

5. $f \in C[0,1] \Rightarrow f$ uniform continuă pe $[0,1]$, adică $\forall \varepsilon > 0\ \exists \delta = \delta_\varepsilon$ astfel încât $\forall x, x' \in [0,1]$ cu $|x - x'| < \delta$ să avem $|f(x) - f(x')| < \frac{\varepsilon}{2}$. Folosind a treia proprietate putem scrie

$$f(x) - (B_m f)(x) = \sum_{k=0}^{m} p_{mk}(x) \left[ f(x) - f\left(\frac{k}{m}\right) \right] \text{ şi}$$
$$|f(x) - (B_m f)(x)| \leq \sum_{k=0}^{m} p_{mk}(x) \left| f(x) - f\left(\frac{k}{m}\right) \right| =$$
$$= \sum_{k \in I_m} p_{mk}(x) \left| f(x) - f\left(\frac{k}{m}\right) \right| + \sum_{k \in J_m} p_{mk}(x) \left| f(x) - f\left(\frac{k}{m}\right) \right|$$

unde

$$I_m = \left\{ k : \left| \frac{k}{m} - x \right| < \delta \right\} \text{ iar } J_m = \left\{ k : \left| \frac{k}{m} - x \right| \geq \delta \right\}.$$

Dacă $M = \max_{x \in [0,1]} |f(x)|$ obţinem în continuare

$$|f(x) - (B_m f)(x)| \leq \frac{\varepsilon}{2} \sum_{k=0}^{m} p_{mk}(x) + 2M \sum_{k \in J_m} p_{mk}(x)$$
$$\leq \frac{\varepsilon}{2} + 2M \sum_{k \in J_m} p_{mk}(x)$$

Deoarece

$$\left| \frac{k}{m} - x \right| \geq \delta \Rightarrow \frac{\left(\frac{k}{m} - x\right)^2}{\delta^2}$$

obţinem succesiv, folosind proprietatea 3

$$\sum_{k\in J_m} p_{mk}(x) \le \frac{1}{\delta^2}\sum_{k\in J_m}\left(\frac{k}{m}-x\right)^2 p_{mk}(x) \le \frac{1}{\delta^2}\sum_{k=0}^{m}\left(\frac{k}{m}-x\right)^2 p_{mk}(x)$$
$$= \frac{1}{\delta^2}\left(\underbrace{\sum_{k=0}^{m}\frac{k^2}{m^2}p_{mk}(x)}_{x^2+\frac{x(1-x)}{m}} - 2x\underbrace{\sum_{k=0}^{m}\frac{k}{m}p_{mk}(x)}_{x} + x^2\underbrace{\sum_{k=0}^{m}p_{mk}(x)}_{1}\right)$$
$$= \frac{1}{\delta^2}\frac{x(1-x)}{m} \le \frac{1}{4m\delta^2}, \qquad x\in[0,1].$$

Deci, $|f(x)-(B_mf)(x)| \le \frac{\varepsilon}{2} + \frac{M}{2m\delta^2}$, adică

$$|f(x)-(B_mf)(x)| \le \varepsilon, \text{ dacă } m > \frac{M}{\varepsilon\delta^2}, \qquad x\in[0,1].$$

Această proprietate este o demonstraţie constructivă a teoremei lui Weierstrass. □

**Definiţia 4.1.3** *Formula*

$$f = B_mf + R_mf \tag{4.1.7}$$

*se numeşte* formula de aproximare a lui Bernstein *iar* $R_mf$ termenul rest.

**Teorema 4.1.4** *Dacă* $f\in C^2[0,1]$, *atunci*

$$(Rmf)(x) = -\frac{x(1-x)}{2m}f''(\xi), \qquad \xi\in[0,1] \tag{4.1.8}$$

*şi*

$$|(R_mf)(x)| \le \frac{1}{8m}\|f''\|_\infty. \tag{4.1.9}$$

*Demonstraţie.* Conform teoremei 4.1.2, proprietăţile 1 şi 2, $B_mf$ reproduce polinoamele de grad 1, adică $B_mf = f$, pentru orice $f\in\mathbb{P}_1$. Aplicând teorema lui Peano se obţine

$$(R_mf)(x) = \int_0^1 K(x;t)f''(t)\,\mathrm{d}t,$$

unde

$$K(x;t) = (x-t)_+ - \sum_{k=0}^{m} p_{m,k}(x) \left(\frac{k}{m} - t\right)_+ .$$

Cum $K(x;t) \leq 0$ pentru $x, t \in [0,1]$ corolarul la teorema lui Peano ne conduce la

$$(R_m f)(x) = -\frac{x(1-x)}{2m} f''(\xi), \qquad \xi \in [0,1].$$

Inegalitatea (4.1.9) rezultă din faptul că $x(1-x) \leq \frac{1}{4}$ pe $[0,1]$. □

Importanţă teoremei lui Weierstass a făcut ca ea să fie mereu în atenţia matematicienilor. S-au dat numeroase demonstraţii constructive şi generalizări. Una dintre ele este teorema lui Stone şi Weierstrass.

Fie $D$ un compact, $C(D)$ spaţiul funcţiilor definite pe compactul $D$ înzestrat cu norma $\|f\| = \max_D |f(x)|$. $C(D)$ este o algebră.

Enunţăm fără demonstraţie:

**Teorema 4.1.5 (Stone-Weierstrass, 1948)** *Dacă $A$ este o subalgebră a lui $C(D)$ care separă punctele lui $D$, atunci închiderea $\bar{A}$ a lui $A$ este fie $C(D)$, fie mulţimea funcţiilor continue care se anulează într-un punct al lui D.*

**Observaţia 4.1.6.** 1. Dacă $A$ conţine funcţiile constante, atunci $\bar{A} = C(D)$.

2. O familie $S$ de funcţii definite pe $E$ separă punctele lui $E$ dacă $\forall u, v \in E, u \neq v, \exists h \in S$ astfel încât $h(u) \neq h(v)$. ♢

Pentru detalii asupra demonstraţiilor teoremei lui Weierstrass şi asupra teoremei Stone-Weierstrass a se vedea [39].

# 4.2. B-spline

## 4.2.1. Noţiuni şi rezultate de bază

Fie $m$,$n \in \mathbb{N}$ şi punctele $t_0$, $t_1$, …, $t_m$, cu $t_i \leq t_{i+1}$. Fie $[a,b] \subset \mathbb{R}$ astfel încât $t_k \leq a$ şi $t_{n-k} \geq b$. Pentru $j = \overline{1, m-i}$ punem

$$\omega_{i,j}(x) = \begin{cases} \dfrac{x - t_i}{t_{i+j} - t_i}, & \text{dacă } t_i < t_{i+j} \\ 0, & \text{altfel.} \end{cases} \tag{4.2.1}$$

**Definiţia 4.2.1** *Pentru* $i = \overline{0, m-k-1}$ funcţiile B-spline *de rang* $i$ *şi grad* $k$ *se definesc prin recurenţă astfel*

$$B_{i,0}(x) = \begin{cases} 1, & \textit{dacă } x \in [t_i, t_{i+1}], \\ 0, & \textit{în caz contrar.} \end{cases} \tag{4.2.2}$$

$$B_{i,k}(x) = \omega_{i,k}(x) B_{i,k-1}(x) + (1 - \omega_{i+1,k}(x)) B_{i+1,k-1}(x), \quad \textit{pentru } k \geq 1. \tag{4.2.3}$$

**Propoziţia 4.2.2** *Au loc următoarele proprietăţi:*

*(a)* $B_{i,k}$ *este un polinom de grad* $k$ *pe porţiuni;*

*(b)* $B_{i,k}(x) = 0$ *pentru* $x \notin [t_i, t_{i+k+1})$*;*

*(c)* $B_{i,k}(x) > 0$ *pentru* $x \in (t_i, t_{i+k+1})$*;* $B_{i,k}(t_i) = 0$ *înafara cazului când* $t_i = t_{i+1} = \cdots = t_{i+k} < t_{i+k+1}$ *(nod de ordin* $k+1$*) când avem* $B_{i,k}(t_i) = 1$*.*

*(d)*

$$\sum_{i=0}^{m-k-1} B_{i,k}(x) = 1, \qquad \forall x \in [a, b).$$

*(e) Fie* $x \in (t_i, t_{i+k+1})$*;* $B_{i,k}(x) = 1 \Longleftrightarrow t_{i+1} = \cdots = t_{i+k} = x$*;*

*(f)* $B_{i,k}$ *este continuă (şi indefinit derivabilă) la dreapta* $\forall x \in \mathbb{R}$ *(precizăm că* $B_{i,k}(x) = 0$*,* $\forall x \in [t_0, t_m]$*.*

*Demonstraţie.* (a), (b), (c), (d) şi (f) sunt evidente pentru $k = 0$; (a), (b), (c) şi (f) pentru $B_{i,k}$ se deduc prin recurenţă după $k$. Să demonstrăm (d). Fie $x \in [a, b]$; există un $j$, $k \leq j \leq m-k-1$ astfel încât $x \in [t_j, t_{j+1})$. Dacă $x = t_j$ şi $B_{j,k}(x) = 1$ proprietatea este evidentă (conform (c)). În celelalte cazuri avem

$$\begin{aligned} \sum_{i=0}^{m-k-1} B_{i,k}(x) &= \sum_{i=j-k}^{j} B_{i,k}(x) \\ &= \sum_{i=j-k}^{j} \left[\omega_{i,k}(x) B_{i,k-1}(x) + (1 - \omega_{i+1,k}(x)) B_{i+1,k-1}(x)\right] \\ &= \omega_{j-k,k}(x) B_{j-k,k-1}(x) + \sum_{i=j+1-k}^{j} B_{i,k-1}(x) \\ &\quad + (1 - \omega_{j+1,k}(x)) B_{j+1,k-1}(x). \end{aligned}$$

Dar $B_{j-k,k-1}(x) = 0$ şi $B_{j+1,k-1}(x) = 0$, întrucât $x \in [t_j, t_{j+1})$ (conform lui (b)) şi

$$\sum_{i=j+1-k}^{j} B_{i,k-1}(x) = 1$$

(din ipoteza inducţiei).
(e) este o consecinţă imediată a lui (d) şi (c). □

**Observaţia 4.2.3.** 1. Deoarece B-splinele sunt nenegative ((b) şi (c)) şi suma lor este 1, spunem că ele formează o *partiţie a unităţii*.

2. Fiecare $B_{i,k}$ are un suport unic.

3. În cazul când $t_{m-k} = \cdots = t_m = b$, pentru ca (d) şi (e) să fie adevărate pe întreg $[a, b]$ punem $B_{m-k-1,k}(b) = 1$.

4. B-splinele se pot defini şi pentru un şir infinit de noduri $t_i$, în definiţia fiecărui $B_{i,k}$ intervenind un număr finit de noduri.

5. Dacă $t_i$ este un nod de multiplicitate $\geq k + 2 (t_i = t_{i+k+1})$ are loc $B_{i,k} \equiv 0$; reciproca este de asemenea adevărată. ♢

**Propoziţia 4.2.4 (Identitatea lui Marsden)** *Fie $n = m - k$ şi*

$$\Psi_{i,k}(t) = \begin{cases} (t_{i+1} - t) \dots (t_{i+k} - t), & \textit{pentru } k \geq 1; \\ 1, & \textit{pentru } k = 0. \end{cases} \tag{4.2.4}$$

*Are loc relaţia*

$$(x - t)^k = \sum_{i=0}^{n-1} \Psi_{i,k}(t) B_{i,k}(x). \tag{4.2.5}$$

*Demonstraţie.* Se face prin inducţie după $k$. Cazul $k = 0$ este trivial (propoziţia 4.2.2 (d)). Presupunem că

$$\sum_{i=0}^{n-1} \Psi_{i,k}(t) B_{i,k}(x) = (x - t)^{k-1}$$

este adevărată. Avem $B_{0,k-1}(x) = 0$ pentru $x \in [a, b]$.

$$\sum_{i=0}^{n-1}\Psi_{i,k}(t)B_{i,k}(x)=$$

$$\begin{aligned}&=\sum_{i=0}^{n-1}\Psi_{i,k}(t)\left[\omega_{i,k}(x)B_{i,k-1}(x)+(1-\omega_{i+1,k}(x))B_{i+1,k-1}(x)\right]\\&=\Psi_{0,k}(t)\omega_{0,k}(x)B_{0,k-1}(x)\\&+\sum_{i=1}^{n-1}B_{i,k-1}(x)\left[\Psi_{i,k}(t)\omega_{i,k}(x)+\Psi_{i-1,k}(t)(1-\omega_{i,k}(x))\right].\end{aligned}$$

Dacă $t_i=t_{i+k}$, atunci $B_{i,k-1}\equiv 0$ şi dacă $t_i<t_{i+k}$ avem

$$\begin{aligned}&\Psi_{i,k}(t)(t)\omega_{i,k}(x)+\Psi_{i-1,k}(t)(1-\omega_{i,k}(x))=\\&=\Psi_{i,k-1}(t)\left[(t_{i+k}-t)\omega_{i,k}(x)+(t_i-t)(1-\omega_{i,k}(x))\right]\\&=\Psi_{i,k-1}(t)\left(\omega_{i,k}(x)(t_{i+k}-t_i)+t_i-t\right)=\Psi_{i,k-1}(t)(x-t).\end{aligned}$$

Deci

$$\sum_{i=0}^{n-1}\Psi_{i,k}(t)B_{i,k}(x)=(x-t)\sum_{i=1}^{n-1}\Psi_{i,k-1}(t)B_{i,k-1}(x)=(x-t)^k,$$

ultima egalitate avân loc pe baza ipotezei inducţiei. □

**Observaţia 4.2.5.** Fie $t=(t_0,\ldots,t_m)$ şi $\mathbb{P}_{k,t}([a,b])=\mathbb{P}_{k,t}$ spaţiul funcţiilor polinomiale pe porţiuni pe $[a,b]$ de grad $\leq k$ şi cu racord de clasă $C^{k-p_j}$ în $t_j$ dacă $t_j$ este un nod de multiplicitate $p_j$. Prin convenţie, un racord de clasă $C^{k-p_j}$, cu $k-p_j<0$ nu impune nici un fel de condiţii asupra lui $t_j$.
Dacă toate nodurile sunt de multiplicitate $\leq k+1$, atunci funcţiile $B_{i,k}$, $i=\overline{0,n-1}$ formează o bază a lui $\mathbb{P}_{k,t}$ şi $\dim\mathbb{P}_{k,t}=n=m-k$. Dacă anumite noduri au multiplicitate $\geq k+2$, atunci $\mathbb{P}_{k,t}=\left\langle B_{i,k}|i=\overline{0,n-1}\right\rangle$, dar $\{B_{i,k}\}$ nu formează o bază. ♢

## 4.2.2. Algoritmul de evaluare a unui B-spline

Fie $x\geq t_k$ şi $S(x)=\sum_{i=0}^{n-1}a_iB_{i,k}(x)$. Aplicând direct definiţia inductivă a B-splinelor se obţine:

**Propoziţia 4.2.6** *Are loc*

$$S(x)=\sum a_i^{(0)}(x)B_{i,k}(x)=\sum a_i^{(1)}(x)B_{i,k-1}(x)=\cdots=\sum a_i^{(k)}(x)B_{i,0}(x)$$

*cu*

$$\begin{aligned}a_i^{(0)}(x)&=a_i\\a_i^{(r+1)}(x)&=\omega_{i,k-r}(x)a_i^{(r)}(x)+(1-\omega_{i,k-r}(x))a_{i-1}^{(r)}(x).\end{aligned}$$

**Observaţia 4.2.7.** 1. Algoritmul este valabil şi pentru $x < t_k$, considerând că $B_{i,k} \equiv 0$ pentru $i \leq 0$.

2. Dacă evaluăm $S(x)$ pentru $x \in [t_j, t_{j+1})$, avem $S(x) = a_j^k(x)$, deoarece $B_{j,0}(x)$ este funcţia caracteristică a intervalului $[t_j, t_{j+1})$ şi pentru a calcula $a_j^{(k)}(x)$ este suficient să evaluăm $a_i^{(r)}$ numai pentru $j - k + r - 1 < i \leq j$, celelalte $B_{i,k-r}(x)$ fiind nule pentru $x \in [t_j, t_{j+1})$. Avem aşadar

$$S(x) = \sum_{i=j-k}^{j} a_i^{(0)}(x) B_{i,k}(x) = \sum_{i=j-k+1}^{j} a_i^{(1)}(x) B_{i,k-1}(x) = \cdots = a_j^{(k)}(x). \qquad \diamondsuit$$

Algoritmul se prezintă, aşadar, sub formă triunghiulară, fiecare element obţinându-se prin combinaţia convexă a două elemente din linia anterioară:

$$\begin{array}{ccccccccccc}
a_{j-k} & & a_{j-k+1} & & \cdots & & a_{j-1} & & a_j \\
& a_{j-k+1}^{(1)} & & \cdots & & \cdots & & a_j^{(1)} & \\
& & \ddots & & & & \iddots & & \\
& & & a_{j-1}^{(k-1)} & & a_{j-1}^{(k-1)} & & & \\
& & & & a_j^{(k)} & & & &
\end{array}$$

(Algoritmul Cox-DeBoor)

Nodurile $t$ formează o diviziune

$$\Delta : \ t_0 \leq t_1 \cdots \leq t_k \leq a \leq \cdots \leq b \leq t_n \leq t_{n+1} \leq t_{n+k}.$$

Vom presupune că multiplicitatea oricărui nod $\leq k + 1$. O alegere frecventă este

$$t_0 = t_1 = \cdots = t_k = a < t_{k+1} \leq \cdots \leq t_{n-1} < b = t_n = \cdots = t_{n+k}.$$

În cazul particular $t_0 = \cdots = t_k = a < b = t_{k+1} = \cdots = t_{2k+1}$, funcţiile B-spline devin polinoame fundamentale Bernstein

$$p_{i,k}(x) = \binom{k}{i} x^i (1-x)^{k-i}.$$

## 4.2.3. Aplicaţii în grafica pe calculator

### Curbe B-spline şi Bézier

Fie $P_0, \ldots, P_{n-1} \in \mathbb{R}^s$, $s \in \mathbb{N}^*$.

**Definiţia 4.2.8** *(a) Se numeşte* curbă B-spline *asociată poligonului de control* $(P_0, \ldots, P_{n-1})$ *curba parametrică* $(\gamma)$ *definită prin*

$$(\gamma) \qquad S(t) = \sum_{i=0}^{n-1} P_i B_{i,k}(t), \qquad a \leq t \leq b.$$

*(b) Se numeşte* curbă Bézier [2] *asociată poligonului de control* $(P_0, \ldots, P_{n-1})$ *curba parametrică*

$$B(t) = \sum_{i=0}^{k} P_i p_{i,k}(t), \qquad p_{i,k}(x) = \binom{k}{i} x^i (1-x)^{k-i}.$$

**Propoziţia 4.2.9** *Curbele B-spline au următoarele proprietăţi:*

1. $\gamma$ *nu trece în general prin punctele* $P_i$*; dacă* $t_0 = \cdots = t_k$ *şi* $t_n = \cdots = t_{n+k} = b$ *avem* $S(a) = P_0$ *şi* $S(b) = P_{n-1}$*; în acest caz* $\gamma$ *este tangentă în* $P_0$ *şi* $P_{n-1}$ *la laturile poligonului de control.*

2. $\gamma$ *este în învelitoarea convexă a punctelor* $P_0, \ldots, P_{n-1}$*. Mai exact dacă* $t_i \leq t \leq t_{i+1}$*,* $S(t)$ *este în învelitoarea convexă a punctelor* $P_{i-k}, \ldots, P_i$*. Curba Bézier este situată în învelitoarea convexă a punctelor* $P_0, \ldots, P_{n-1}$*.*

3. *În cazul când nodurile* $t_i$ *(*$k+1 \leq i \leq n-1$*) sunt simple,* $\gamma \in C^{k-1}$ *şi este formată din* $n$ *arce parametrice polinomiale, de grad* $\leq k$*.*

4. $\gamma$ *este invariantă la deplasările lui* $\mathbb{R}^s$*: dacă* $h$ *este o deplasare a lui* $\mathbb{R}^s$*, atunci punctele* $h(P_i)$ *sunt punctele de control ale curbei* $h(S(t))$*.*

5. $S(t)$ *este regularizantă: dacă* $P$ *este poligonul de control al lui* $\gamma$ *şi dacă* $H$ *este un hiperplan, avem* $\mathrm{card}(H \cap \gamma) \leq \mathrm{card}(H \cap P)$*.*

**Observaţia 4.2.10.** Curbele B-spline au un caracter local în sensul că:

---

[2] Piérre Bezier(1910-1999) Fiu şi nepot de ingineri, a studiat ingineria mecanică la Universitatea din Paris, unde în 1977 a obţinut doctoratul în Matematică . Timp de 42 de ani (1933-1975) a lucrat la Uzinele Renault. A introdus curbele care îi poartă numele. Este considerat fondator al domeniului CAD-CAM, realizând primul sistem de acest tip (UNISURF - incepând cu 1960). Laureat al mai multor premii presigioase, printre care ,,Steven Anson Coons" al ACM (Association for Computing Machinery).

(a) Dacă $X$ este un punct al curbei corespunzând unei valori $t_0$ a parametrului, atunci poziţia lui $X$ nu depinde decât de cel mult $k+1$ puncte $P_i$, deoarece dacă $t_j \leq t_0 < t_{j+1}$ avem

$$X = S(t_0) = \sum_{i=0}^{n-1} P_i B_{i,k}(t_0) = \sum_{i=j-k}^{k} P_i B_{i,k}(t_0).$$

(b) Fiecare $P_j$ nu influenţează curba $S(t)$ decât pentru valorile lui $t$ pentru care $B_{j,k}(t) \neq 0$, adică $t_j \leq t < t_{j+k+1}$ (în particular dacă modificăm $P_j$ numai o porţiune a curbei se modifică.

(c) Caracterul local nu are loc în cazul curbelor Bézier: modificarea unui punct de control atrage modificarea întregii curbe. ♢

**Algoritmi pentru curbe B-spline**

Algoritmul 4.1 de evaluare a curbelor B-spline este o simplă traducere a algoritmului de evaluare pentru funcţii B-spline.

**Algoritmul 4.1** Algoritmul Cox-deBoor pentru evaluarea unei curbe B-spline

**Intrare:** Poligonul de contrul $P_i$, $i = \overline{0, n-1}$, punctul t în care se face evaluarea şi diviziunea $t_i$
**Ieşire:** Valoarea $S(t)$

Fie $S(t) = \sum_{i=0}^{n-1} P_i B_{i,k}(t)$ şi să presupunem că $t_j \leq t < t_{j+1}$.
**for** $i := j-k$ **to** $j$ **do**
  $P_i^0(t) := P_i$;
**end for**
**for** $r := 0$ **to** $k-1$ **do**
  **for** $i := j-k+r+1$ **to** $j$ **do**
    $P_i^{r+1} := \omega_{i,k-r}(t) P_i^r + (1 - \omega_{i,k-r}(t) P_{i-1}^r$
    $= \dfrac{(t-t_i)P_i^r(t) + (t_{i+k-r} - t)P_{i-1}^r}{t_{i+k-r} - t_i}$;
  **end for**
**end for**
$S_j(t) := P_k^k(t)$;

În cazul particular al unei curbe Bézier se obţine un algoritm mult mai simplu, care nu depinde de $j$ (vezi algoritmul 4.2). El este datorat lui Paul de Faget de Casteljau.

Dăm şi două posibilităţi de alegere a punctelor diviziunii.

**Algoritmul 4.2** Algoritmul de Casteljau pentru evaluarea unei curbe Bezier

**Intrare:** Poligonul de contrul $P_i$, $i = \overline{0,k}$ şi punctul t în care se face evaluarea
**Ieşire:** Valoarea $B(t)$

1: Fie $B(t) = \sum_{i=0}^{k} P_i p_{i,k}(t)$.
2: **for** $i := 0$ **to** $k$ **do**
3: $\quad P_i^0(t) := P_i$;
4: **end for**
5: **for** $r := 0$ **to** $k-1$ **do**
6: $\quad$ **for** $i := r+1$ **to** $k$ **do**
7: $\quad\quad P_i^{r+1} := (1-t)P_{i-1}^r(t) + tP_i^r(t)$;
8: $\quad$ **end for**
9: **end for**
10: $B(t) := P_k^k(t)$;

**Varianta 1:** $t_j = j$;

**Varianta 2:**

$$t_j = \begin{cases} 0, & 0 \le j \le k; \\ j-k, & k+1 \le j \le n; \\ n-k, & j > n. \end{cases}$$

## 4.2.4. Exemple

Să desenăm curba cuadratică corespunzând nodurilor $t_0 = t_1 = t_2 = 1$, $t_3 = 2$, $t_4 = 3$, $t_5 = 4$ şi $t_6 = t_7 = t_8 = 5$ şi al cărei poligon de control este dat în figura 4.3. Graficul curbei apare de asemenea în figura 4.3.

Dacă se alege o repartiţie uniformă a nodurilor şi dorim o curbă de clasă $C^1$, putem alege $t_i = i$, $i \in \mathbb{Z}$. Avem $B_i(t) = B_{i+r}(t+r)$, $\forall r \in \mathbb{Z}$ (figura 4.4), adică elementele bazei B-spline se deduc una din alta print-o translaţie întreagă. Să considerăm $S(t) = \sum_{i=-\infty}^{\infty} P_i B_{i,r}(t)$ şi să punem $P_{6+i} = P_i$; avem $S(t-6) = S(t)$, iar curba este definită pe un interval arbitrar de lungime 6. Se obţine în acest mod o curbă închisă (figura 4.5).

În figura 4.6 se dau două curbe Bézier de grad 3 pentru două forme diferite ale poligonului de control. Urmează acum un exemplu mai detaliat de aplicare a algoritmului Cox-deBoor. Fie nodurile $a = 0 = t_0 = t_1 = t_2 = t_3$, $t_4 = 1$, $t_5 = 2$, $b = 3 = t_6 = t_7 = t_8 = t_9$. Gradul curbei va fi $k = 3$. Poligonul de control are 6 puncte şi este dat în figura 4.7. Avem $S(t) = \sum_{i=0}^{5} P_i B_{i,3}(t)$; să calculăm $S(\tau)$ pentru

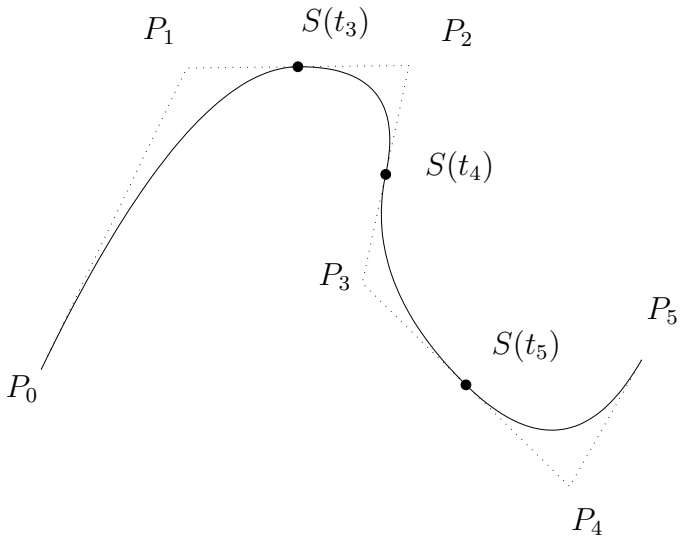



Figura 4.3: Curba B-spline cuadratică şi poligonul ei de control

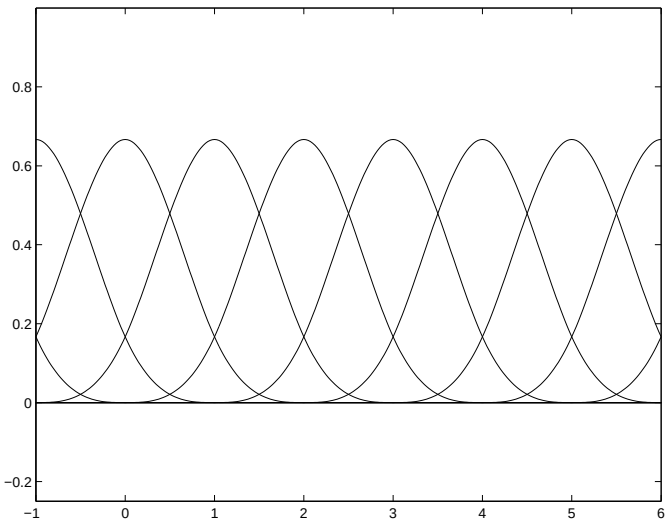



Figura 4.4: Bază pentru noduri din $\mathbb{Z}$

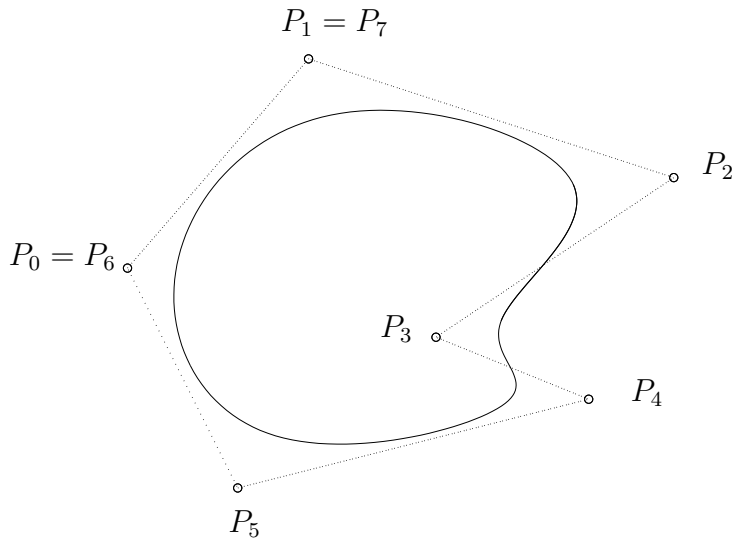



Figura 4.5: Curbă B-spline închisă, periodică

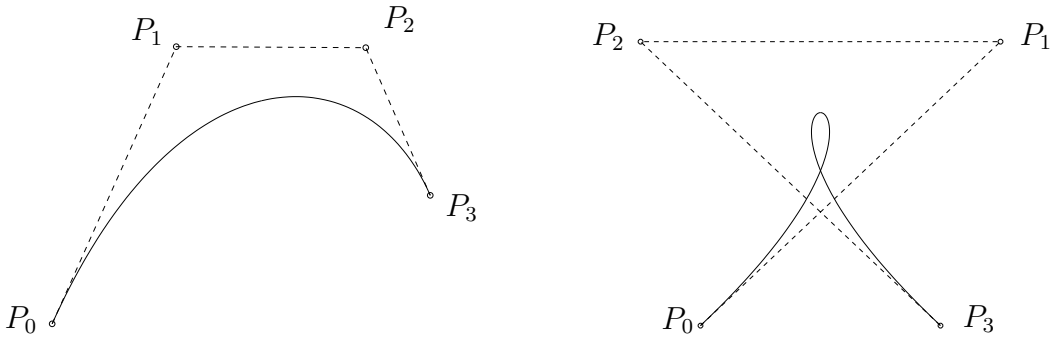



Figura 4.6: Două curbe Bézier de grad 3

$\tau = \frac{3}{2}$. Deoarece $t_4 < \tau < t_5$, calculele se organizează tabelar sub forma

$$\begin{array}{ccccccc} P_1 & & P_2 & & P_3 & & P_4 \\ & P_2^1 & & P_3^1 & & P_4^1 & \\ & & P_3^2 & & P_4^2 & & \\ & & & P_4^3 & & & \end{array}$$

unde $P_4^3 = S\left(\frac{3}{2}\right)$, iar

$$P_2^1 = \frac{3}{4}P_2 + \frac{1}{4}P_1 \qquad P_3^1 = \frac{1}{2}P_1 + \frac{1}{2}P_2 \qquad P_4^1 = \frac{1}{4}P_4 + \frac{3}{4}P_3$$
$$P_3^2 = \frac{3}{4}P_3^1 + \frac{1}{4}P_2^1 \qquad P_4^2 = \frac{1}{4}P_4^1 + \frac{3}{4}P_3^1$$
$$P_4^3 = \frac{1}{2}P_4^2 + \frac{1}{2}P_3^2$$

Graficul curbei apare în figura 4.7. Se observă că S(t) este tangentă în punctul $P_j^k(t)$ la segmentul $P_k^{k-1}P_j^{k-1}$. Să ilustrăm acum caracterul local al curbelor B-spline. În figura

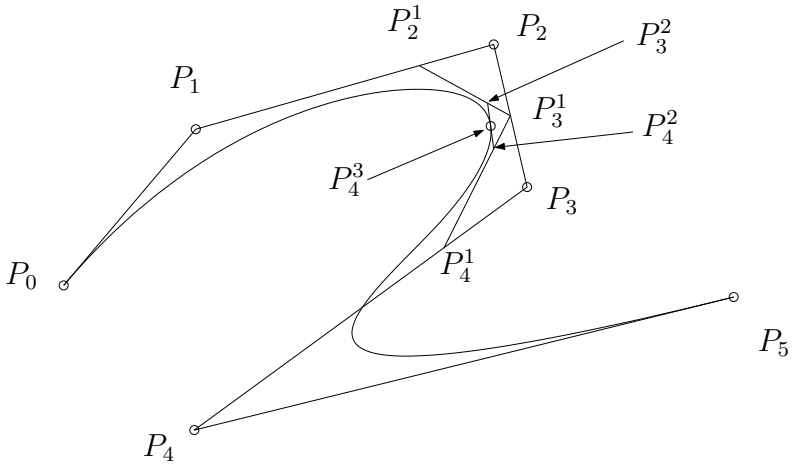



Figura 4.7: Un exemplu de aplicare a algoritmului Cox-deBoor

4.8 este reprezentată o curbă B-spline de gradul 3 corespunzătoare unui poligon de control cu 9 puncte (linie continuă). Se modifică coordonatele punctului $P_3$, iar poligonul de control modificat şi B-spline-ul corespunzător apar cu linie întreruptă.

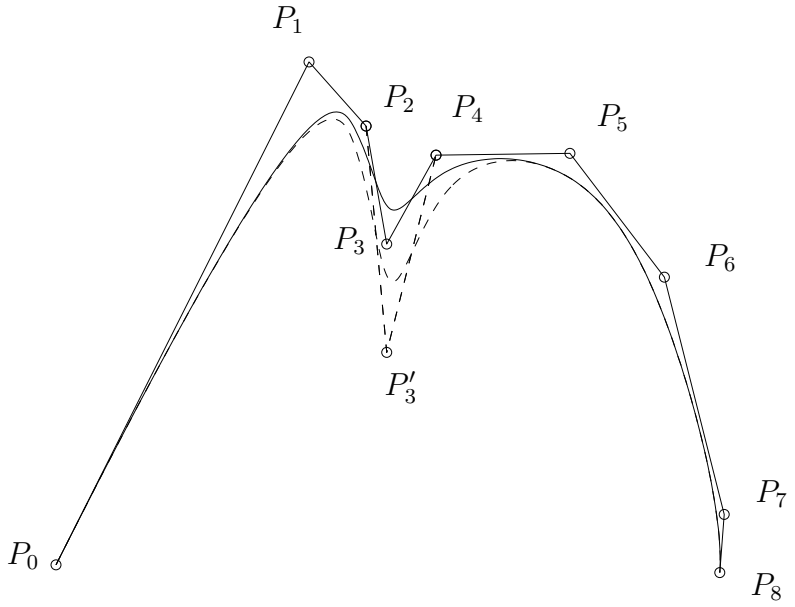



Figura 4.8: Caracterul local al curbelor B-spline. Se ilustrează efectul modificării coordonatelor unui punct ($P_3$)

## 4.3. Funcţii spline cu variaţie diminuată

Funcţiile spline cu variaţie diminuată au fost introduse în 1964 de Isaac J. Schoenberg [3] ca o generalizare a polinoamelor Bernstein.

Fie diviziunea

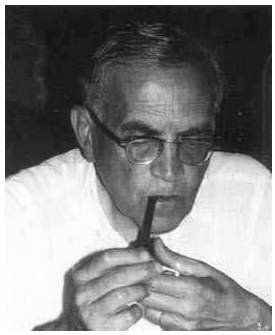
$$\Delta:\ t_0 \leq t_1 \cdots \leq t_k \leq a \leq \cdots \leq b \leq t_n \leq t_{n+1} \leq t_{n+k}.$$

Vom presupune că multiplicitatea oricărui nod nu depăşeşte $k+1$.

**Definiţia 4.3.1** *Fie $f : [t_0, t_{n+k}] \mapsto \mathbb{R}$ şi diviziunea $\Delta = (t_i)_{i=\overline{0,n+k}}$ ca mai sus. Punem*

$$\xi_i = \frac{t_{i+1} + \cdots + t_i + k}{k}, \qquad i = \overline{0, n-1} \tag{4.3.1}$$

[3] Isaac J. Schoenberg (1993-1990) - matematician născut la Galaţi. A studiat la universităţile din Iaşi, Berlin şi G otingen. În 1926 îsi susţine doctoratul la Universitatea din Iaşi. Din 1930 a activat în Statele Unite (1941-1966 University of Pennsylvania, 1966-1973 University of Wisconsin). Contribuţii în domeniul Teoriei aproximării. Rezultatele sale în domeniul funcţiilor spline l-au făcut celebru (de fapt el a introdus termenul de funcţie spline). A fost şi un om sensibil, de aleasă cultură.



*şi definim operatorul* $S_\Delta$

$$(S_\Delta f)(x) = \sum_{i=0}^{n-1} f(\xi_i) B_{i,k}(x). \tag{4.3.2}$$

*Acest operator se numeşte* operator spline cu variaţie diminuată *sau* operatorul lui Schoenberg*, iar* $S_\Delta f$ *se numeşte funcţie spline cu variaţie diminuată.*

$S_\Delta f$ fiind o combinaţie liniară de funcţii B-spline este un spline de gradul $k$. Pentru punctele $\xi$ avem

$$a = \xi_1 < \cdots < \xi_{n-1} = b.$$

**Propoziţia 4.3.2** $S_\Delta$ *este un operator liniar şi poziv şi* $\forall f \in \mathbb{P}_1$ *avem* $S_\Delta f = f$.

*Demonstraţie.* Liniaritatea şi pozitivitatea rezultă din definiţie folosind proprietăţile B-splinelor. Trebuie să arătăm că $S_\Delta$ reproduce pe 1 şi pe $x$. $S_\Delta 1 = 1$ rezultă din partiţia unităţii. Din identitatea lui Marsden (4.2.4) egalând coeficienţii lui $t^{k-1}$ obţinem

$$\sum_{i=0}^{n-1} \xi_i B_{i,k}(x) = x,$$

de unde $S_\Delta x = x$. □

**Propoziţia 4.3.3** $S_\Delta f$ *are următoarea proprietate*

$$V(S_\Delta f - \ell) \le V(f - \ell), \text{ pe } [a,b] \forall \ell \in \mathbb{P}_1$$

*unde* $V(g)$ *este numărul variaţiilor de semn ale lui* $g$*.*

Numele de spline cu variaţie diminuată vine tocmai de la această proprietate.

**Teorema 4.3.4** *Are loc inegalitatea*

$$\|g - S_\Delta g\|_\infty \le \frac{k^2}{2} \|\Delta\|^2 \|g''\|_\infty. \tag{4.3.3}$$

*Demonstraţie.* Fie $x_0 \in (a,b)$; trebuie estimată cantitatea $|g(x_0) - S_\Delta f(x_0)|$. Presupunem că $x_0 \in [t_i, t_{i+1})$, ceea ce implcă $\xi_{i-k} \le x_0 < \xi_i$ şi atrage $B_{j,k}(x_0) = 0$ pentru $j \notin [i-k, i]$. Fie $P(x) = g(x_0) + g'(x_0)(x - x0)$ ecuaţia tangentei la $g$ în $x_0$. Avem $S_\Delta P = P$ şi deci

$$S_\Delta(g - P) = S_\Delta g - S_\Delta P = S_\Delta g - P = (S_\Delta g - g) + g - P$$

şi pentru $x = x_0$

$$g(x_0) - S_\Delta g(x_0) = -S_\Delta(g-P)(x_0) = -\sum_{j=i-k}^{i}(g-P)(\xi_j)B_{j,k}(x_0)$$

de unde

$$|g(x_0) - S_\Delta g(x_0)| \leq \sup\{|(g-P)(x) : x \in (\xi_{i-k}, \xi_i)\}.$$

Dar din formula lui Taylor $(g-P)(x) = \frac{(x-x_0)^2}{2}g''(\xi)$ cu $\xi \in (x_0, x)$ şi $|x-x_0| \leq k\|\Delta\|$ pentru $\xi \in (\xi_{i-k}, \xi_i)$, de unde

$$|g(x_0) - S_\Delta g(x_0)| \leq \frac{k^2}{2}\|\Delta\|^2\|g''\|_\infty.$$

□

**Corolarul 4.3.5**

$$\|x^2 - S_\Delta e_2\|_\infty \leq k^2\|\Delta\|^2.$$

**Definiţia 4.3.6** *Fie* $f : [a,b] \to \mathbb{R}$. *Formula*

$$F = S_\Delta f + R_\Delta f \tag{4.3.4}$$

*se numeşte* formula de aproximare spline cu variaţie diminuată, *iar* $R_\Delta f$ *este* termenul rest.

**Teorema 4.3.7** *Dacă* $f \in C^2[a,b]$ *atunci*

$$(R_\Delta f)(x) = \int_a^b \varphi(x;t)f''(t)dt \tag{4.3.5}$$

*unde*

$$\varphi(x;t) = (x-t)_+ - \sum_{i=0}^{n-1} B_{i,k}(x)(\xi_i - t)_+$$

*respectiv*

$$(R_\Delta f)(x) = -\frac{1}{2}[S_\Delta e_2(x) - x^2]f''(\xi).$$

*Demonstraţie.* Se aplică teorema lui Peano şi se ţine cont că $\varphi(x;t)$ are semn constant pentru $t, x \in [a,b]$. □

**Observaţia 4.3.8.** În cazul particular când $t_0 = \cdots = t_k = a < b = t_{k+1} \cdots = t_{2k+1}$ se obţine operatorul Bernstein ($S_\Delta f = B_k f$). ♢

# 4.4. Operatori liniari şi pozitivi

**Definiţia 4.4.1** *Operatorul $L : X \to Y$ se numeşte* liniar şi pozitiv *dacă pentru orice $f, g \in X$ şi orice scalari $\alpha, \beta$*

$$L(\alpha f + \beta g) = \alpha Lf + \beta Lg$$
$$f \geq 0 \Rightarrow Lf \geq 0.$$

**Observaţia 4.4.2.** $X$ şi $Y$ sunt spaţii liniare ordonate. De obicei $X = \{f : E_1 \subset \mathbb{R} \to \mathbb{R}\}$, $Y = \{g = Lf : E_2 \subset \mathbb{R} \to \mathbb{R}\}$. ♢

**Teorema 4.4.3** *Dacă $L : X \to Y$ este un operator liniar şi pozitiv şi $f, g \in X$, atunci*

*(i)* $f \leq g \Rightarrow Lf \leq Lg$ *(monotonie);*

*(ii)* $|Lf| \leq L|f|$.

**Observaţia 4.4.4.** Pentru a pune în evidenţă faptul că un operator liniar L se aplică unei funcţii $f$, ca funcţie de variabila independentă $t$, iar valoarea funcţiei obţinute se consideră pe punctul $x$ vom scrie $L(f(t); x)$ în loc de $(Lf)(x)$. ♢

Teorema care urmează, numită prima teoremă a lui Korovkin sau teorema Bohman-Popoviciu [4]-Korovkin, este un criteriu de convergenţă uniformă pentru şirurile construite cu ajutorul operatorilor liniari şi pozitivi.

**Teorema 4.4.5** *Fie $(L_m)$, $L_m : C[a, b] \to C[a, b]$, un şir de operatori liniari şi pozitivi. Dacă*

$$\begin{aligned} (L_m e_0)(x) &= 1 + u_m(x); \\ (L_m e_1)(x) &= x + v_m(x); \\ (L_m e_2)(x) &= x^2 + w_m(x) \end{aligned} \tag{4.4.1}$$

[4] Tiberiu Popoviciu (1906-1975) mare matematician român, membru al Academiei române, fondatorul şcolii de Analiză numerică şi Teoria aproximării din Cluj-Napoca. Doctorat în Franţa, la École Normale Supérieure (1933). Contribuţii importante Si de pionierat în domeniul Analizei matematice, Teoriei funcţiilor, Teoriei aproximării şi Analizei numerice. A contribuit de asemenea la începuturile şi dezvoltarea informaticii româneşti şi clujene.

*şi* $\lim_{m\to\infty} u_m(x) = 0$, $\lim_{m\to\infty} v_m(x) = 0$ *şi* $\lim_{m\to\infty} w_m(x) = 0$, *uniform pe* $[a,b]$, *atunci pentru orice* $f \in C[a,b]$

$$\lim_{m\to\infty} (L_m f)(x) = f(x)$$

*uniform pe* $[a,b]$.

*Demonstraţie.* Fie $M = \max_{a\le x\le b} |f(x)|$ şi $t$ un punct oarecare, dar fix, din intervalul $[a,b]$. Funcţia $f$ fiind uniform continuă pe $[a,b]$, $\forall \varepsilon > 0$, $\exists \delta > 0$, astfel încât pentru $x \in [a,b]$ şi $|t-x| < \delta$ să avem $|f(t)-f(x)| < \frac{\varepsilon}{2}$. Fie $J = \{x \in [a,b] : |t-x| \ge \delta\}$. Pentru $x \in J$ avem

$$|f(t)-f(x)| \le 2M \le \frac{2M}{\delta^2}(t-x)^2$$

prin urmare

$$|f(t)-f(x)| \le \frac{\varepsilon}{2} + \frac{2M}{\delta^2}(t-x)^2, \qquad x,t \in [a,b]. \tag{4.4.2}$$

Cu notaţia din observaţia 4.4.4 ($L_m(f(t);x)$ în loc de $(L_m f)(x)$) pentru $t = x$ avem $L_m(f(x);x) = f(x)\cdot L_m(1;x) = f(x)(L_m e_0)(x)$. Din enunţ (egalităţile (4.4.1)) se observă că putem scrie

$$\begin{aligned}(L_m f)(x) - f(x) &= L_m(f(t);x) - f(x)L_m(1;x) + f(x)u_m(x)\\ &= L_m(f(t)-f(x);x) + f(x)u_m(x).\end{aligned}$$

astfel că avem

$$|(L_m f)(x) - f(x)| \le |L_m(f(t)-f(x);x)| + |f(x)||u_m(x)|. \tag{4.4.3}$$

Folosind liniaritatea, teorema 4.4.3 şi inegalitatea (4.4.2) deducem că

$$\begin{aligned}|L_m(f(t)-f(x);x)| &\le L_m(|f(t)-f(x)|;x) < L_m\left(\frac{\varepsilon}{2} + \frac{2M}{\delta^2}(t-x)^2;x\right)\\ &= \frac{\varepsilon}{2}L_m(1;x) + \frac{2M}{\delta^2}L_m((t-x)^2;x).\end{aligned}$$

Ultima inegalitate şi (4.4.3) ne dau

$$\begin{aligned}|(L_m f)(x) - f(x)| &< \frac{\varepsilon}{2}L_m(1;x) + \frac{2M}{\delta^2}L_m((t-x)^2;x) + |f(x)||u_m(x)|\\ &< \frac{\varepsilon}{2} + \left(|f(x) + \frac{\varepsilon}{2}\right)|u_m(x)| + \frac{2M}{\delta^2}L_m((t-x)^2;x).\end{aligned} \tag{4.4.4}$$

Din enunţ mai rezultă că

$$L_m((t-x)^2;x) = x^2 u_m(x) - 2x v_m(x) + w_m(x). \tag{4.4.5}$$

Deoarece $u_m \rightrightarrows 0$, $v_m \rightrightarrows 0$, $w_m \rightrightarrows 0$,

$$\left(|f(x)+\frac{\varepsilon}{2}\right)|u_m(x)|+\frac{2M}{\delta^2}L_m((t-x)^2;x)<\frac{\varepsilon}{2},\ \text{pentru } m>N_\varepsilon$$

din (4.4.4) se obţine

$$|(L_mf)(x)-f(x)|<\frac{\varepsilon}{2}.$$

□

**Corolarul 4.4.6** *Dacă* $(L_m)$, $L_m : C[a,b] \to C[a,b]$, $m \in \mathbb{N}^*$ *este un şir de operatori liniari şi pozitivi şi*

$$\lim_{m\to\infty}(L_m)(1;x)=1, \qquad \lim_{m\to\infty}(L_m)((t-x)^2;x)=0 \tag{4.4.6}$$

*pentru orice* $f \in C[a,b]$ *şirul* $(L_mf)$ *converge uniform la* $f$ *pe* $[a,b]$.

Concluzia rezultă (4.4.4) şi (4.4.5).

**Observaţia 4.4.7.** Funcţiile $e_0$, $e_1$, $e_2$ din teorema 4.4.5 se numesc funcţii de probă. Numărul de funcţii de probă nu poate fi modificat. ♢

Un rezultat asemănător cu cel din teorema 4.4.5 are loc şi pentru funcţiile $2\pi$-periodice cu funcţiile de probă $1$, $\cos x$, $\sin x$, iar corolarul corespunzător se aplică pentru $1$ şi $\sin^2\frac{t-x}{2}$. Fie $C_{2\pi}$ spaţiul funcţiilor continue $2\pi$-periodice.

**Teorema 4.4.8 (teorema a doua a lui Korovkin)** *Fie* $(L_m)$ *un şir de operatori liniari pozitivi* $L_m : C_{2\pi} \to C_{2\pi}$. *Dacă* $\forall x \in \mathbb{R}$

$$\begin{aligned}&L_m(1;x)=1+u_m(x);\\ &L_m(\sin t;x)=\sin x+v_m(x);\\ &L_m(\cos t;x)=\cos x+w_m(x)\end{aligned}$$

*şi*

$$\lim_{m\to\infty}u_m(x)=\lim_{m\to\infty}u_m(x)=\lim_{m\to\infty}u_m(x)=0$$

*uniform pe* $\mathbb{R}_+$, *atunci* $\forall f \in C_{2\pi}$ $L_mf \rightrightarrows f$ *pe* $\mathbb{R}$.

*Demonstraţie.* Este analoagă cu a teoremei 4.4.5, analoaga inegalităţii (4.4.2) fiind

$$|f(t)-f(x)|<\frac{\varepsilon}{2}+2M\sin^{-2}\frac{\delta}{2}\sin^2\frac{t-x}{2}.$$

□

## Exemple

**Exemplul 4.4.9 (Operatorul lui Bernstein).** Fie $(B_m)$ şirul operatorilor Bernstein. Avem

$$B_m(1;x)=1,\ B_m(t,x)=x,\ B_m(t^2;x)=x^2+\frac{x(1-x)}{m},\ x\in[0,1].$$

Deci $u_m(x)=0$, $v_m(x)=0$, $w_m(x)=\frac{x(1-x)}{m}$. Deoarece $\lim_{m\to\infty}\frac{x(1-x)}{m}=0$ uniform pe $[a,b]$, $B_mf\rightrightarrows f$ pe $[0,1]$, rezultat echivalent cu teorema 4.1.2. ♢

**Exemplul 4.4.10 (Operatorul lui Schoenberg).** Operatorul spline cu variaţie diminuată $S_\Delta$ este liniar şi pozitiv. De asemenea

$$\begin{aligned}&S_\Delta(1;x)=1;\\&S_\Delta(t;x)=x;\\&S_\Delta(t^2;x)=x^2+E(x).\end{aligned}$$

Deoarece $E(x)\rightrightarrows 0$ când $\|\Delta\|\to 0$, $S_\Delta f\rightrightarrows f$. ♢

**Exemplul 4.4.11 (Operatorul Hermite-Fejér).** Pornind de la operatorul de interpolare Hermite cu noduri duble rădăcini ale polinomului Cebâşev de speţa I, $T_{m+1}$

$$x_k=\cos\frac{2k+1}{2(m+1)}\pi,\qquad k=\overline{0,m},$$

$$(H_{2m+1}f)(x)=\sum_{k=0}^{m}h_{k0}(x)f(x_k)+\sum_{k=0}^{m}h_{k1}(x)f(x_k)$$

şi omiţând a doua sumă, Fejér [5] a obţinut operatorul

$$(F_{2m+1}f)(x)=\sum_{k=0}^{m}h_k(x)f(x_k),$$

unde

$$h_k(x)=h_{k0}(x)=(1-x_kx)\left[\frac{T_{m+1}(x)}{(m+1)(x-x_k)}\right]^2.$$

[5] Leopold Fejér (1880-1959) Matematian maghiar, lider al generaţiei sale. A avut contribuţii importante în domeniul aproximării şi interpolării în real şi complex şi teoriei seriilor Fourier. Rezultatul său privind convergenţa seriilor Fourier a fost obţinut când era încă student. A funcţionat ca profesor şi la universitatea din Cluj.

**Propoziţia 4.4.12** *Operatorul lui Fejér are următoatele proprietăţi:*

1. $(F_{2m+1}f)(x_k) = f(x_k), \qquad (F_{2m+1}f)'(x_k) = 0 \qquad \sum_{k=0}^{m} h_k(x) = 1.$
2. $F_{2m+1}$ *este un operator liniar şi pozitiv.*
3. *Dacă* $f \in C[-1,1]$, *atunci* $F_{2m+1}f \rightrightarrows f$ *pe* $[-1,1]$, *când* $m \to \infty$.

*Demonstraţie.* 1. Rezultă din proprietăţile polinoamelor de interpolare Hermite.

2. Liniaritatea rezultă din definiţie, iar pozitivitatea din

$$1 - x_k x \geq 1 - |x_k| > 0, \ x \in [-1,1].$$

3. Din proprietatea 1 rezultă că $F_{2m+1}(1;x) = 1$, pentru $x \in [-1,1]$.

$$F_{2m+1}((t-x)^2;x) = \sum_{k=0}^{m}(1-x_k x)\left[\frac{T_{m+1}(x)}{(m+1)(x-x_k)}\right]^2 (x_k - x)^2$$
$$= \frac{1}{(m+1)^2}T_{m+1}^2(x)\sum_{k=0}^{m}(1-x_k x).$$

Datorită simetriei nodurilor $x_k$ faţă de origine, $\sum_{k=0}^{m} x_k = 0$ şi se obţine

$$F_{2m+1}((t-x)^2;x) = \frac{1}{m+1}T_{m+1}^2(x) \leq \frac{1}{m+1};$$

adică

$$\lim_{m\to\infty} F_{2m+1}((t-x)^2;x) = 0, \text{ uniform pe } [-1,1].$$

Se aplică apoi corolarul 4.4.6 la teorema 4.4.5.

□

Fiind un operator polinomial, operatorul lui Fejér poate da o demonstraţie constructivă a teoremei lui Weierstrass. ♢

# 4.5. Cea mai bună aproximare uniformă

Fie $f \in C[a,b]$ şi presupunem că un polinom de grad cel mult $n$ minimizează norma $\|f - p_n\|_\infty$. Un astfel de polinom se numeşte *polinomul de cea mai bună aproximare uniformă* a lui $f$.

În teoria aproximării uniforme rolul central este jucat de *punctele de alternanţă Cebâşev* pentru funcţie $R(x) = f(x) - p_n(x)$. Punctele de alternanţă de grad $m$ sunt nodurile unei grile (diviziuni)

$$a \leq x_1 \leq \cdots \leq x_m \leq b$$

având următoarele proprietăţi:

(1) $|R(x_i)| = \max_{a\le x\le b}|R(x)|, \qquad i = \overline{1,m}.$

(2) $R(x_i)R(x_{i+1}) < 0, \qquad = i = \overline{1,m-1}.$

Vom nota mulţimea tuturor diviziunilor de acest tip pe $[a,b]$ prin $\mathcal{A}(m,a,b,R)$.

**Teorema 4.5.1 (Cebîşev)** *Pentru ca un polinom $p_n$ de grad cel mult $n$ să fie cea mai bună aproximantă uniformă a lui $f \in C[a,b]$, este necesar şi suficient ca $\mathcal{A}(n+2,a,b,R)$ să fie nevidă.*

*Demonstraţia suficienţei.* Presupunem că există un polinom $q_n \in \mathbb{P}_n$ astfel încât

$$\|f-q_n\|_\infty \le \|f-p_n\|_\infty.$$

Atunci în punctele de alternanţă Cebîşev

$$|f(x_i)-q_n(x_i)| < |f(x_i)-p_n(x_i)|$$

ceea ce implică faptul că funcţia $g(x) \equiv (f(x)-p_n(x))-(f(x)-q_n(x))$ are în punctele $x_i$ acelaşi semn ca $R(x) = f(x)-p_n(x)$. Deoarece semnele lui $R(x_i)$ alternează, există un zero în interiorul fiecărui subinterval $[x_i,x_{i+1}]$. Deci, $g$ are $n+1$ zerori pe $[a,b]$. Aceasta nu se poate întâmpla dacă $g$ nu este identic nul. □

**Exemplul 4.5.2.** Vom determina polinomul de cea mai bună aproximare uniformă de grad întâi pentru funcţia $f(x) = \sqrt{x}$ pe $[a,b] \subset \mathbb{R}_+$.

Polinomul căutat are forma $P_1^* = c_0 + c_1x$. Eroarea de aproximare este $e_1(x) = c_0 + c_1x - \sqrt{x}$. Derivata ei, $e_1'(x) = c_1 - \frac{1}{2\sqrt{x}}$ se anulează în $x_1 = \frac{1}{4c_1^2}$. Conform teoremei lui Cebîşev abaterea maximă se realizează în trei puncte din $[a,b]$ şi obţinem sistemul neliniar

$$\begin{cases} c_0 + c_1a - \sqrt{a} = E_1 \\ c_0 + \frac{1}{4c_1} - \frac{2}{c_1} = -E_1 \\ c_0 + c_1b - \sqrt{3} = E_1 \end{cases},$$

cu soluţiile

$$\begin{aligned} c_0 &= \frac{1}{2}\left(\sqrt{a} - \frac{a}{\sqrt{a}+\sqrt{b}} + \frac{\sqrt{a}+\sqrt{b}}{4}\right), \\ c_1 &= \frac{1}{\sqrt{a}+\sqrt{b}}, \\ E_1 &= c_0 + c_1a - \sqrt{a}. \end{aligned}$$

♢

CAPITOLUL 5

# Aproximarea funcţionalelor liniare

## Cuprins



## 5.1. Introducere

Fie $X$ un spaţiu liniar, $L_1, \ldots, L_m$ funcţionale liniare reale, liniar independente, definite pe $X$ şi $L : X \to \mathbb{R}$ o funcţională liniară reală astfel încât $L, L_1, \ldots, L_m$ să fie liniar independente.

**Definiţia 5.1.1** *O formulă de aproximare a funcţionalei $L$ în raport cu funcţionalele $L_1, \ldots, L_m$ este o formulă de forma*

$$L(f) = \sum_{i=1}^{m} A_i L_i(f) + R(f), \quad f \in X. \tag{5.1.1}$$

*Parametrii reali $A_i$ se numesc coeficienţii formulei, iar $R(f)$ termenul rest.*

Pentru o formulă de aproximare de forma (5.1.1), dându-se funcţionalele $L_i$, se pune problema determinării coeficienţilor $A_i$ şi a studiului termenului rest corespunzător valorilor obţinute pentru coeficienţi.

**Observaţia 5.1.2.** Forma funcţionalelor $L_i$ depinde de informaţiile deţinute asupra lui $f$ (ele exprimând de fapt aceste informaţii), dar şi de natura problemei de aproximare, adică de forma lui $L$. ♢

**Exemplul 5.1.3.** (E1) Dacă $X = \{f |\ f : [a,b] \to \mathbb{R}\}$, $L_i(f) = f(x_i)$, $i = \overline{0,m}$, $x_i \in [a,b]$ şi $L(f) = f(\alpha)$, $\alpha \in [a,b]$, formula de interpolare Lagrange

$$f(\alpha) = \sum_{i=0}^{m} \ell_i(\alpha) f(x_i) + (Rf)\alpha$$

este o formulă de tip (5.1.1), cu coeficienţii $A_i = \ell_i(\alpha)$, 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 5.1.4.** Dacă $X$ şi $L_i$ sunt ca în exemplul 5.1.3 şi există $f^{(k)}(\alpha)$, $\alpha \in [a,b]$, $k \in \mathbb{N}^*$, iar $L(f) = f^{(k)}(\alpha)$ se obţine 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 şi *formulă de derivare numerică* . ♢

**Exemplul 5.1.5.** Dacă $X$ este un spaţiu de funcţii definite pe $[a,b]$, integrabile pe $[a,b]$ şi pentru care există $f^{(j)}(x_k)$, $k = \overline{0,m}$, $j \in I_k$, cu $x \in [a,b]$ şi $I_k$ mulţimi 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 obţine 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*. ♢

**Definiţia 5.1.6** *Dacă $\mathbb{P}_r \subset X$, numărul $r \in \mathbb{N}$ cu proprietatea că $Ker(R) = \mathbb{P}_r$ se numeşte* grad de exactitate *al formulei de aproximare (5.1.1).*

**Observaţia 5.1.7.** Deoarece $R$ este o funcţională liniară proprietatea $Ker(R) = \mathbb{P}_r$ este echivalentă cu $R(e_k) = 0$, $k = \overline{0, r}$ şi $R(e_{r+1}) \neq 0$, unde $e_k(x) = x^k$. ♢

Putem acum să formulăm *problema generală de aproximare:* dându-se o funcţională liniară $L$ pe $X$, $m$ funcţionale liniare $L_1, L_2, \ldots, L_m$ pe $X$ şi valorile lor (”datele”) $\ell_i = L_i f$, $i = \overline{1, m}$ aplicate unei anumite funcţii $f$ şi un subspaţiu liniar $\Phi \subset X$ cu $\dim \Phi = m$, dorim să găsim o formulă de aproximare de tipul

$$Lf \approx \sum_{i=1}^{m} a_i L_i f \tag{5.1.2}$$

care să fie exactă (adică să aibă loc egalitate), pentru orice $f \in \Phi$.

Este natural (deoarece dorim să interpolăm) să facem următoarea

**Ipoteză:** „problema de interpolare“

Să se găsească $\varphi \in \Phi$ astfel încât

$$L_i \varphi = s_i, \quad i = \overline{1, m} \tag{5.1.3}$$

are o soluţie unică $\varphi(\cdot) = \varphi(s, \cdot)$, pentru $s = [s_1, \ldots, s_m]^T$, arbitrar.

Putem exprima ipoteza noastră mai explicit în termenii unei baze date $\varphi_1, \varphi_2, \ldots, \varphi_m$ a lui $\Phi$ şi a matricei Gram [1] asociată

$$G = [L_i\varphi_j] = \begin{vmatrix} L_1\varphi_1 & L_1\varphi_2 & \ldots & L_1\varphi_m \\ L_2\varphi_1 & L_2\varphi_2 & \ldots & L_2\varphi_m \\ \ldots & \ldots & \ldots & \ldots \\ L_m\varphi_1 & L_m\varphi_2 & \ldots & L_m\varphi_m \end{vmatrix} \in \mathbb{R}^{m\times m}. \tag{5.1.4}$$

Cerem ca

$$\det G \neq 0. \tag{5.1.5}$$

Este uşor de văzut că această condiţie este independentă de alegerea particulară a bazei. Pentru a arăta că solvabilitatea unică a lui (5.1.3) şi condiţia (5.1.5) sunt echivalente, exprimăm $\varphi$ din (5.1.3) ca o combinaţie liniară a funcţiilor de bază

$$\varphi = \sum_{j=1}^{nm} c_j\varphi_j \tag{5.1.6}$$

şi observăm că condiţiile de interpolare

$$L_i\left(\sum_{j=1}^{m} c_j\varphi_j\right) = s_i, \quad i = \overline{1,m}$$

pot fi scrise (ţinând cont de liniaritatea lui $L_i$) sub forma

$$\sum_{j=1}^{m} c_j L_i\varphi_j = s_i, \quad i = \overline{1,m},$$

adică

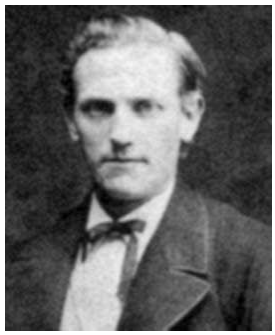
$$Gc = s, \quad c = [c_1, c_2, \ldots, c_m]^T, \quad s = [s_1, s_2, \ldots, s_m]^T. \tag{5.1.7}$$

Aceasta are o soluţie unică pentru $s$ arbitrar dacă şi numai dacă are loc (5.1.5).

Avem două abordări pentru rezolvarea acestei probleme.

---

[1] Jórgen Pedersen Gram (1850-1916), matematician danez, a studiat la Universitatea din Copenhaga. După absolvire a intrat la o companie de asigurări ca asistent calculator şi apoi a promovat treptat până a ajuns director. A avut contribuţii în domeniul dezvoltării în serie a funcţiilor şi aproximării Cebîşev şi în sensul celor mai mici pătrate. „Determinantul Gram" a fost introdus de el în legătură cu studiile sale asupra liniar independenţei.



**Metoda interpolării.** Rezolvăm problema generală de aproximare prin interpolare

$$Lf \approx L\varphi(\ell;\cdot), \quad \ell = [\ell_1, \ell_2, \ldots, \ell_m]^T, \quad \ell_i = L_i f \tag{5.1.8}$$

Cu alte cuvinte aplicăm $L$ nu lui $f$, ci soluţiei $\varphi(\ell;\cdot)$ a problemei de aproximare (5.1.3) în care $s = \ell$. Ipoteza noastră ne garantează că $\varphi(\ell;\cdot)$ este unic determinat. În particular, dacă $f \in \Phi$, atunci (5.1.8) are loc cu egalitate, deoarece $\varphi(\ell;\cdot) = f(\cdot)$, în mod trivial. Astfel, aproximanta noastră (5.1.8) satisface condiţiile de exactitate cerute pentru (5.1.2). Rămâne doar să arătăm că (5.1.8) produce o aproximare de forma (5.1.2). Pentru aceasta să observăm că interpolantul în (5.1.8) este

$$\varphi(\ell;\cdot) = \sum_{j=1}^{m} c_j \varphi_j(\cdot)$$

unde vectorul $c = [c_1, c_2, \ldots, c_m]^T$ satisface (5.1.7) cu $s = \ell$

$$Gc = \ell, \quad \ell = [L_1 f, L_2 f, \ldots, L_m f]^T.$$

Scriind

$$\lambda_j = L\varphi_j, \quad j = \overline{1,m}, \quad \lambda = [\lambda_1, \lambda_2, \ldots, \lambda_m]^T, \tag{5.1.9}$$

avem din liniaritatea lui $L$

$$L\varphi(\ell;\cdot) = \sum_{j=1}^{m} c_j L\varphi_j = \lambda^T c = \lambda^T G^{-1}\ell = [(G^T)^{-1}\lambda]^T \ell,$$

adică

$$L\varphi(\ell;\cdot) = \sum_{i=1}^{m} a_i L_i f, \quad a = [a_1, a_2, \ldots, a_m]^T = (G^T)^{-1}\lambda. \tag{5.1.10}$$

**Metoda coeficienţilor nedeterminaţi.** Aici determinăm coeficienţii din (5.1.3) astfel încât egalitatea să aibă loc $\forall\, f \in \Phi$, care, conform liniarităţii lui $L$ şi $L_i$ este echivalentă cu egalitatea pentru $f = \varphi_1$, $f = \varphi_2, \ldots,$ $f = \varphi_m$, adică

$$\left(\sum_{j=1}^{m} a_j L_j\right) \varphi_i = L\varphi_i, \quad i = \overline{1,m},$$

sau conform (5.1.8)

$$\sum_{j=1}^{m} a_j L_j \varphi_i = \lambda_i, \quad i = \overline{1,m}.$$

Evident, matricea sistemului este $G^T$, deci

$$a = [a_1, a_2, \ldots, a_m]^T = (G^T)^{-1}\lambda,$$

în concordanţă cu (5.1.10). Astfel, metoda interpolării şi cea a coeficienţilor nedeterminaţi sunt matematic echivalente – ele conduc la exact aceeaşi aproximare.

S-ar părea că, cel puţin în cazul polinoamelor (adică $\Phi = \mathbb{P}_d$), prima metodă este mai puternică, deoarece poate conduce la o expresie a erorii de interpolare (aplicând funcţionala formulei de interpolare $f = a_n f + r_n f$). Dar şi în cazul metodei coeficienţilor nedeterminaţi, din condiţia de exactitate, se poate exprima restul cu ajutorul teoremei lui Peano (teorema 3.3.2).

## 5.2. Derivare numerică

Pentru simplitate vom considera doar derivata de ordinul I. Se pot aplica tehnici analoage şi pentru alte derivate. Vom rezolva problema prin interpolare: în loc să derivăm $f \in C^{m+1}[a, b]$, vom deriva polinomul său de interpolare:

$$f(x) = (L_m f)(x) + (R_m f)(x). \tag{5.2.1}$$

Scriem polinomul de interpolare în forma Newton

$$\begin{aligned}(L_m f)(x) = (N_m f)(x) = f_0 + (x - x_0) f[x_0, x_1] + \cdots + \\ + (x - x_0)\ldots(x - x_{m-1}) f[x_0, x_1, \ldots, x_m]\end{aligned} \tag{5.2.2}$$

şi restul sub forma

$$(R_m f)(x) = (x - x_0)\ldots(x - x_m)\frac{f^{(m+1)}(\xi(x))}{(m+1)!}. \tag{5.2.3}$$

Derivând (5.2.2) în raport cu $x$ şi punând $x = x_0$ obţinem

$$\begin{aligned}(L_m f)(x_0) = f[x_0, x_1] + (x_0 - x_1) f[x_0, x_1, x_2] + \cdots + \\ + (x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_{m-1}) f[x_0, x_1, \ldots, x_m].\end{aligned} \tag{5.2.4}$$

Presupunând că $f$ este continuu derivabilă pe un interval convenabil se obţine pentru rest

$$(R_m f)'(x_0) = (x_0 - x_1)\ldots(x_0 - x_m)\frac{f^{(m+1)}(\xi(x_0))}{(m+1)!}. \tag{5.2.5}$$

Deci, derivând (5.2.4) obţinem

$$f'(x_0) = (L_m f)'(x_0) + \underbrace{(R_m f)'(x_0)}_{e_m}. \tag{5.2.6}$$

Dacă $H = \max_i |x_0 - x_i|$ eroarea are forma $e_m = O(H^m)$, când $H \to 0$.

Putem obţine formule de aproximare de grad arbitrar, dar ele sunt de utilitate practică limitată.

**Observaţia 5.2.1.** Derivarea numerică este o operaţie critică şi de aceea este bine să fie evitată pe cât posibil, deoarece chiar dacă aproximanta este bună, nu rezultă că derivata aproximantei este o aproximaţie bună a derivatei (vezi figura 5.1). Aceasta rezultă şi din exemplul 5.2.2 ♢

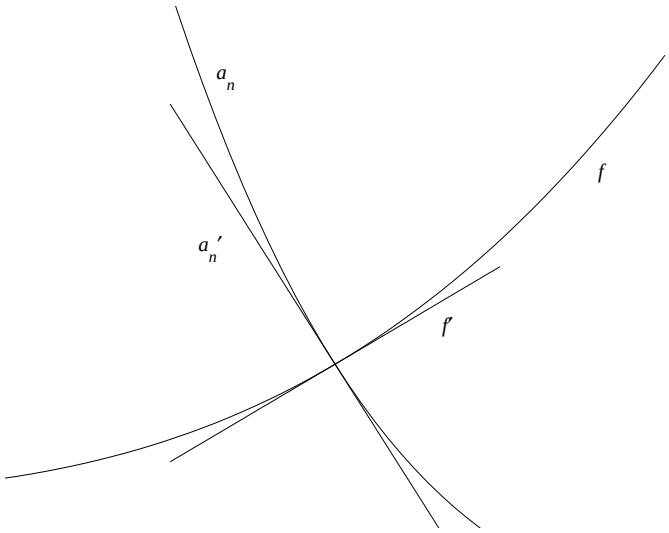



Figura 5.1: Neajunsurile derivării numerice

**Exemplul 5.2.2.** Fie funcţia

$$f(x) = g(x) + \frac{1}{n}\sin n^2(x-a), \quad g \in C^1[a,b].$$

Se constată că $d(f,g) \to 0 \quad (n \to \infty)$, dar $d(f',g') = n \nrightarrow 0$. ♢

Formulele de derivare numerică sunt utile pentru deducerea unor metode numerice, în special pentru ecuaţii diferenţiale ordinare şi ecuaţii cu derivate parţiale.

Se pot folosi şi alte procedee de aproximare: Taylor, Hermite, spline, metoda celor mai mici pătrate.

## 5.3. Integrare numerică

Problema este de a calcula integrala definită a unei funcţii date pe un interval mărginit $[a, b]$. Dacă $f$ are o comportare bună, aceasta este o problemă de rutină, pentru care metodele cele mai simple de integrare cum ar fi regula trapezelor sau regula repetată a lui Simpson sunt satisfăcătoare, prima având anumite avantaje asupra celei de-a doua în cazul când $f$ este periodică, cu perioada $b - a$. Complicaţiile apar atunci când $f$ are singularităţi (dar rămâne integrabilă), sau când intervalul de integrare este nemărginit (care este o altă manifestare a comportării singulare). Descompunând integrala pe subintervale, dacă este necesar, în mai multe integrale, se poate presupune că singularitatea, dacă locul ei este cunoscut, este la unul sau la ambele capete ale intervalului $[a, b]$. Astfel de integrale improprii pot fi tratate ca şi cuadraturi cu ponderi, adică să încorporăm singularitatea într-o pondere, care devine un factor al integrandului, lăsând celălalt factor să aibă o comportare bună. Cel mai important exemplu este formula lui Gauss relativă la o astfel de pondere. În fine, este posibil să se accelereze convergenţa unei scheme de cuadratură prin recombinări convenabile. Un astfel de exemplu este metoda lui Romberg.

Fie $f : [a, b] \to \mathbb{R}$ integrabilă pe $[a, b]$, $F_k(f)$, $k = \overline{0, m}$ informaţii despre $f$ (de regulă funcţionale liniare) şi $w : [a, b] \to \mathbb{R}_+$ o funcţie pondere integrabilă pe $[a, b]$.

**Definiţia 5.3.1** *O formulă de forma*

$$\int_a^b w(x)f(x)dx = Q(f) + R(f), \tag{5.3.1}$$

*unde*

$$Q(f) = \sum_{j=0}^{m} A_j F_j(f),$$

*se numeşte* formulă de integrare numerică *a funcţiei $f$ sau* formulă de cuadratură. *Parametrii $A_j$, $j = \overline{0, m}$ se numesc* coeficienţii formulei, *iar $R(f)$* termenul rest *al ei. $Q$ se numeşte* funcţională de cuadratură.

**Definiţia 5.3.2** *Numărul natural $d = d(Q)$ cu proprietatea că $\forall\ f \in \mathbb{P}d$, $R(f) = 0$ şi $\exists\ g \in \mathbb{P}d + 1$ astfel încât $R(g) \neq 0$ se numeşte* grad de exactitate *al formulei de cuadratură .*

Deoarece $R$ este liniar, rezultă că o formulă de cuadratură are gradul de exactitate $d$ dacă şi numai dacă $R(e_j) = 0$, $j = \overline{0, d}$ şi $R(e_{d+1}) \neq 0$.

Dacă gradul de exactitate al unei formule de cuadratură este cunoscut, restul se poate determina cu ajutorul teoremei lui Peano.

## 5.3.1. Formula trapezului şi formula lui Simpson

Aceste formule au fost denumite de Gautschi în [20] „caii de bătaie" ai integrării numerice. Ele îşi fac bine munca când intervalul de integrare este mărginit şi integrandul este neproblematic. Formula trapezelor este surprinzător de eficientă chiar şi pentru intervale infinite.

Ambele reguli se obţin aplicând cele mai simple tipuri de interpolare subintervalelor diviziunii

$$a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b, \quad x_k = a + kh, \quad h = \frac{b-a}{n}. \tag{5.3.2}$$

În cazul regulii trapezelor se interpolează liniar pe fiecare subinterval $[x_k, x_{k+1}]$ şi se obţine

$$\int_{x_k}^{x_{k+1}} f(x)dx = \int_{x_k}^{x_{k+1}} (L_1 f)(x)dx + \int_{x_k}^{x_{k+1}} (R_1 f)(x)dx, \quad f \in C^1[a,b], \tag{5.3.3}$$

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

$$R_1(f) = \int_{x_k}^{x_{k+1}} K_1(t) f''(t)dt,$$

iar

$$\begin{aligned} K_1(t) &= \frac{(x_{k+1}-t)^2}{2} - \frac{h}{2}[(x_k - t)_+ + (x_{k+1} - t)_+] = \\ &= \frac{(x_k - t)^2}{2} - \frac{h(x_{k+1} - t)}{2} = \\ &= \frac{1}{2}(x_{k+1} - t)(x_k - t) \le 0. \end{aligned}$$

Deci

$$R_1(f) = -\frac{h^3}{12} f''(\xi_k), \quad \xi_k \in (x_k, x_{k+1})$$

şi

$$\int_{x_k}^{x_{k+1}} f(x)dx = \frac{h}{2}(f_k + f_{k+1}) - \frac{1}{12} h^3 f''(\xi_k). \tag{5.3.4}$$

Această formulă se numeşte *regula (elementară a) trapezului*. Însumând pentru toate subintervalele se obţine *regula trapezelor* sau *formula compusă a trapezului* sau *formula repetată a trapezului*.

$$\int_a^b f(x)dx = h\left(\frac{1}{2}f_0 + f_1 + \cdots + f_{n-1} + \frac{1}{2}f_n\right) - \frac{1}{12}h^3\sum_{k=0}^{n-1} f''(\xi_k).$$

Deoarece $f''$ este continuă pe $[a,b]$, restul se poate scrie sub forma

$$R_{1,n}(f) = -\frac{(b-a)h^2}{12}f''(\xi) = -\frac{(b-a)^3}{12n^2}f''(\xi). \tag{5.3.5}$$

Deoarece $f''$ este mărginită în modul pe $[a,b]$ avem

$$R_{1,n}(f) = O(h^2),$$

când $h \to 0$ şi deci regula trapezelor converge când $h \to 0$ (sau echivalent, $n \to \infty$), atunci când $f \in C^2[a,b]$.

Dacă în locul interpolării liniare se utilizează interpolarea pătratică se obţine *regula lui Simpson repetată* . Varianta ei elementară, numită *regula lui Simpson* [2] sau *formula lui Simpson* este

$$\int_{x_k}^{x_{k+2}} f(x)dx = \frac{h}{3}(f_k + 4f_{k+1} + f_{k+2}) - \frac{1}{90}h^5 f^{(h)}(\xi_k), \quad x_k \le \xi_k \le x_{k+2}, \tag{5.3.6}$$

unde s-a presupus că $f \in C^4[a,b]$.

Să demonstrăm formula pentru restul formulei lui Simpson. Deoarece gradul de exactitate este 3, conform teoremei lui Peano avem

$$R_2(f) = \int_{x_k}^{x_{k+2}} K_2(t) f^{(4)}(t)\,\mathrm{d}t.$$

unde

$$K_2(t) = \frac{1}{3!}\left\{\frac{(x_{k+2}-t)^4}{4} - \frac{h}{3}\left[(x_k-t)_+^3 + 4\,(x_{k+1}-t)_+^3 + (x_{k+2}-t)_+^3\right]\right\},$$

[2] Thomas Simpson (1710-1761), matematician englez autodidact, autor al mai multor texte, populare la vremea respectivă. Simpson şi-a publicat formula în 1743, dar ea a fost cunoscută deja, printre alţii, de Cavalieri (1639), Gregory (1668) şi Cotes (1722).

adică

$$K_2(t) = \frac{1}{6}\begin{cases} \frac{(x_{k+2}-t)^4}{4} - \frac{h}{3}\left[4\left(x_{k+1}-t\right)^3 + \left(x_{k+2}-t\right)^3\right], & t \in [x_k, x_{k+1}] \\ \frac{(x_{k+2}-t)^4}{4} - \frac{h}{3}(x_{k+2}-t)^3, & t \in [x_{k+1}, x_{k+2}] \end{cases}$$

Se arată că pentru $t \in [x_k, x_{k+2}]$, $K_2(t) \leq 0$ şi deci putem aplica corolarul la teorema lui Peano.

$$R_2(f) = \frac{1}{4!} f^{(4)}(\xi_k) R_2(e_4),$$

$$\begin{aligned} R_2(e_4) &= \frac{x_{k+2}^5 - x_k^5}{5} - \frac{h}{3}\left[x_k^4 + 4\left(\frac{x_{k+2}+x_k}{2}\right)^4 + x_{k+2}^4\right] \\ &= 3h\left[\frac{x_{k+2}^4 + x_{k+2}^3 x_k + x_{k+2}^2 x_k^2 + x_{k+2} x_k^3 + x_k^4}{5}\right. \\ &\left. - \frac{4x_k^4 + x_k^4 + 4x_k^3 x_{k+2} + 6x_k^2 x_{k+2}^2 + 4x_k x_{k+2}^3 + x_{k+2}^4 + 4x_{k+2}^4}{24}\right] \\ &= \frac{h}{40}\left(-x_k^4 + 4x_k^3 x_{k+2} + 6x_k^2 x_{k+2}^2 + 4x_k x_{k+2}^3 - x_{k+2}^4\right) \\ &= -\frac{h}{40}(x_{k+2} - x_k)^4. \end{aligned}$$

Deci,

$$R_2(f) = -\frac{h^5}{90} f^{(4)}(\xi_k).$$

Pentru regula repetată a lui Simpson obţinem

$$\int_a^b f(x)dx = \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 4f_{n-1} + f_n) + R_{2,n}(f) \quad (5.3.7)$$

cu

$$R_{2,n}(f) = -\frac{1}{180}(b-a)h^4 f^{(4)}(\xi) = -\frac{(b-a)^5}{2880n^4} f^{(4)}(\xi), \quad \xi \in (a,b). \quad (5.3.8)$$

Se observă că $R_{2,n}(f) = O(h^4)$, de unde rezultă convergenţa când $n \to \infty$. Se observă şi creşterea ordinului cu 1, ceea ce face ca regula repetată a lui Simpson să fie foarte populară şi larg utilizată.

Regula trapezelor lucrează foarte bine pentru polinoame trigonometrice. Presupunem fără a restrânge generalitatea că $[a,b] = [0, 2\pi]$ şi fie

$$\mathbb{T}_m[0,2\pi] = \{t(x) :\ t(x) = a_0 + a_1\cos x + a_2\cos 2x + \cdots + a_m \cos mx +$$

$$+b_1 \sin x + b_2 \sin 2x + \cdots + b_m \sin mx\}$$

Atunci

$$R_{n,1}(f) = 0, \ \forall \ f \in \mathbb{T}_{n-1}[0, 2\pi] \tag{5.3.9}$$

Aceasta se verifică luând

$$f(x) = e_\nu(x) = e^{i\nu x} = \cos \nu x + i \sin \nu x, \quad \nu = 0, 1, 2, \ldots$$

$$R_{n,1}(e_\nu) = \int_0^{2\pi} e_\nu(x)dx - \frac{2\pi}{n}\left[\frac{1}{2}e_\nu(0) + \sum_{k=1}^{n-1} e_\nu\left(\frac{2k\pi}{n}\right) + \frac{1}{2}e_\nu(2\pi)\right] =$$

$$= \int_0^{2\pi} e^{i\nu x}dx - \frac{2\pi}{n}\sum_{k=0}^{n-1} e^{\frac{2\pi i k\nu}{n}}.$$

Când $\nu = 0$, aceasta este evident 0 şi în caz contrar, deoarece

$$\int_0^{2\pi} e^{i\nu x}dx = (i\nu)^{-1}e^{i\nu x}\Big|_0^{2\pi} = 0,$$

$$R_{n,1}(e_\nu) = \begin{cases} -2\pi & \text{dacă} \quad \nu = 0 \ (\mathrm{mod}\ n), \ \nu > 0 \\ -\dfrac{2\pi}{n}\dfrac{1 - e^{i\nu n \cdot 2\pi/n}}{1 - e^{i\nu \cdot 2\pi/n}} = 0 & \text{dacă} \quad \nu \neq 0 \ (\mathrm{mod}\ n) \end{cases} \tag{5.3.10}$$

În particular, $R_{n,1}(e_\nu) = 0$ pentru $\nu = 0, 1, \ldots, n-1$, ceea ce demonstrează (5.3.9). Luând partea reală şi imaginară în (5.3.10) se obţine

$$R_{n,1}(\cos \nu \cdot) = \begin{cases} -2\pi, & \nu = 0(\mathrm{mod} n), \ \nu \neq 0 \\ 0, & \text{în caz contrar} \end{cases}$$

$$R_{n,1}(\sin \nu \cdot) = 0.$$

De aceea, dacă $f$ este $2\pi$-periodică şi are o dezvoltare Fourier uniform convergentă

$$f(x) = \sum_{\nu=0}^{\infty}[a_\nu(f)\cos \nu x + b_\nu(f)\sin \nu x], \tag{5.3.11}$$

unde $a_\nu(f), b_\nu(f)$ sunt coeficienţii Fourier ai lui $f$, atunci

$$R_{n,1}(f) = \sum_{\nu=0}^{\infty}[a_\nu(f)R_{n,1}(\cos \nu \cdot) + b_\nu(f)R_{n,1}(\sin \nu \cdot)] =$$

$$= -2\pi \sum_{l=1}^{\infty} a_{ln}(f) \tag{5.3.12}$$

Din teoria seriilor Fourier se ştie că coeficienţii Fourier ai lui $f$ tind către zero cu atât mai repede cu cât $f$ este mai netedă. Mai exact, dacă $f \in C^r[\mathbb{R}]$, atunci $a_\nu(f) = O(\nu^{-r})$ când $\nu \to \infty$ (şi similar pentru $b_\nu(f)$). Deoarece conform (5.3.12)

$$R_{n,1}(f) \simeq -2\pi a_n(f),$$

rezultă că

$$R_{n,1}(f) = O(n^{-r}) \text{ când } n \to \infty \tag{5.3.13}$$

($f \in C^r[\mathbb{R}]$, $2\pi$ periodică), care dacă $r > 2$ este mai bună decât $R_{n,1}(f) = O(n^{-2})$, valabilă pentru funcţii $f$ neperiodice. În particular, dacă $r = \infty$, atunci regula trapezului converge mai repede decât orice putere a lui $n^{-1}$. Trebuie observat totuşi că $f$ trebuie să fie netedă pe întreg $\mathbb{R}$. Pornind de la o funcţie $f \in C^r[0, 2\pi]$ şi prelungind-o pe întreaga axă reală prin periodicitate, în general nu se obţine o funcţie $f \in C^r[\mathbb{R}]$.

O altă împrejurare în care regula trapezelor excelează este pentru funcţii definite pe $\mathbb{R}$ şi care au proprietatea următoare pentru un anumit $r \geq 1$

$$f \in C^{2r+1}[\mathbb{R}], \quad \int_{\mathbb{R}} \left| f^{(2r+1)}(x) \right| dx < \infty,$$

$$\lim_{x \to -\infty} f^{(2\rho-1)}(x) = \lim_{x \to +\infty} f^{(2\rho-1)}(x) = 0, \quad \rho = 1, 2, \ldots, r. \tag{5.3.14}$$

În acest caz se poate arăta că

$$\int_{\mathbb{R}} f(x)dx = h \sum_{k=-\infty}^{\infty} f(kh) + R(f; h) \tag{5.3.15}$$

are o eroare $R(f, h)$ ce satisface $R(f; h) = O(h^{2r+1})$, $h \to \infty$. Deci, dacă (5.3.14) are loc pentru orice $r \in \mathbb{N}$, atunci eroarea tinde la zero mai repede decât orice putere a lui $h$.

## 5.3.2. Formule Newton-Cotes cu ponderi şi formule de tip Gauss

O *formulă de cuadratură cu ponderi* este o formulă de tipul

$$\int_a^b f(t)w(t)dt = \sum_{k=1}^{n} w_k f(t_k) + R_n(f) \tag{5.3.16}$$

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 (5.3.16) este bine definită, cel puţin în cazul când $f$ este polinom. Realizăm aceasta cerând ca toate momentele funcţiei pondere

$$\mu_s = \int_a^b t^s w(t)dt, \quad s = 0, 1, 2, \ldots \tag{5.3.17}$$

să existe şi să fie finite.

Spunem că (5.3.16) este de *tip interpolator*, dacă are gradul de exactitate $d = n-1$. Formulele de tip interpolator sunt chiar formulele obţinute prin interpolare, adică pentru care

$$\sum_{k=1}^{n} w_k f(t_k) = \int_a^b L_{n-1}(f; t_1, \ldots, t_n, t) w(t) dt \tag{5.3.18}$$

sau echivalent

$$w_k = \int_a^b \ell_k(t) w(t) dt, \quad k = 1, 2, \ldots, n, \tag{5.3.19}$$

unde

$$\ell_k(t) = \prod_{\substack{l=1 \\ l \neq k}}^{n} \frac{t - t_l}{t_k - t_l} \tag{5.3.20}$$

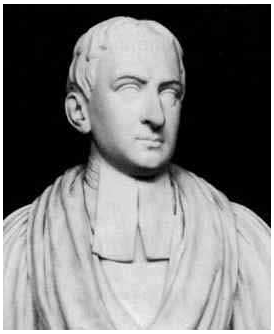
sunt polinoamele fundamentale Lagrange asociate nodurilor $t_1, t_2, \ldots, t_n$. Faptul că (5.3.16) are gradul de exactitate $d = n-1$ este evident, deoarece pentru orice $L_{n-1}(f; \cdot) \equiv f(\cdot)$ în (5.3.18). Reciproc, dacă (5.3.16) are gradul de exactitate $d = n-1$, atunci luând $f(t) = \ell_r(t)$ în (5.3.17) ne dă

$$\int_a^b \ell_r(t) w(t) dt = \sum_{k=1}^{n} w_k \ell_r(t_k) = w_r, \quad r = 1, 2, \ldots, n,$$

adică (5.3.19).

Observăm că dacă se dau $n$ noduri distincte $t_1, \ldots, t_n$ este posibil întotdeauna să construim o formulă de tip (5.3.16) care este exactă pentru orice polinom de grad $\leq n-1$. În cazul $w(t) \equiv 1$ pe $[-1, 1]$ şi $t_k$ sunt echidistante pe $[-1, 1]$ problema a fost intuită de Newton în 1687 şi rezolvată în detaliu de Cotes [3] în jurul anului 1712. Prin extensie vom numi formula (5.3.16) cu $t_k$ prescrise şi $w_k$ date de (5.3.19) *formulă de tip Newton-Cotes* .

Chestiunea care se pune în mod natural este dacă nu am putea face aceasta mai bine, adică dacă nu am putea obţine gradul de exactitate $d > n-1$ printr-o alegere judicioasă



[3] Roger Cotes (1682-1716), fiul precoce al unui pastor de ţară englez, a fost însărcinat cu pregătirea celei de-a doua ediţii a lucrării lui Newton *Principia*. El a dezvoltat idea lui Newton referitoare la integrarea numerică şi a publicat coeficienţii pentru formulele de integrare numerică cu $n$-puncte, pentru $n < 11$ (numere Cotes). La moartea sa prematură la vârsta de 33 de ani, Newton a spus despre el: „Daca ar fi trăit, am fi putut şti ceva."

a nodurilor $t_k$ (ponderile $w_k$ fiind date în mod necesar de (5.3.19)). Răspunsul este surprinzător de simplu şi de direct. Pentru a-l formula, considerăm polinomul nodurilor

$$u_n(t) = \prod_{k=1}^{n}(t - t_k). \tag{5.3.21}$$

**Teorema 5.3.3** *Dându-se un întreg $k$, $0 \leq k \leq n$, formula de cuadratură (5.3.16) are gradul de exactitate $d = n - 1 + k$ dacă şi numai dacă sunt satisfăcute următoarele condiţii:*

*(a) formula (5.3.16) este de tip interpolator ;*

*(b) polinomul nodurilor $u_n$ din (5.3.21) satisface*

$$\int_a^b u_n(t)p(t)w(t)\,\mathrm{d}t = 0,\ \forall\, p \in \mathbb{P}_{k-1}.$$

Condiţia din (b) impune $k$ restricţii asupra nodurilor $t_1, t_2, \ldots, t_n$ din (5.3.16). (Dacă $k = 0$, nu avem nici o restricţie, deoarece, aşa cum ştim, putem atinge gradul de exactitate $d = n-1$). Într-adevăr $u_n$ trebuie să fie ortogonale pe $\mathbb{P}_{k-1}$ relativ la funcţia 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 însuşi, ceea ce este imposibil. Astfel $k = n$ este optimal, obţinându-se o formulă de cuadratură cu gradul maxim de exactitate $d_{max} = 2n - 1$. Condiţia (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 numeşte *formulă de cuadratură de tip Gauss* asociată cu funcţia pondere $w$. Deci nodurile ei sunt zerourile lui $\pi_n(\cdot, w)$, iar ponderile (coeficienţii) $w_k$ sunt daţi de (5.3.19) adică

$$\begin{aligned} &\pi_n(t_k; w) = 0 \\ &w_k = \int_a^b \frac{\pi_n(t, w)}{(t - t_k)\pi_n'(t_k, w)} w(t)dt, \quad k = 1, 2, \ldots, n \end{aligned} \tag{5.3.22}$$

Formula a fost dezvoltată de Gauss în 1814 în cazul special $w(t) \equiv 1$ pe $[-1, 1]$ şi extinsă la funcţii pondere mai generale de către Christoffel [4] în 1877. De aceea se mai numeşte *formulă de cuadratură Gauss-Christoffel* .

[4] Elvin Bruno Christoffel (1829-1900), a activat pentru perioade scurte la Berlin şi Zürich şi cea mai mare parte a vieţii la Strasbourg. Este cunoscut pentru lucrările sale de geometrie, în particular de analiza tensorilor, care a jucat un rol important în teoria relativităţii a lui Einstein.

*Demonstraţia teoremei 5.3.3.* Vom demonstra întâi necesitatea lui (a) şi (b). Deoarece gradul de exactitate este $d = n - 1 + k \geq n - 1$, condiţia (a) este trivială. Condiţia (b) rezultă de asemenea imediat, deoarece pentru orice $p \in \mathbb{P}_{k-1}$, $u_n p \in \mathbb{P}_{n-1+k}$. Deci

$$\int_a^b u_n(t)p(t)w(t) = \sum_{k=1}^{n} w_k u_k(t_k)p(t_k),$$

care se anulează, căci $u_n(t_k) = 0$ pentru $k = 1, 2, \ldots, n$.

Pentru a demonstra suficienţa lui (a) şi (b) trebuie să arătăm că pentru orice $p \in \mathbb{P}_{n-1+k}$ avem $R_n(p) = 0$ în (5.3.16). Dându-se orice astfel de $p$, îl împărţim 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 şi $r$ 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 demonstraţia. □

Cazul $k = n$ va fi discutat în secţiunea 5.3.3. Vom menţiona două cazuri importante când $k < n$, care sunt de interes practic. Primul este formula de cuadratură Gauss-Radau în care o extremitate de interval, de exemplu $a$, este finită şi serveşte ca nod, să zicem $t_1 = a$. Gradul maxim de exactitate care se poate obţine este $d = 2n - 2$ şi corespunde lui $k = n - 1$ în teorema (5.3.3). Partea (b) a teoremei ne spune că nodurile rămase $t_2, \ldots, t_n$ trebuie să fie rădăcinile polinomului $\pi_{n-1}(\cdot, w_a)$, unde $w_a(t) = (t - a)w(t)$. La fel, în formulele Gauss-Lobatto, ambele capete sunt finite şi servesc ca noduri, să zicem $t_1 = a$, $t_n = b$, iar nodurile rămase $t_2, \ldots, t_{n-1}$ sunt zerourile lui $\pi_{n-2}(\cdot; w_{a,b})$, $w_{a,b}(t) = (t - a)(b - t)w(t)$, obţinându-se astfel gradul de exactitate $d = 2n - 3$.

## 5.3.3. Proprietăţi ale cuadraturilor gaussiene

Regula de cuadratură a lui Gauss, dată de (5.3.16) şi (5.3.22), pe lângă faptul că este optimală (adică are grad maxim de exactitate) are şi unele proprietăţi interesante.

(i) Toate nodurile sunt reale, distincte şi situate în intervalul deschis $(a, b)$. Aceasta este o proprietate cunoscută satisfăcută de zerourile polinoamelor ortogonale.

(ii) Toţi coeficienţii (ponderile) $w_k$ sunt pozitivi. Demonstraţia se bazează pe o observaţie ingenioasă a lui Stieltjes

$$0 < \int_a^b \ell_j^2(t)w(t)dt = \sum_{k=1}^n w_k \ell_j^2(t_k) = w_j, \quad j = 1, 2, \ldots, n,$$

prima egalitate rezultând din faptul că gradul de exactitate este $d = 2n - 1$.

(iii) Dacă $[a, b]$ este mărginit, atunci formula lui Gauss converge pentru orice funcţie continuă. Adică $R_n(f) \to 0$, când $n \to \infty$, pentru orice $f \in C[a, b]$. Aceasta este o consecinţă a teoremei de aproximare a lui Weierstrass, din care rezultă că dacă $\widehat{p}_{2n-1}(f; \cdot)$ este polinomul de cea mai bună aproximare a lui $f$ pe $[a, b]$ în sensul normei uniforme atunci

$$\lim_{n\to\infty} \|f(\cdot) - \widehat{p}_{2k-1}(f; \cdot)\|_\infty = 0.$$

Deoarece $R_n(\widehat{p}_{2n-1}) = 0$ (căci $d = 2n - 1$), avem succesiv

$$\begin{aligned}
|R_n(f)| &= |R_n(f - \widehat{p}_{2n-1})| = \\
&= \left| \int_a^b [f(t) - \widehat{p}_{2n-1}(f; t)]w(t)dt - \sum_{k=1}^n w_k[f(t_k) - \widehat{p}_{2n-1}(f; t_k)] \right| \leq \\
&\leq \int_a^b |f(t) - \widehat{p}_{2n-1}(f; t)|w(t)dt + \sum_{k=1}^n w_k |f(t_k) - \widehat{p}_{2n-1}(f; t_k)| \leq \\
&\leq \|f(\cdot) - \widehat{p}_{2n-1}(f; \cdot)\|_\infty \left[ \int_a^b w(t)dt + \sum_{k=1}^n w_k \right].
\end{aligned}$$

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

$$|R_n(f)| \leq 2\mu_0 \|f - \widehat{p}_{2n-1}\|_\infty \to 0, \text{ când } n \to \infty.$$

Proprietatea care urmează este baza unui algoritm eficient de obţinere a unor formule de cuadratură gaussiană.

(iv) Fie $\alpha_k = \alpha_k(w)$ şi $\beta_k = \beta_k(w)$ coeficienţii din formula de recurenţă pentru polinoamele ortogonale

$$\begin{aligned}
&\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k \pi_{k-1}(t), \qquad k = 0, 1, 2, \ldots \\
&\pi_0(t) = 1, \qquad \pi_{-1}(t) = 0,
\end{aligned} \tag{5.3.23}$$

unde

$$\begin{aligned} \alpha_k &= \frac{(t\pi_k, \pi_k)}{(\pi_k, \pi_k)} \\ \beta_k &= \frac{(\pi_k, \pi_k)}{(\pi_{k-1}, \pi_{k-1})}, \end{aligned} \tag{5.3.24}$$

cu $\beta_0$ definit (ca de obicei) prin

$$\beta_0 = \int_a^b w(t)dt \ (= \mu_0).$$

*Matricea Jacobi* de ordinul $n$ pentru funcţia pondere $w$ este o matrice simetrică tridiagonală definită prin

$$J_n(w) = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & 0 \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & & \ddots & \\ & & \ddots & \ddots & \sqrt{\beta_{n-1}} \\ 0 & & & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix}.$$

**Teorema 5.3.4** *Nodurile $t_k$ ale unei formule de tip Gausss sunt valorile proprii ale lui $J_n$*

$$J_n v_k = t_k v_k, \quad v_k^T v_k = 1, \quad k = 1, 2, \ldots, n, \tag{5.3.25}$$

*iar ponderile $w_k$ sunt exprimabile cu ajutorul componentelor $v_k$, ale vectorilor proprii normalizaţi corespunzători prin*

$$w_k = \beta_0 v_{k,1}^2, \quad k = 1, 2, \ldots, n \tag{5.3.26}$$

Astfel, pentru a obţine o formulă de cuadratură Gaussiană trebuie rezolvată o problemă de vectori şi valori proprii pentru o matrice tridiagonală simetrică. Pentru această problemă există metode foarte eficiente. Astfel, abordarea bazată pe valori şi vectori proprii este mai eficientă decât cea clasică. În plus, abordarea clasică se bazează pe două probleme prost condiţionate: rezolvarea ecuaţiilor polinomiale (coeficienţii sunt obţinuţi prin aplicarea de relaţii de recurenţă, deci sunt deja perturbaţi) şi rezolvarea unui sistem de ecuaţii având matricea Vandermonde.

*Demonstraţia teoremei 5.3.4.* Fie $\tilde{\pi}_k(\cdot) = \tilde{\pi}_k(\cdot, w)$ polinomul ortogonal normalizat, deci $\pi_k = \sqrt{(\pi_k, \pi_k)_{\mathrm{d}\lambda}}\tilde{\pi}_k$. Inserând aceasta în (5.3.23), împărţind cu $\sqrt{(\pi_k, \pi_k)_{\mathrm{d}\lambda}}$ şi utilizând (5.3.24), se obţine

$$\tilde{\pi}_{k+1}(t) = (t - \alpha_k)\frac{\tilde{\pi}_k}{\sqrt{\beta_{k+1}}} - \beta_k \frac{\tilde{\pi}_{k-1}}{\sqrt{\beta_{k+1}\beta_k}},$$

sau înmulţind cu $\sqrt{\beta_{k+1}}$ şi reordonând,

$$t\tilde{\pi}_k(t) = \alpha_k\tilde{\pi}_k(t) + \sqrt{\beta_k}\tilde{\pi}_{k-1}(t) + \sqrt{\beta_{k+1}}\tilde{\pi}_{k+1}(t), \quad k = 0, 1, \ldots n-1. \tag{5.3.27}$$

Cu ajutorul matricei lui Jacobi $J_n$, putem scrie această relaţie sub forma vectorială

$$t\tilde{\pi}(t) = J_n\tilde{\pi}(t) + \sqrt{\beta_n}\tilde{\pi}_n(t)e_n, \tag{5.3.28}$$

unde $\tilde{\pi}(t) = [\tilde{\pi}_0(t), \tilde{\pi}_1(t), \ldots, \tilde{\pi}_{n-1}(t)]^T$ şi $e_n(t) = [0, 0, \ldots, 0, 1]^T$ sunt vectori din $\mathbb{R}^n$. Deoarece $t_k$ sunt zerouri ale lui $\tilde{\pi}_n$ rezultă din (5.3.28) că

$$t_k\tilde{\pi}(t_k) = J_n\tilde{\pi}(t_k), \qquad k = 1, 2, \ldots, n. \tag{5.3.29}$$

Aceasta demonstrează prima relaţie a teoremei 5.3.4, deoarece $\tilde{\pi}$ este un vector nenul cu prima componentă

$$\tilde{\pi}_0 = \beta_0^{-1/2}. \tag{5.3.30}$$

Pentru a demonstra a doua relaţie, observăm din (5.3.29) că vectorul propriu normalizat $v_k$ este

$$v_k = \frac{1}{[\tilde{\pi}(t_k)^T\tilde{\pi}(t_k)]}\tilde{\pi}(t_k) = \left(\sum_{\mu=1}^{n}\tilde{\pi}_{\mu-1}^2(t_k)\right)^{-1/2}\tilde{\pi}(t_k).$$

Comparând prima componentă din ambi membrii şi ridicând la pătrat pe baza formulei (5.3.30)

$$\frac{1}{\sum_{\mu=1}^{n}\tilde{pi}_{\mu-1}^2(t_k)} = \beta_0 v_{k,1}^2, \qquad k = 1, 2, \ldots, n. \tag{5.3.31}$$

Pe de altă parte, luând $f(t) = \tilde{\pi}_{\mu-1}(t)$ în formula de cuadratură de tip Gauss (5.3.16), utilizând ortogonalitatea şi din nou (5.3.30) se obţine că

$$\beta_0^{1/2}\delta_{\mu-1,0} = \sum_{k=0}^{n} w_k\tilde{\pi}_{\mu-1}(t_k)$$

sau în formă matricială

$$Pw = \beta_0^{1/2}e_1, \tag{5.3.32}$$

unde $\delta_{\mu-1,0}$ este simbolul lui Kronecker, $P \in \mathbb{R}^{n\times n}$ este matricea vectorilor proprii, $w \in \mathbb{R}^n$ este vectorul coeficienţilor gaussieni şi $e_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^n$. Deoarece coloanele lui $P$ sunt ortogonale, avem

$$P^TP = D, \qquad D = \operatorname{diag}(d_1, d_2, \ldots, d_n), \qquad d_k = \sum_{\mu=1}^{n}\tilde{\pi}_{\mu-1}^2(t_k).$$

Înmulţim acum (5.3.32) la stânga cu $P^T$ şi obţinem

$$Dw = \beta_0^{1/2} P^T e_1 = \beta_0^{1/2} * \beta_0^{-1/2} e = e, \qquad e = [1, 1, \ldots, 1]^T.$$

Deci, $w = D^{-1}e$, adică,

$$w_k = \frac{1}{\sum_{\mu=1}^{n} \tilde{\pi}_{\mu-1}^2(t_k)}, \qquad k = 1, 2, \ldots, n.$$

Comparând cu (5.3.31) se obţine rezultatul dorit. □

Pentru detalii privind aspectele algoritmice legate de polinoame ortogonale şi cuadraturi gaussiene a se vedea [21].

(v) Markov [5] a observat că formula de cuadratură a lui Gauss poate fi obţinută cu ajutorul formulei de interpolare a lui Hermite cu noduri duble:

$$f(x) = (H_{2n-1}f)(x) + u_n^2(x) f[x, x_1, x_1, \ldots, x_n, x_n],$$

$$\int_a^b w(x) f(x) dx = \int_a^b w(x)(H_{2n-1}f)(x) dx + \\ + \int_a^b w(x) u_n^2(x) f[x, x_1, x_1, \ldots, x_n, x_n] dx.$$

Dar gradul de exactitate $2n-1$ implică

$$\int_a^b w(x)(H_{2n-1}f)(x) dx = \sum_{i=1}^{n} w_i (H_{2n-1}f)(x_i) = \sum_{i=1}^{n} w_i f(x_i),$$

$$\int_a^b w(x) f(x) dx = \sum_{i=1}^{n} w_i f(x_i) + \int_a^b w(x) u^2(x) f[x, x_1, x_1, \ldots, x_n, x_n] dx,$$

[5] 

Andrei Andreievici Markov (1856-1922), matematician rus, activ în Sankt Petersburg. A avut contribuţii importante în teoria probabilităţilor, teoria numerelor şi teoria constructivă a aproximării.

deci

$$R_n(f) = \int_a^b w(x)u_n^2(x)f[x, x_1, x_1, \ldots, x_n, x_n]dx.$$

Cum $w(x)u^2(x) \geq 0$, aplicând teorema de medie pentru integrale şi teorema de medie pentru diferenţe divizate avem

$$R_n(f) = f[\eta, x_1, x_1, \ldots, x_n, x_n] \int_a^b w(x)u^2(x)dx =$$
$$= \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b w(x)[\pi_n(x, w)]^2 dx, \quad \xi \in [a, b].$$

Vom încheia secţiunea cu o tabelă cu funcţiile pondere clasice, polinoamele lor ortogonale corespunzătoare şi coeficienţii din formula de recurenţă $\alpha_k, \beta_k$ (vezi tabela 5.1).

| polinoamele | notaţia | ponderea | intervalul | $\alpha_k$ | $\beta_k$ |
|---|---|---|---|---|---|
| Legendre | $P_n(l_n)$ | 1 | [-1,1] | 0 | 2 ($k$=0)<br>$(4-k^{-2})^{-1}$ ($k$>0) |
| Cebâşev #1 | $T_n$ | $(1-t^2)^{-\frac{1}{2}}$ | $[-1,1]$ | 0 | $\pi$ ($k$=0)<br>$\frac{1}{2}\pi$ ($k$=1)<br>$\frac{1}{4}$ ($k$>0) |
| Cebâşev #2 | $u_n(Q_n)$ | $(1-t^2)^{\frac{1}{2}}$ | $[-1,1]$ | 0 | $\frac{1}{2}\pi$ ($k$=0)<br>$\frac{1}{4}$ ($k$>0) |
| Jacobi | $P_n^{(\alpha,\beta)}$ | $(1-t)^\alpha(1-t)^\beta$<br>$\alpha>-1$, $\beta>-1$ | $[-1,1]$ | vezi observaţia 5.3.5 | vezi observaţia 5.3.5 |
| Laguerre | $L_n^{(\alpha)}$ | $t^\alpha e^{-t}$ $\alpha>-1$ | $[0,\infty)$ | $2k+\alpha+1$ | $\Gamma(1+\alpha)$ ($k$=0)<br>$k(k+\alpha)$ ($k$>0) |
| Hermite | $H_n$ | $e^{-t^2}$ | $\mathbb{R}$ | 0 | $\sqrt{\pi}$ ($k$=0) |

Tabela 5.1: Polinoame ortogonale

**Observaţia 5.3.5.** Pentru polinoamele Jacobi avem

$$\alpha_k = \frac{\beta^2 - \alpha^2}{(2k+\alpha+\beta)(2k+\alpha+\beta+2)}$$

şi

$$\beta_0 = 2^{\alpha+\beta+1}B(\alpha+1, \beta+1),$$
$$\beta_k = \frac{4k(k+\alpha)(k+\alpha+\beta)}{(2k+\alpha+\beta-1)(2k+\alpha+\beta)^2(2k+\alpha+\beta+1)}, \quad k > 0. \qquad \diamondsuit$$

## 5.4. Cuadraturi adaptive

La metodele de integrare numerică erorile nu depind numai de dimensiunea intervalului utilizat, ci şi de valoarea derivatelor de un anumit ordin ale funcţiei care urmează a fi integrată. Aceasta implică faptul că metodele nu vor lucra bine pentru funcţii cu derivatele de un anumit ordin mari – în special funcţii care au fluctuaţii mari pe unele subintervale sau pe tot intervalul. Este rezonabil să utilizăm subintervale mici acolo unde derivatele sunt mari şi subintervale mari acolo unde derivatele sunt mici. O metodă care face aceasta într-o manieră sistematică se numeşte cuadratură adaptivă.

Abordarea generală într-o cuadratură adaptivă este de a utiliza două metode diferite pe fiecare subinterval, de a compara rezultatul şi de a subdiviza intervalul dacă diferenţele sunt mari. Există situaţia nefericită în care se utilizează două metode proaste, rezultatele sunt proaste, dar diferenţa dintre ele este mică. Un mod de a evita o astfel de situaţie 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 (algoritmul 5.1). Să presupunem că

$metint(a, b : real; f : functie, n : integer) : real$

este o funcţie care aproximează $\int_a^b f(x)dx$ folosind o cuadratură repetată cu $n$ subintervale. Pentru $m$ se alege o valoare mică (4 sau 5). Structura algoritmului: DIVIDE AND

**Algoritmul 5.1** Cuadratură adaptivă

**Intrare:** $f$ - funcţia de integrat, $a$, $b$ - limitele de integrare, $\varepsilon$ - toleranţa, $metint$ - o cuadratură repetată
**Ieşire:** 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**

CONQUER.

Spre deosebire de alte metode, la care se decide cât de mult se munceşte 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 paşi depinde de natura funcţiei de integrat. Posibilităţi de îmbunătăţire: $metint(a, b, f, 2m)$ este apelat de două ori, precizia poate fi scalată cu raportul dintre dimensiunea intervalului curent şi dimensiunea întregului interval. Pentru detalii suplimentare recomandăm [19].

# 5.5. Cuadraturi iterate. Metoda lui Romberg

Un dezavantaj al cuadraturilor adaptive este acela că calculează repetat valorile funcţiei în noduri, iar atunci când este rulat un astfel de program apare un consum suplimentar de timp de calcul datorită recursivităţii sau gestiunii stivei într-o implementare iterativă. Cuadraturile iterative înlătură aceste inconveniente. Ele aplică la primul pas o cuadratură repetată şi apoi subdivid intervalele în părţi egale folosind la fiecare pas aproximantele calculate anterior. Vom exemplifica această tehnică printr-o metodă care porneşte de la formula repetată a trapezului şi îmbunătăţeşte convergenţa utilizând extrapolarea Richardson.

Primul pas al procesului presupune aplicarea formulei repetate a trapezului cu $n_1 = 1$, $n_2 = 2, \ldots$, $n_p = 2^{p-1}$, unde $p \in \mathbb{N}^*$. Valoarea pasului $h_k$ corespunzătoare lui $n_k$ va fi

$$h_k = \frac{b-a}{n_k} = \frac{b-a}{2^{k-1}}.$$

Cu aceste notaţii regula trapezului devine

$$\int_a^b f(x)dx = \frac{h_k}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{2^{n-1}-1} f(a+ih_k)\right] - \frac{b-a}{12}h_k^2 f''(\mu_k) \qquad (5.5.1)$$

$\mu_k \in (a, b)$.

Notăm cu $R_{k,1}$ rezultatul aproximării conform (5.5.1).

$$R_{1,1} = \frac{h_1}{2}[f(a) + f(b)] = \frac{b-a}{2}[f(a) + f(b)] \qquad (5.5.2)$$

$$\begin{aligned} R_{2,1} &= \frac{h_2}{2}[f(a) + f(b) + 2f(a+h_2)] = \\ &= \frac{b-a}{4}\left[f(a) + f(b) + 2f\left(a + \frac{b-a}{2}\right)\right] = \\ &= \frac{1}{2}\left[R_{1,1} + h_1 f\left(a + \frac{1}{2}h_1\right)\right]. \end{aligned}$$

şi în general

$$R_{k,1} = \frac{1}{2}\left[R_{k-1,1} + h_{k-1}\sum_{i=1}^{2^{k-2}} f\left(a + \left(i - \frac{1}{2}\right)h_{k-1}\right)\right], \quad k = \overline{2,n} \qquad (5.5.3)$$

Urmează îmbunătăţirea prin extrapolare Richardson [6] (deşi a fost introdusă de Richardson şi Gaunt se pare că este datorată lui Arhimede [7])

$$I = \int_a^b f(x)dx = R_{k-1,1} - \frac{(b-a)}{12}h_k^2 f''(a) + O(h_k^4).$$

Vom elimina termenul în $h_k^2$ combinând două ecuaţii

$$\begin{aligned} I =& R_{k-1,1} - \frac{(b-a)}{12}h_k^2 f''(a) + O(h_k^4), \\ I =& R_{k,1} - \frac{b-a}{48}h_k^2 f''(a) + O(h_k^4). \end{aligned}$$

Obţinem

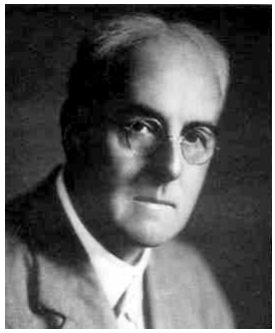
$$I = \frac{4R_{k,1} - R_{k-1,1}}{3} + O(h^4).$$

Definim

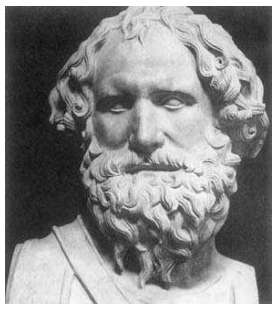
$$R_{k,2} = \frac{4R_{k,1} - R_{k-1,1}}{3}. \tag{5.5.4}$$

Se aplică extrapolarea Richardson şi acestor valori. În general dacă $f \in C^{2n+2}[a,b]$, atunci pentru $k = \overline{1,n}$ putem scrie

$$\int_a^b f(x)dx = \frac{h_k}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{2^{k-1}-1} f(a+ih_k)\right] + \tag{5.5.5}$$



[6] Lewis Fry Richardson (1881-1953), matematician englez. A avut contribuţii la predicţia numerică a vremii, propunând rezolvarea ecuaţiilor hidro şi termodinamice ale meteorologiei cu metode bazate pe diferenţe finite. A realizat un studiu de referinţă asupra turbulenţei atmosferice, introducând cantităţile numite astăzi „numerele Richardson". La 50 de ani şi-a luat licenţa în psihologie şi a dezvoltat o teorie ştiinţifică a relaţiilor internaţionale. A fost ales membru al Royal Society în 1926.



[7] Arhimede(287 î.e.n. - 212 î.e.n.), matematician grec din Siracuza, unul dintre cei mai importanţi ai întregii antichităţi. A pus la punct o metodă de integrare care i-a permis să determine arii, volume şi suprafeţe ale multor corpuri. A dat o aproximare precisă a lui $\pi$, metode de aproximare precisă a rădăcinii pătrate şi un sistem de reprezentare a numerelor mari. În mecanică, Arhimede a descoperit teoremele fundamentale referitoare la centrul de greutate al figurilor plane şi solidelor şi principiul care îi poartă numele, referitor la un corp scufundat într-un lichid. Maşinile sale de război au contribuit la apărarea oraşului său în timpul asediului romanilor, care a durat trei ani. A murit ucis de un soldat roman la sfârşitul asediului.

$$+\sum_{i=1}^{k} K_i h_k^{2i} + O(h_k^{2k+2}),$$

unde $K_i$ nu depinde de $h_k$.

Formula (5.5.5) se justifică în modul următor. Fie $a_0 = \int_a^b f(x)dx$ şi

$$A(h) = \frac{h}{2}\left[f(a) + 2\sum_{k=1}^{n-1} f(a+kh) + f(b)\right], \quad h = \frac{b-a}{k}.$$

Dacă $f \in C^{2k+1}[a,b]$, $k \in \mathbb{N}^*$ are loc următorul rezultat datorat lui Euler [8] şi MacLaurin [9]

$$A(h) = a_0 + a_1 h^2 + a_2 h^4 + \cdots + a_k h^{2k} + O(h^{2k+1}), \quad h \to 0 \tag{5.5.6}$$

unde

$$a_k = \frac{B_{2k}}{(2k)!}[f^{(2k-1)}(b) - f^{(2k-1)}(a)], \quad k = 1, 2, \ldots, K.$$

[8] Leonhard Euler (1707-1783), matematician elveţian, a urmat cursurile lui Jakob Bernoulli la Universitea din Basel, luând şi lecţii particulare de la Johann Bernoulli. După ce la 20 de ani nu a reuşit să obţină o catedră de fizică la Basel, a emigrat la Sankt Petersburg; mai târziu s-a mutat la Berlin şi apoi din nou la Sankt Petersburg. Indiscutabil, Euler a fost cel mai prolific matematician al secolului al XVIII-lea, lucând în aproape toate ramurile calculului diferenţial şi integral şi fiind unul dintre fondatorii calculului variaţional. A elaborat lucrări de pionierat în ştiinţele aplicate: hidrodinamică, mecanica materialelor deformabile şi solidului rigid, optică, astronomie. Nici chiar orbirea sa la vârsta de 59 de ani nu i-a afectat productivitatea fenomenală. Se pare că opera sa nu a fost încă editată în întregime, apărând până acum 71 de volume.

[9] Colin Maclaurin (1698-1768), matematician scoţian. A aplicat calculul infinitezimal la probleme de geometrie. Este cunoscut pentru dezvoltarea în serie în jurul originii dar a avut şi contribuţii la teoria ecuaţiilor.

Cantităţile $B_k$ sunt numerele lui Bernoulli [10], adică coeficienţii dezvoltării

$$\frac{z}{e^z-1}=\sum_{k=0}^{\infty}\frac{B_k}{k!}z^k,\quad |z|<2\pi.$$

Formula (5.5.6) se numeşte *formula Euler-MacLaurin* .
Eliminând succesiv puterile lui $h$ din (5.5.5) se obţine

$$R_{k,j}=\frac{4^{j-1}R_{k,j-1}-R_{k-1,j-1}}{4^{j-1}-1},\quad k=\overline{2,n},\quad j=\overline{2,i}.$$

Calculele se pot aranja tabelar, astfel:

$$\begin{array}{ccccc} R_{1,1} & & & & \\ R_{2,1} & R_{2,2} & & & \\ R_{3,1} & R_{3,2} & R_{3,3} & & \\ \vdots & \vdots & \vdots & \ddots & \\ R_{n,1} & R_{n,2} & R_{n,3} & \dots & R_{n,n} \end{array}$$

Deoarece $(R_{n,1})$ este convergent şi $(R_{n,n})$ este convergent, mai rapid decât $(R_{n,1})$. Drept criteriu de oprire se poate folosi $|R_{n-1,n-1}-R_{n,n}|\le\varepsilon$.

## 5.6. Cuadraturi adaptive II

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 combinaţie a două aproximante de tip Simpson:

$$S_{k,2}=S_{k,1}+\frac{S_{k,1}-S_{k-1,1}}{15}=R_{k,2}+\frac{R_{k,2}-R_{k-1,2}}{15}.$$

[10] Jacob Bernoulli (1654-1705), fratele mai mare al lui Johann Bernoulli, activ în Basel. A fost unul dintre primii matematicieni care a apreciat importanţa introducerii de către Leibniz şi Newton a calculului diferenţial şi integral, pe care l-a îmbogăţit cu contribuţii originale, în competiţie (nu întotdeauna amicală) cu fratele său mai mic. Este unul dintre fondatorii Teoriei probabilităţilor, prin lucrarea sa *Ars conjectandi* şi prin „legea numerelor mari.

Relaţia

$$S_{k,2} = S_{k,1} + \frac{S_{k,1} - S_{k-1,1}}{15}, \tag{5.6.1}$$

va fi folosită la elaborarea unui algoritm de quadratură adaptivă. Fie $c = (a + b)/2$. Formula elementară a lui Simpson este

$$S = \frac{h}{6} \left( f(a) + 4f(c) + f(b) \right).$$

Pentru două subintervale se obţine

$$S_2 = \frac{h}{12} \left( f(a) + 4f(d) + 2f(c) + 4f(e) + f(b) \right),$$

unde $d = (a + c)/2$ şi $e = (c + b)/2$. Cantitatea $Q$ se obţine aplicând (5.6.1) celor două aproximante:

$$Q = S_2 + (S_2 - S)/15.$$

Putem să dam acum un algoritm recursiv pentru aproximarea integralei. Funcţia $adquad$ evaluează integrandul aplicând regula lui Simpson. Ea apelează recursiv $quadstep$ şi aplică extrapolarea. Descrierea se dă în algoritmul 5.2.

**Algoritmul 5.2** Cuadratură adaptivă bazată pe metoda lui Simpson şi extrapolare

**Intrare:** funcţia $f$, intervalul $[a, b]$, eroarea $\varepsilon$
**Ieşire:** Valoarea aproximativă a integralei

**function** $adquad(f, a, b, \varepsilon) : real$
$c := (a + b)/2$;
$fa = f(a)$; $fb := f(b)$; $fc := f(c)$;
$Q := quadstep(f, a, b, \varepsilon, fa, fc, fb)$;
**return** $Q$;
**function** $quadstep(f, a, b, \varepsilon, fa, fc, fb) : real$
$h := b - a$; $c := (a + b)/2$;
$fd := f((a + c)/2)$; $fe := f((c + b)/2)$;
$Q1 := h/6 * (fa + 4 * fc + fb)$;
$Q2 := h/12 * (fa + 4 * fb + 2 * fc + 4 * fe + fb)$;
**if** $|Q1 - Q2| < \varepsilon$ **then**
$Q := Q2 + (Q2 - Q1)/15$;
**else**
$Qa := quadstep(f, a, c, \varepsilon, fa, fd, fc)$;
$Qb := quadstep(f, c, b, \varepsilon, fc, fe, fb)$;
$Q := Qa + Qb$;
**end if**
**return** $Q$;

CAPITOLUL 6

# Rezolvarea numerică a ecuaţiilor neliniare

## Cuprins



## 6.1. Ecuaţii neliniare

Problema discutată în acest capitol se poate scrie generic sub forma

$$f(x) = 0, \tag{6.1.1}$$

dar admite diverse interpretări, depinzând de semnificaţia lui $x$ şi $f$. Cel mai simplu caz este cel al unei singure ecuaţii cu o singură necunoscută, caz în care $f$ este o funcţie dată de o variabilă reală sau complexă şi încercăm să găsim valorile acestei variabile pentru care $f$ se anulează. Astfel de valori se numesc *rădăcini* ale ecuaţiei (6.1.1) sau *zerouri* ale funcţiei $f$.

Dacă $x$ din (6.1.1) este un vector, să zicem $x = [x_1, x_2, \ldots, x_d]^T \in \mathbb{R}^d$ şi $f$ este de asemenea un vector ale cărui componente sunt funcţii de cele $d$ variabile $x_1, x_2, \ldots, x_d$, atunci (6.1.1) reprezintă un sistem de ecuaţii.

Se spune că sistemul este neliniar dacă cel puţin una dintre componentele lui $f$ depinde neliniar de cel puţin una din variabilele $x_1, x_2, \ldots, x_d$. Dacă toate componentele lui $f$ sunt funcţii liniare de $x_1, \ldots, x_d$ avem de-a face cu un sistem de ecuaţii algebrice liniare. Mai general (6.1.1) ar putea reprezenta o ecuaţie funcţională, dacă $x$ este un element al unui spaţiu de funcţii şi $f$ este un operator (liniar sau neliniar) ce acţionează pe acest spaţiu. În fiecare din aceste situaţii zeroul din dreapta lui (6.1.1) poate avea diverse interpretări: numărul zero în primul caz, vectorul nul în al doilea şi funcţia identic nulă în cel de-al treilea.

Mare parte din acest capitol este consacrată unei ecuaţii neliniare scalare. Astfel de ecuaţii apar frecvent în analiza sistemelor în vibraţie, unde rădăcinile corespund frecvenţelor critice (rezonanţă). Cazul special al ecuaţiilor algebrice, unde $f$ din (6.1.1) este un polinom, este de importanţă considerabilă şi merită un tratament special.

## 6.2. Iteraţii, convergenţă şi eficienţă

Nici chiar cele mai simple ecuaţii - de exemplu cele algebrice - nu admit soluţii care să fie exprimabile prin expresii raţionale sau radicali. Din acest motiv este imposibil, în general, să calculăm rădăcinile ecuaţiilor neliniare printr-un număr finit de operaţii aritmetice. De aceea este nevoie de o metodă iterativă, adică de o procedură care generează o secvenţă infinită de aproximaţii $\{x_n\}_{n\in\mathbb{N}}$ astfel încât

$$\lim_{n\to\infty} x_n = \alpha, \tag{6.2.1}$$

unde $\alpha$ este o rădăcină a ecuaţiei. În cazul unui sistem $x_k$ şi $\alpha$ sunt vectori de dimensiune adecvată, iar convergenţa trebuie înţeleasă în sensul convergenţei pe componente.

Deşi convergenţa unui proces iterativ este de dorit, pentru a putea fi practică, este necesar ceva mai mult decât convergenţa. Ceea ce se doreşte este o convergenţă rapidă. Conceptul de bază pentru măsurarea vitezei de convergenţă este ordinul de convergenţă.

**Definiţia 6.2.1** *Spunem că* $x_n$ converge *către* $\alpha$ *(cel puţin)* liniar *dacă*

$$|x_n - \alpha| \leq e_n \tag{6.2.2}$$

*unde* $\{e_n\}$ *este un şir pozitiv ce satisface*

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = c, \quad 0 < c < 1. \tag{6.2.3}$$

*Dacă (6.2.2) şi (6.2.3) au loc cu egalitate în (6.2.2) atunci* $c$ *se numeşte* eroare asimptotică .

În această definiţie, expresia ,,cel puţin“ se leagă de faptul că avem doar inegalitate în (6.2.2), ceea ce dorim în practică. De fapt, strict vorbind, marginea $e_n$ converge liniar, însemnând că în cele din urmă (pentru $n$ suficient de mare) fiecare din aceste margini ale erorii este aproximativ o fracţie constantă din precedenta.

**Definiţia 6.2.2** *Se spune că* $x_n$ converge *către* $\alpha$ *(cel puţin) cu* ordinul $p \geq 1$ *dacă (6.2.2) are loc cu*

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^p} = c, \quad c > 0 \tag{6.2.4}$$

Astfel convergenţa de ordinul 1 coincide cu convergenţa liniară, în timp ce convergenţa de ordinul $p > 1$ este mai rapidă. De notat că în acest ultim caz nu se pune nici o restricţie asupra constantei $c$: odată ce $e_n$ este suficient de mic, exponentul $p$ va avea grijă de convergenţă. Şi în acest caz, dacă avem egalitate în (6.2.2), $c$ se numeşte eroare asimptotică.

Aceleaşi definiţii se aplică şi şirurilor vectoriale, cu modulul înlocuit cu orice normă vectorială.

Clasificarea convergenţei în raport cu ordinul este destul de rudimentară, deoarece sunt tipuri de convergenţă la care definiţiile (6.2.1) şi (6.2.2) nu se aplică. Astfel, un şir $\{e_n\}$ poate converge către zero mai încet decât liniar, de exemplu dacă $c = 1$ în (6.2.3). Acest tip de convergenţă se numeşte subliniară. La fel, $c = 0$ în (6.2.3) conduce la convergenţă superliniară, dacă (6.2.4) nu are loc pentru nici un $p > 1$.

Este instructiv să examinăm comportarea lui $e_n$, dacă în loc de relaţia la limită avem egalitate pentru un anumit $n$, să zicem

$$\frac{e_{n+1}}{e_n^p} = c, \quad n = n_0, n_0 + 1, n_0 + 2, \ldots \tag{6.2.5}$$

Pentru $n_0$ suficient de mare, relaţia este aproape adevărată. Printr-o simplă inducţie se obţine că

$$e_{n_0+k} = c^{\frac{p^k-1}{p-1}} e_{n_0}^{p^k}, \quad k = 0, 1, 2, \ldots, \tag{6.2.6}$$

care desigur are loc pentru $p > 1$, dar şi pentru $p = 1$ când $p \downarrow 1$:

$$e_{n_0+k} = c^k e_{n_0}, \quad k = 0, 1, 2, \ldots, \quad (p = 1) \tag{6.2.7}$$

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 (6.2.2) reprezintă numărul suplimentar de cifre zecimale corecte din aproximaţia $x_{n_0+k}$ (în contrast cu $x_{n_0}$). Logaritmând (6.2.6) şi (6.2.7) obţinem

$$\delta_k = \begin{cases} k \log \frac{1}{c}, & \text{dacă} \quad p = 1 \\ p^k \left[ \frac{1-p^{-k}}{p-1} \log \frac{1}{c} + (1 - p^{-k}) \log \frac{1}{e_{n_0}} \right], & \text{dacă} \quad p > 1 \end{cases}$$

deci când $k \to \infty$

$$\delta_k \sim c_1 k \quad (p = 1), \quad \delta_k \sim c_p p^k \quad (p > 1), \tag{6.2.8}$$

unde $c_1 = \log \frac{1}{c} > 0$, dacă $p = 1$ şi

$$c_p = \frac{1}{p-1} \log \frac{1}{c} + \log \frac{1}{e_{n_0}}$$

(presupunem că $n_0$ este suficient de mare şi deci $e_{n_0}$ suficient de mic, pentru a avea $c_p > 0$). Aceasta ne arată că numărul de cifre zecimale corecte creşte liniar odată cu $k$ când $p = 1$ şi exponenţial când $p > 1$. În ultimul caz, $\delta_{k+1}/\delta_k \sim p$, însemnând că (pentru $k$ mare) numărul de cifre zecimale corecte, creşte pe iteraţie, cu un factor $p$.

Dacă fiecare iteraţie necesită $m$ unităţi de lucru (o „unitate de lucru" este efortul necesar pentru a calcula o valoare a funcţiei sau a unei anumite derivate a sa), atunci *indicele de eficienţă* al iteraţiei poate fi definit prin

$$\lim_{k \to \infty} [\delta_{k+1}/\delta_k]^{1/m} = p^{1/m}.$$

Aceasta ne dă o bază comună de comparare între diversele metode iterative. Metodele liniare au indicele de eficienţă 1.

Calculele practice necesită o regulă de oprire care să termine iteraţia atunci când s-a obţinut (sau se crede că s-a obţinut) precizia dorită. Ideal, ne oprim atunci când $\|x_n - \alpha\| < tol$, $tol$ dat. Deoarece $\alpha$ nu este cunoscut se obişnuieşte să se înlocuiască $x_n - \alpha$ cu $x_n - x_{n-1}$ şi se impune cerinţa ca

$$\|x_n - x_{n-1}\| \leq tol \tag{6.2.9}$$

unde

$$tol = \|x_n\| \varepsilon_r + \varepsilon_a \tag{6.2.10}$$

cu $\varepsilon_r, \varepsilon_a$ valori date ale erorii. Ca o măsură de siguranţă, am putea cere ca (6.2.9) să aibă loc pentru mai multe valori consecutive ale lui $n$, nu doar pentru una singură. Alegând $\varepsilon_r = 0$ sau $\varepsilon_a = 0$ se obţine un test de eroare absolută sau relativă. Este totuşi prudent să utilizăm un test mixt, cum ar fi, să zicem $\varepsilon_e = \varepsilon_a = \varepsilon$. 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 funcţie $crit\_oprire$ care implementează testul de oprire.

## 6.3. Metoda şirurilor Sturm

Există situaţii în care este de dorit să selectăm o rădăcină particulară dintre mai multe şi să avem scheme iterative care converg către ea. Acesta este cazul, de exemplu, pentru polinoame ortogonale, ale căror rădăcini sunt reale şi distincte. De asemenea am putea dori să facem o selecţie a unei anumite rădăcini: cea mai mare, a doua ca mărime, a treia ş.a.m.d. şi să o calculăm fără a mai calcula şi altele. Aceasta este posibil dacă combinăm înjumătăţirea cu teorema lui Sturm [1].

Să considerăm ecuaţia

$$f(x) := \pi_d(x) = 0, \tag{6.3.1}$$

unde $\pi_d$ este un polinom de grad $d$, ortonormal în raport cu o anumită măsură. Ştim că $\pi_d$ este polinomul caracteristic al unei matrice simetrice tridiagonale şi poate fi calculat recursiv printr-o relaţie de recurenţă de forma

$$\begin{aligned} &\pi_0(x) = 1, \qquad \pi_1(x) = x - \alpha_0 \\ &\pi_{k+1}(x) = (x - \alpha_k)\pi_k(x) - \beta_k \pi_{k-1}(x), \quad k = 1, 2, \ldots, d-1 \end{aligned} \tag{6.3.2}$$

cu $\beta_k$ pozitiv. Recurenţa (6.3.2) este utilă nu numai pentru calculul lui $\pi_d(x)$, dar şi pentru că are următoarea proprietate utilă, datorată lui Sturm.

**Propoziţia 6.3.1** *Fie $\sigma(x)$ numărul de schimbări de semn (zerourile nu contează) în secvenţa de numere*

$$\pi_d(x), \pi_{d-1}(x), \ldots, \pi_1(x), \pi_0(x). \tag{6.3.3}$$

*Atunci, pentru orice două numere $a, b$ cu $a < b$, numărul de zerouri a lui $\pi_d$ pe intervalul $a < x \leq b$ este egal cu $\sigma(a) - \sigma(b)$.*

Deoarece $\pi_k(x) = x^k + \ldots$, este evident că $\sigma(-\infty) = d$, $\sigma(+\infty) = 0$, astfel încât numărul de rădăcini reale ale lui $\pi_d$ este $\sigma(-\infty) - \sigma(\infty) = d$. Mai mult dacă $\xi_1 > \xi_2 > \cdots > \xi_d$ desemnează zerourile lui $\pi_d$ în ordine descrescătoare, avem comportarea lui $\sigma$ ca în figura 6.1.

---

[1] 

Jaques Charles François Sturm (1803-1855), matematician şi fizician elveţian, cunoscut pentru teorema sa asupra şirurilor Sturm, descoperită în 1829 şi pentru teoria sa asupra ecuaţiei diferenţiale Sturm-Liouville. A avut contribuţii semnificative şi în domeniul geometriei proiective şi diferenţiale.

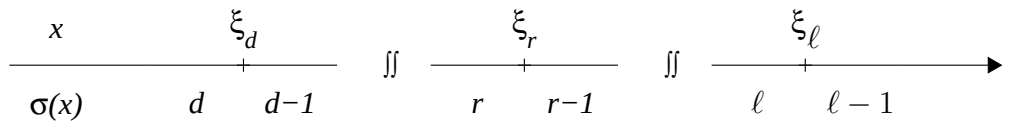



Figura 6.1: Ilustrarea metodei lui Sturm

Este uşor de văzut că:

$$\sigma(x) \leq r-1 \Longleftrightarrow x \geq \xi_r. \tag{6.3.4}$$

Într-adevăr, presupunem că $x \geq \xi_r$. Atunci $\{\#\text{ zerouri } \leq x\} \geq d+1-r$. Deci conform teoremei lui Sturm, $\sigma(-\infty)-\sigma(x) = d-\sigma(x) = \{\#\text{ zerouri } \leq x\} \leq d_1 - r$, adică $\sigma(x) \leq r-1$. Reciproc, dacă $\sigma(x) \leq r-1$, atunci, din teorema lui Sturm, $\{\#\text{ zerouri } \leq x\} = d-\sigma(x) \geq d+1-r$, ceea ce implică $x \geq \xi_r$ (vezi figura 6.1).

Ideea de bază este de a controla procesul de înjumătăţire nu ca mai sus, ci mai degrabă verificând inegalitatea (6.3.4) pentru a vedea dacă suntem în stânga sau în dreapta lui $\xi_r$. Pentru a iniţializa procedura, avem nevoie de valorile $a_1 = a$, $b_1 = b$ astfel încât $a < \xi_d$ şi $b > \xi_1$. Acestea se obţin trivial ca şi capete ale intervalului de ortogonalitate al lui $\pi_d$, dacă acesta este finit. Mai general putem aplica teorema lui Gershgorin matricei lui Jacobi $J_d$ asociate polinomului (6.3.2)

$$J_n = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & 0 \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_2 & \ddots & \\ & & \ddots & \ddots & \sqrt{\beta_{n-1}} \\ 0 & & & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix}$$

ţinând cont că zerourile lui $\pi_d$ sunt valori proprii ale lui $J_d$.

Teorema lui Gershgorin afirmă că valorile proprii ale matricei $A = [a_{ij}]$ de ordin $d$ sunt localizate în reuniunea discurilor

$$\left\{ z \in \mathbb{C} : \ |z - a_{ii}| \leq r_i, \ r_i = \sum_{j \neq i} |a_{ij}| \right\}, \quad i = \overline{1,d}.$$

În acest mod, $a$ poate fi ales ca fiind cel mai mic şi $b$ cel mai mare dintre cele $d$ numere $\alpha_0 + \sqrt{\beta_1}$, $\alpha_1 + \sqrt{\beta_1} + \sqrt{\beta_2}, \ldots, \alpha_{d-2} + \sqrt{\beta_{d-2}} + \sqrt{\beta_{d-1}}$, $\alpha_{d-1} + \sqrt{\beta_{d-1}}$. Metoda şirurilor lui Sturm continuă după cum urmează, pentru orice $r$ cu $1 \leq r \leq d$:

**for** $n := 1, 2, 3, \ldots$ **do**

$x_n := \frac{1}{2}(a_n + b_n)$;
**if** $\sigma(x_n) > r - 1$ **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**

Deoarece iniţial $\sigma(a) = d > r - 1$, $\sigma(b) = 0 \leq r - 1$, rezultă din construcţie că

$$\sigma(a_n) > r - 1, \quad \sigma(b_n) \leq r - 1, \quad n = 1, 2, 3, \ldots$$

însemnând că $\xi_r \in [a_n, b_n]$, pentru orice $n = 1, 2, 3, \ldots$. Mai mult, deoarece în metoda înjumătăţirii, $b_n - a_n = \dfrac{b-a}{2^{n-1}}$, metoda converge cel puţin liniar către rădăcina $\xi_r$.

## 6.4. Metoda falsei poziţii

Ca în metoda înjumătăţirii, presupunem că avem două numere $a < b$ astfel încât

$$f \in C[a, b], \quad f(a)f(b) < 0 \tag{6.4.1}$$

şi generăm un şir descendent de intervale $[a_n, b_n]$, $n = 1, 2, 3, \ldots$ cu $a_1 = a$, $b_1 = b$ astfel încât $f(a_n)f(b_n) < 0$. Spre deosebire de metoda înjumătăţirii, pentru a determina următorul interval nu luăm mijlocul lui $[a_n, b_n]$, ci soluţia $x = x_n$ a ecuaţiei liniare

$$(L_1 f)(x; a_n, b_n) = 0.$$

Aceasta pare să fie o alegere mai flexibilă decât în metoda înjumătăţirii deoarece $x_n$ va fi mai apropiat de capătul de care $|f|$ este mai mic.

Procedura decurge după cum urmează:

**for** $n := 1, 2, \ldots$ **do**
$x_n := a_n - \dfrac{a_n - b_n}{f(a_n) - f(b_n)} f(a_n)$;
**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**

Iteraţia se poate termina când $\min(x_n - a_n, b_n - x_n) \leq tol$, unde $tol$ este o valoare dată.

Convergenţa se analizează mai uşor dacă presupunem că $f$ este convexă sau concavă pe $[a, b]$. Dacă $f$ este convexă, avem

$$f''(x) > 0, \quad x \in [a, b], \quad f(a) < 0, \quad f(b) > 0. \tag{6.4.2}$$

Şirul

$$x_{n+1} = x_n - \frac{x_n - b}{f(x_n) - f(b)} f(x_n), \quad n \in \mathbb{N}^*, \quad x_1 = a \tag{6.4.3}$$

este monoton crescător şi mărginit superior de $\alpha$, deci convergent către o limită $x$, iar $f(x) = 0$ (vezi figura 6.2).

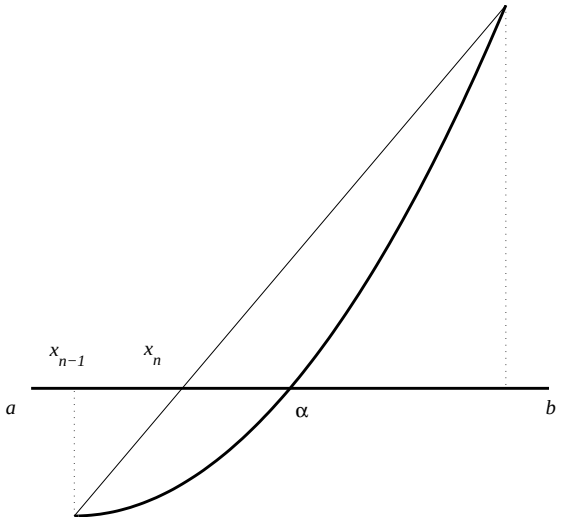



Figura 6.2: Metoda falsei poziţii

Viteza de convergenţă se determină scăzând $\alpha$ din ambii membri ai lui (6.4.3) şi utilizând faptul că $f(\alpha) = 0$:

$$x_{n+1} - \alpha = x_n - \alpha - \frac{x_n - b}{f(x_n) - f(b)} [f(x_n) - f(\alpha)].$$

Împărţind 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 $n \to \infty$ şi utilizând faptul că $x_n \to \alpha$, obţinem

$$\lim_{n \to \infty} \frac{x_{n+1} - \alpha}{x_n - \alpha} = 1 - (b - \alpha) \frac{f'(\alpha)}{f(b)}. \tag{6.4.4}$$

Deci metoda converge liniar, cu eroarea asimptotică

$$c = 1 - (b-a)\frac{f'(\alpha)}{f(b)}.$$

Datorită ipotezei convexităţii avem $c \in (0,1)$. Analog se face demonstraţia în cazul când $f$ este concavă. Dacă $f$ nu este nici convexă nici concavă pe $[a,b]$, ci $f \in C^2[a,b]$ şi $f''(\alpha) \neq 0$, $f''$ are semn constant pe o vecinătate a lui $\alpha$ şi pentru un $n$ suficient de mare $x_n$ ajunge în acea vecinătate şi se poate proceda ca mai sus.

Dezavantaje. (i) Convergenţa lentă; (ii) Faptul că unul din capete poate rămâne fix. Dacă $f$ este turtită în vecinătatea rădăcinii şi $a$ este apropiat de $\alpha$ şi $b$ depărtat convergenţa poate fi foarte lentă.

## 6.5. Metoda secantei

Este o variantă a metodei falsei poziţii, în care nu se mai cere ca $f$ să aibă valori de semne contrare, nici măcar la capetele intervalului iniţial.

Se aleg două valori arbitrare de pornire $x_0, x_1$ şi se continuă cu

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n), \quad n \in \mathbb{N}^*. \tag{6.5.1}$$

Aceasta preîntâmpină apariţia unei false poziţii şi sugerează o convergenţă mai rapidă. Din păcate, nu mai are loc convergenţa „globală“ pe $[a,b]$ ci doar convergenţa „locală“, adică numai dacă $x_0$ şi $x_1$ sunt suficient de apropiate de rădăcină.

Vom avea în continuare nevoie de o relaţie între trei erori consecutive

$$\begin{aligned} x_{n+1} - \alpha &= x_n - \alpha - \frac{f(x_n)}{f[x_{n-1}, x_n]} = (x_n - \alpha)\left(1 - \frac{f(x_n) - f(\alpha)}{(x_n - \alpha) f[x_{n-1}, x_n]}\right) \\ &= (x_n - \alpha)\left(1 - \frac{f[x_n, \alpha]}{f[x_{n-1}, x_n]}\right) = (x_n - \alpha)\frac{f[x_{n-1}, x_n] - f[x_n, \alpha]}{f[x_{n-1}, x_n]} \\ &= (x_n - \alpha)(x_{n-1} - \alpha)\frac{f[x_n, x_{n-1}, \alpha]}{f[x_{n-1}, x_n]}. \end{aligned}$$

Deci

$$(x_{n+1} - \alpha) = (x_n - \alpha)(x_{n-1} - \alpha)\frac{f[x_n, x_{n-1}, \alpha]}{f[x_{n-1}, x_n]}, \quad n \in \mathbb{N}^* \tag{6.5.2}$$

Din (6.5.2) rezultă imediat că dacă $\alpha$ este o rădăcină simplă ($f(\alpha) = 0$, $f'(\alpha) \neq 0$) şi $x_n \to \alpha$ şi dacă $f \in C^2$ pe o vecinătate a lui $\alpha$, convergenţa este superliniară. Cât este ordinul de convergenţă?

Înlocuim raportul diferenţelor divizate din (6.5.2) cu o constantă, ceea ce este aproape adevărat când $n$ este mare. Punând $c_k = |x_k - \alpha|$, avem

$$e_{n+1} = e_n e_{n-1} C, \quad C > 0$$

Înmulţind ambii membri cu $C$ şi punând $E_n = Ce_n$ obţinem

$$E_{n+1} = E_n E_{n-1}, \quad E_n \to 0.$$

Logaritmând şi punând $y_n = \frac{1}{E_n}$ obţinem

$$y_{n+1} = y_n + y_{n-1}, \tag{6.5.3}$$

care este recurenţa pentru şirul lui Fibonacci. Soluţia este

$$y_1 = c_1 t_1^n + c_2 t_2^n,$$

$c_1, c_2$ constante şi

$$t_1 = \frac{1}{2}(1 + \sqrt{5}), \qquad t_2 = \frac{1}{2}(1 - \sqrt{5}).$$

Deoarece $y_n \to \infty$, avem $c_1 \neq 0$ şi $y_n \sim c_1 t_1^n$, căci $|t_2| < 1$. Revenind la substituţie $\frac{1}{E_n} \sim e^{c_1 t_1^n}$, $\frac{1}{e_n} \sim C e^{c_1 t_1^n}$ şi deci

$$\frac{e_{n+1}}{e_n^{t_1}} \sim \frac{C^{t_1} e^{c_1 t_1^n t_1}}{C e^{c_1 t_1^{n+1}}} = C^{t_1 - 1}, \quad n \to \infty.$$

Ordinul de convergenţă este $t_1 = \dfrac{1 + \sqrt{5}}{2} \approx 1.61803\ldots$ (secţiunea de aur).

**Teorema 6.5.1** *Fie $\alpha$ un zero simplu al lui $f$ şi fie $I_\varepsilon = \{x \in \mathbb{R} : \ |x - \alpha| < \varepsilon\}$ şi presupunem că $f \in C^2[I_\varepsilon]$. Definim pentru $\varepsilon$ suficient de mic*

$$M(\varepsilon) = \max_{\substack{s \in I_\varepsilon \\ t \in I_\varepsilon}} \left| \frac{f''(s)}{2f'(t)} \right|. \tag{6.5.4}$$

*Presupunem că*

$$\varepsilon M(\varepsilon) < 1 \tag{6.5.5}$$

*Atunci metoda secantei converge către rădăcina unică $\alpha \in I_\varepsilon$ pentru orice valori de pornire $x_0 \neq x_1$ cu $x_0 \in I_\varepsilon$, $x_1 \in I_\varepsilon$.*

**Observaţia 6.5.2.** Se observă că $\lim\limits_{\varepsilon \to 0} M(\varepsilon) = \left| \dfrac{f''(\alpha)}{2f'(\alpha)} \right| < \infty$, deci (6.5.5) poate fi satisfăcută pentru $\varepsilon$ suficient de mic. Natura locală a convergenţei este cuantificată prin cerinţa ca $x_0, x_1 \in I_\varepsilon$. $\diamondsuit$

*Demonstraţie.* 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 $f(\alpha) = 0$ şi $\xi \in (x, \alpha)$ (sau $(\alpha, x)$). Astfel dacă $x \in I_\varepsilon$, atunci şi $\xi \in I_\varepsilon$ şi avem

$$f(x) = (x-\alpha)f'(\alpha)\left[1 + \frac{x-\alpha}{2}\frac{f''(\xi)}{f'(\alpha)}\right]$$

Aici, dacă $x \neq \alpha$, toţi trei factorii sunt diferiţi 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 $x = \alpha$. Să arătăm că $x_n \in I_\varepsilon$ pentru orice $n$, în afară de cazul când $f(x_n) = 0$, în care $x_n = \alpha$ şi metoda converge într-un număr finit de paşi. Vom demonstra aceasta prin inducţie: presupunem că $x_{n-1}, x_n \in I_\varepsilon$ şi $x_n \neq x_{n-1}$. 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 (6.5.2) rezultă

$$|x_{n+1} - \alpha| \leq \varepsilon^2 \left|\frac{f''(\xi_n)}{2f'(\xi_1)}\right| \leq \varepsilon\varepsilon M(\varepsilon) < \varepsilon,$$

adică $x_{n+1} \in I_\varepsilon$. Mai mult, din relaţia între trei erori consecutive, (6.5.2), rezultă $x_{n+1} \neq x_n$ în afară de cazul când $f(x_n) = 0$ (şi atunci $x_n = \alpha$). Utilizând (6.5.2) 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 şi $x_n \to \alpha$ când $n \to \infty$. □

Deoarece este nevoie de o singură evaluare a lui $f$ pe pas, indicele de eficienţă este $p = \frac{1+\sqrt{5}}{2} \approx 1.61803\ldots$. Pseudocodul metodei este dat în algoritmul 6.1.

**Algoritmul 6.1** Metoda secantei pentru ecuaţii neliniare în $\mathbb{R}$

**Intrare:** Funcţia $f$, valorile de pornire $x_0$ şi $x_1$, numarul maxim de iteraţii, $Nmax$, informaţii de toleranţă $tol$

**Ieşire:** O aproximaţie a rădăcinii sau un mesaj de eroare

1: $xc := x_1; \qquad xv = x_0;$
2: $fc := f(x_1); \qquad fv := f(x_0);$
3: **for** $k := 1$ **to** $Nmax$ **do**
4: $\quad xn := xc - fc * (xc - xv)/(fc - fv);$
5: $\quad$ **if** $crit\_oprire(tol)$ **then**
6: $\quad\quad$ **return** $xn$;{Succes}
7: $\quad$ **end if**
8: $\quad xv := xc; \quad fv := fc; \quad xc := xn; \quad fc = f(xn);$
9: **end for**
10: $error$("S-a depăşit numărul de iteraţii").

## 6.6. Metoda lui Newton

Poate fi privită ca un caz la limită al metodei secantei, când $x_{n-1} \to x_n$. Obţinem iteraţia

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{6.6.1}$$

O altă interpretare mult mai fructuoasă este liniarizarea ecuaţiei $f(x) = 0$ în $x = x_n$:

$$f(x) \approx f(x_n) + (x - x_n)f'(x_n) = 0.$$

Privită în acest mod metoda lui Newton se poate generaliza la ecuaţii neliniare de toate tipurile (sisteme neliniare, ecuaţii funcţionale, caz în care $f'$ trebuie înţeleasă ca derivată Fréchet), iar iteraţia este

$$x_{n+1} = x_n - [f'(x_n)]^{-1} f(x_n) \tag{6.6.2}$$

Studiul erorii în metoda lui Newton este la fel ca cel al erorii în metoda secantei.

$$\begin{aligned} x_{n+1} - \alpha &= x_n - \alpha - \frac{f(x_n)}{f'(x_n)} \\ &= (x_n - \alpha)\left[1 - \frac{f(x_n) - f(\alpha)}{(x_n - \alpha)f'(x_n)}\right] \\ &= (x_n - \alpha)\left(1 - \frac{f[x_n, \alpha]}{f[x_n, x_n]}\right) = (x_n - \alpha)^2 \frac{f[x_n, x_n, \alpha]}{f[x_n, x_n]} \end{aligned} \tag{6.6.3}$$

De aceea, dacă $x_n \to \alpha$, atunci

$$\lim_{n\to\infty} \frac{x_{n+1} - \alpha}{(x_n - \alpha)^2} = \frac{f''(\alpha)}{2f'(\alpha)}$$

şi ordinul de convergenţă al metodei lui Newton este 2 dacă $f''(\alpha) \neq 0$. Referitor la convergenţa locală a metodei lui Newton avem

**Teorema 6.6.1** *Fie $\alpha$ o rădăcină simplă a ecuaţiei $f(x) = 0$ şi $I_\varepsilon = \{x \in \mathbb{R} : |x-\alpha| \leq \varepsilon\}$. Presupunem că $f \in C^2[I_\varepsilon]$. Definim*

$$M(\varepsilon) = \max_{\substack{s\in I_\varepsilon \\ t\in I_\varepsilon}} \left|\frac{f''(s)}{2f'(t)}\right| \tag{6.6.4}$$

*Dacă $\varepsilon$ este suficient de mic astfel încât*

$$2\varepsilon M(\varepsilon) < 1, \tag{6.6.5}$$

*atunci pentru orice $x_0 \in I_\varepsilon$, metoda lui Newton este bine definită şi converge pătratic către singura rădăcină $\alpha \in I_\varepsilon$.*

Criteriul de oprire pentru metoda lui Newton

$$|x_n - x_{n-1}| < \varepsilon$$

se bazează pe următoarea propoziţie:

**Propoziţia 6.6.2** *Fie $(x_n)$ şirul de aproximante generat prin metoda lui Newton. Dacă $\alpha$ este o rădăcină simplă din $[a, b]$, $f \in C^2[a, b]$ şi metoda este convergentă, atunci există un $n_0 \in \mathbb{N}$ astfel încât*

$$|x_n - \alpha| \leq |x_n - x_{n-1}|, \qquad n > n_0.$$

*Demonstraţie.* Vom arăta întâi că

$$|x_n - \alpha| \leq \frac{1}{m_1}|f(x_n)|, \qquad m_1 \leq \inf_{x\in[a,b]} |f'(x)|. \tag{6.6.6}$$

Utilizând teorema lui Lagrange, $f(\alpha) - f(x_n) = f'(\xi)(\alpha - x_n)$, cu $\xi \in (\alpha, x_n)$ (sau $(x_n, \alpha)$). Din relaţiile $f(\alpha) = 0$ şi $|f'(x)| \geq m_1$ pentru $x \in (a, b)$ rezultă că $|f(x_n)| \geq m_1|\alpha - x_n|$, adică chiar (6.6.6).

Pe baza formulei lui Taylor avem

$$f(x_n) = f(x_{n-1}) + (x_n - x_{n-1})f'(x_{n-1}) + \frac{1}{2}(x_n - x_{n-1})^2 f''(\mu), \tag{6.6.7}$$

cu $\mu \in (x_{n-1}, x_n)$ sau $\mu \in (x_n, x_{n-1})$. Ţinând cont de modul de obţinere a unei aproximaţii de metoda lui Newton, avem $f(x_{n-1}) + (x_n - x_{n-1})f'(x_{n-1}) = 0$ şi din (6.6.7) se obţine

$$|f(x_n)| = \frac{1}{2}(x_n - x_{n-1})^2|f''(\mu)| \leq \frac{1}{2}(x_n - x_{n-1})^2\|f''\|_\infty,$$

iar pe baza formulei (6.6.6) 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

$$\frac{\|f''\|_\infty}{2m_1}(x_n - x_{n-1}) < 1, \qquad n > n_0$$

şi deci

$$|x_n - \alpha| \leq |x_n - x_{n-1}|, \qquad n > n_0.$$

□

Interpretarea geometrică a metodei lui Newton apare în figura 6.3, iar descrierea metodei în algoritmul 6.2.

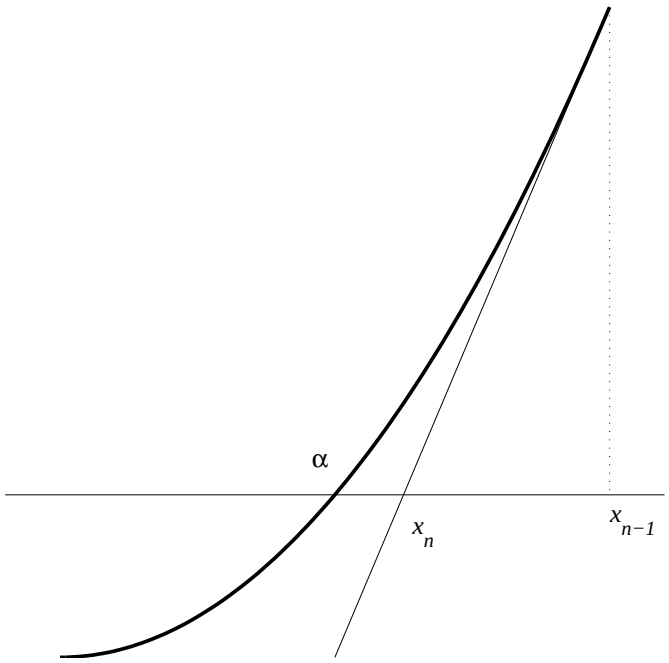



Figura 6.3: Metoda lui Newton

Alegerea valorii de pornire este, în general, o problemă dificilă. În practică, se alege o valoare, iar dacă după un număr maxim fixat de iteraţii nu s-a obţinut precizia dorită, testată prin unul din criteriile uzuale, se încearcă cu altă valoare de pornire. De exemplu, dacă rădăcina este izolată într-un interval $[a, b]$ şi $f''(x) \neq 0$, $x \in (a, b)$, un criteriu de alegere este $f(x_0)f''(x_0) > 0$.

**Algoritmul 6.2** Metoda lui Newton pentru ecuaţii neliniare în $\mathbb{R}$

**Intrare:** Funcţia $f$, derivata $f'$, valoarea de pornire $x_0$, numarul maxim de iteraţii, $Nmax$, informaţii de toleranţă $tol$

**Ieşire:** O aproximaţie 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** $x_{k+1}$;{Succes}
5: **end if**
6: **end for**
7: $error$("S-a depăşit numărul de iteraţii").

# 6.7. Metoda aproximaţiilor succesive

Adesea, în aplicaţii, ecuaţiile neliniare apar sub forma unei *probleme de punct fix*: să se determine $x$ astfel încât

$$x = \varphi(x). \tag{6.7.1}$$

Un număr $\alpha$ ce satisface această ecuaţie se numeşte punct fix al lui $\varphi$. Orice ecuaţie $f(x) = 0$ se poate scrie (în multe moduri diferite) în forma echivalentă (6.7.1). De exemplu, dacă $f'(x) \neq 0$ în intervalul de interes, putem lua

$$\varphi(x) = x - \frac{f(x)}{f'(x)}. \tag{6.7.2}$$

Dacă $x_0$ este o aproximaţie iniţială a unui punct fix $\alpha$ a lui (6.7.1) atunci metoda aproximaţiilor succesive generează un şir de aproximaţii

$$x_{n+1} = \varphi(x_n). \tag{6.7.3}$$

Dacă acest şir converge şi $\varphi$ este continuă, atunci şirul converge către un punct fix a lui $\varphi$. De notat că (6.7.3) este chiar metoda lui Newton dacă $\varphi$ este dată de (6.7.2). Astfel metoda lui Newton poate fi privită ca o iteraţie de tip punct fix, dar nu şi metoda secantei. Pentru o iteraţie de forma (6.7.3), presupunând că $x_n \to \alpha$ când $n \to \infty$, ordinul de convergenţă este uşor de determinat. Să presupunem că în punctul fix $\alpha$ avem

$$\varphi'(\alpha) = \varphi''(\alpha) = \cdots = \varphi^{(p-1)}(\alpha) = 0, \quad \varphi^p(\alpha) \neq 0 \tag{6.7.4}$$

Presupunem că $\varphi \in C^p$ pe o vecinătate a lui $\alpha$. Avem atunci, conform teoremei lui Taylor

$$\varphi(x_n) = \varphi(\alpha) + (x_n - \alpha)\varphi'(\alpha) + \cdots + \frac{(x_n - \alpha)^{p-1}}{(p-1)!}\varphi^{(p-1)}(\alpha)$$
$$+ \frac{(x_n - \alpha)^p}{p!}\varphi^{(p)}(\xi_n) = \varphi(\alpha) + \frac{(x_n - \alpha)^p}{p!}\varphi^{(p)}(\xi_n),$$

unde $\xi_n \in (\alpha, x_n)$ (sau $(x_n, \alpha)$). Deoarece $\varphi(x_n) = x_{n+1}$ şi $\varphi(\alpha) = \alpha$ obţinem

$$\frac{x_{n+1} - \alpha}{(x_n - \alpha)^p} = \frac{1}{p!}\varphi^{(p)}(\xi_n).$$

Când $x_n \to \alpha$, deoarece $\xi_n$ este între $x_n$ şi $\alpha$, deducem pe baza continuităţii că

$$\lim_{n\to\infty} \frac{x_{n+1} - \alpha}{(x_n - \alpha)^p} = \frac{1}{p!}\varphi^{(p)}(\alpha) \neq 0. \tag{6.7.5}$$

Aceasta ne arată că ordinul de convergenţă este exact $p$ şi eroarea asimptotică este

$$c = \frac{1}{p!}\varphi^{(p)}(\alpha). \tag{6.7.6}$$

Combinând aceasta cu condiţia uzuală de convergenţă locală se obţine:

**Teorema 6.7.1** *Fie $\alpha$ un punct fix al lui $\varphi$ şi $I_\varepsilon = \{x \in \mathbb{R} : \ |x - \alpha| \leq \varepsilon\}$. Presupunem că $\varphi \in C^p[I_\varepsilon]$ şi satisface (6.7.4). Dacă*

$$M(\varepsilon) := \max_{t\in I_\varepsilon} |\varphi'(t)| < 1 \tag{6.7.7}$$

*atunci iteraţia (6.7.3) converge către $\alpha$, $\forall\ x_0 \in I_\varepsilon$. Ordinul de convergenţă este $p$, iar eroarea asimptotică este dată de (6.7.6).*

## 6.8. Metoda lui Newton pentru rădăcini multiple

Dacă $\alpha$ este o rădăcină multiplă de ordinul $m$, atunci ordinul de convergenţă 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 $\varphi'(\alpha) = 1 - 1/m < 1$.

O modalitate de a evita aceasta este să rezolvăm ecuaţia

$$u(x) := \frac{f(x)}{f'(x)} = 0$$

care are aceleaşi rădăcini ca şi $f$, dar simple. Metoda lui Newton pentru problema modificată are forma

$$x_{k+1} = x_k - \frac{u(x_k)}{u'(x_k)} = \frac{f(x_k)f'(x_k)}{[f'(x_k)]^2 - f(x_k)f''(x_k)}. \tag{6.8.1}$$

Deoarece $\alpha$ este o rădăcină simplă a lui $u$, convergenţa lui (6.8.1) este pătratică. Singurul dezavantaj teoretic al lui (6.8.1) este derivata a doua necesară suplimentar şi complexitatea mai mare a calculului lui $x_{k+1}$ din $x_k$. În practică aceasta este o slăbiciune, deoarece numitorul lui (6.8.1) poate lua valori foarte mici în vecinătatea lui $\alpha$ când $x_k \to \alpha$.

Convergenţa pătratică a metodei lui Newton se poate realiza nu numai prin modificarea problemei ci şi prin modificarea metodei. În vecinătatea unei soluţii multiple de ordinul $m$, $\alpha$, avem

$$f(x) = (x-\alpha)^m \varphi(x) \approx (x-\alpha)^m \cdot c, \tag{6.8.2}$$

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

$$x_{k+1} := x_k - m\frac{f(x_k)}{f'(x_k)}, \qquad k = 0, 1, 2, \ldots \tag{6.8.3}$$

converge pătratic către rădăcina multiplă de ordinul $m$ când se întrebuinţează o valoare corectă a lui $m$ în (6.8.3). Eficienţa variantei (6.8.3) 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 (6.8.2) $\alpha$ prin $x_k$

$$\begin{aligned} f(x_{k-1}) &\approx (x_{k-1} - x_k)^m \cdot c \\ f(x_{k-2}) &\approx (x_{k-2} - x_k)^m \cdot c. \end{aligned}$$

În continuare se obţine $m$:

$$m \approx \frac{\log\left[f(x_{k-1})/f(x_{k-2})\right]}{\log\left[(x_{k-1} - x_k)/(x_{k-2} - x_k)\right]}.$$

Această valoare poate fi utilizată în (6.8.3).

## 6.9. Ecuaţii algebrice

Există multe metode special concepute pentru a rezolva ecuaţii algebrice. Aici vom descrie numai metoda lui Newton aplicată în acest context, concentrându-ne asupra unui mod eficient de a evalua simultan valoarea polinomului şi a primei derivate.

**Metoda lui Newton aplicată ecuaţiilor algebrice.** Considerăm o ecuaţie algebrică de grad $d$

$$f(x) = 0, \quad f(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_0, \tag{6.9.1}$$

în care coeficientul dominant se presupune (fără a restrânge generalitatea) a fi egal cu 1 şi unde putem presupune, fără a restrânge generalitatea că $a_0 \neq 0$. Pentru simplitate vom presupune că toţi coeficienţii sunt reali. Pentru a aplica metoda lui Newton ecuaţiei (6.9.1) este nevoie de a metodă bună de evaluare a polinomului şi derivatei.

Schema lui Horner este bună 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 $f(x_n)$ şi $f'(x_n)$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Se aplică apoi metoda lui Newton polinomului $\dfrac{f(x)}{x - \alpha}$. Pentru rădăcini complexe se începe cu $x_0$ complex şi toate calculele se fac în aritmetică complexă. Este posibil să se împartă cu factori pătratici şi să se folosească aritmetica reală – se ajunge astfel la metoda lui Bairstow. Folosind metoda aceasta de scădere a gradului erorile pot fi mari. O modalitate de îmbunătăţire este de a utiliza rădăcinile astfel calculate ca aproximaţii iniţiale şi a aplica metoda lui Newton polinomului original.

## 6.10. Metoda lui Newton în $\mathbb{R}^n$

Metoda lui Newton este uşor de generalizat la sisteme neliniare

$$F(x) = 0, \tag{6.10.1}$$

unde $F : \Omega \subset \mathbb{R}^n \to \mathbb{R}^n$, iar $x, F(x) \in \mathbb{R}^n$. Sistemul (6.10.1) se scrie pe componente

$$\begin{cases} F_1(x_1, \ldots, x_n) = 0 \\ \vdots \\ F_n(x_1, \ldots, x_n) = 0 \end{cases}$$

Fie $F'(x^{(k)})$ jacobianul lui $F$ în $x^{(k)}$:

$$J := F'(x^{(k)}) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1}(x^{(k)}) & \dots & \frac{\partial F_1}{\partial x_n}(x^{(k)}) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1}(x^{(k)}) & \dots & \frac{\partial F_n}{\partial x_n}(x^{(k)}) \end{bmatrix}. \tag{6.10.2}$$

Cantitatea $1/f'(x)$ se înlocuieşte în acest caz cu inversa jacobianului în $x^{(k)}$:

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}). \tag{6.10.3}$$

Scriem iteraţia sub forma

$$x^{(k+1)} = x^{(k)} + w^{(k)}. \tag{6.10.4}$$

Se observă că $w_k$ este soluţia sistemului de $n$ ecuaţii liniare cu $n$ necunoscute

$$F'(x^{(k)}) w^{(k)} = -F(x^{(k)}). \tag{6.10.5}$$

Este mai eficient şi mai convenabil ca, în loc să inversăm jacobianul la fiecare pas, să rezolvăm sistemul (6.10.5) şi să folosim iteraţia în forma (6.10.4).

**Teorema 6.10.1** *Fie $\alpha$ o soluţie a ecuaţiei $F(x) = 0$ şi presupunem că în bila închisă $B(\delta) \equiv \{x : \|x-\alpha\| \le \delta\}$, există matricea Jacobi a lui $F : \mathbb{R}^n \to \mathbb{R}^n$, este nesingulară şi satisface condiţia Lipschitz*

$$\|F'(x) - F'(y)\|_\infty \le c\|x - y\|_\infty, \ \forall\, x, y \in B(\delta), \quad c > 0.$$

*Punem $\gamma = c \max\{\|[F'(x)]^{-1}\|_\infty : \|\alpha - x\|_\infty \le \delta\}$ şi $0 < \varepsilon < \min\{\delta, \gamma^{-1}\}$. Atunci pentru orice aproximaţie iniţială $x^{(0)} \in B(\varepsilon) := \{x : \|x - \alpha\|_\infty \le \varepsilon\}$ metoda lui Newton este convergentă, iar vectorii $e^{(k)} := \alpha - x^{(k)}$ satisfac următoarele inegalităţi:*

*(a)* $\|e^{(k+1)}\|_\infty \le \gamma \|e^{(k)}\|_\infty^2$

*(b)* $\|e^{(k)}\|_\infty \le \gamma^{-1} (\gamma \|e^{(0)}\|_\infty)^{2^k}$.

*Demonstraţie.* Dacă $F'$ este continuă pe segmentul ce uneşte punctele $x, y \in \mathbb{R}^n$, conform teoremei lui Lagrange

$$F(x) - F(y) = J_k(x - y),$$

unde

$$J_k = \begin{bmatrix} \frac{\partial F_1}{\partial x_1}(\xi_1) & \dots & \frac{\partial F_1}{\partial x_n}(\xi_1) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1}(\xi_n) & \dots & \frac{\partial F_n}{\partial x_n}(\xi_n) \end{bmatrix} \Rightarrow$$

$$e^{(k+1)} = e^{(k)} - [F'(x^{(k)})]^{-1}(F(\alpha) - F(x^{(k)})) = e^{(k)} - [F'(x^{(k)})]^{-1}J_k e^{(k)}$$
$$= [F'(x_{(k)})]^{-1}(F'(x^{(k)}) - J_k)e^{(k)}$$

şi de aici rezultă imediat (a). Din condiţia 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 obţine (b) imediat. □

Metoda lui Newton pentru sisteme de ecuaţii neliniare este descrisă în algoritmul 6.3.

**Algoritmul 6.3** Metoda lui Newton pentru sisteme de ecuaţii neliniare

**Intrare:** Funcţia $F$, derivata Fréchet $F'$, vectorul de pornire $x^{(0)}$, numarul maxim de iteraţii, $Nmax$, informaţii de toleranţă $tol$

**Ieşire:** O aproximaţie 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 depăşit numărul de iteraţii”).

## 6.11. Metode quasi-Newton

O slăbiciune semnificativă a metodei lui Newton pentru rezolvarea sistemelor de ecuaţii neliniare este necesitatea ca la fiecare pas să calculăm matricea jacobiană şi 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 iteraţie a metodei lui Newton. Matricea jacobiană asociată unui sistem de $n$ ecuaţii neliniare scris în forma $F(x) = 0$ necesită evaluarea celor $n^2$ derivate parţiale ale celor $n$ funcţii componente ale lui $F$. În cele mai multe situaţii, evaluarea exactă a derivatelor parţiale este neconvenabilă şi de multe ori imposibilă. Efortul total pentru o iteraţie a metodei lui Newton va fi de cel puţin $n^2+n$ evaluări de funcţii scalare ($n^2$ pentru evaluarea jacobianului şi $n$ pentru evaluarea lui $F$) şi $O(n^3)$ operaţii aritmetice pentru a rezolva sistemul liniar. Acest volum de

calcule este prohibitiv, exceptând valori mici ale lui $n$ şi funcţii scalare uşor de evaluat. Este firesc ca atenţia să fie îndreptată spre reducerea numărului de evaluări şi evitarea rezolvării unui sistem liniar la fiecare pas.

La metoda secantei aproximaţia următoare $x^{(k+1)}$ se obţine ca soluţie a ecuaţiei liniare

$$\bar{l}_k = f(x^{(k)}) + (x - x^{(k)})\frac{f(x_{(k)} + h_k) - f(x_{(k)})}{h_k} = 0.$$

Aici funcţia $\bar{l}_k$ poate fi interpretată în două moduri:

1. ca aproximare a ecuaţiei tangentei

$$l_k(x) = f(x^{(k)}) + (x - x^{(k)})f'(x^{(k)});$$

2. ca interpolare liniară între punctele $x^{(k)}$ şi $x^{(k+1)}$.

Se pot obţine diverse generalizări ale metodei secantei la sisteme de ecuaţii neliniare în funcţie de modul în care se interpretează $\bar{l}_k$. Prima interpretare conduce la metode de tip Newton discretizate, iar a doua la metode bazate pe interpolare.

Metodele de tip Newton discretizate se obţin dacă în metoda lui Newton (6.10.3) $F'(x)$ se înlocuieşte cu cu o aproximare discretă $A(x, h)$. Derivatele parţiale din matricea jacobiană (6.10.2) se vor înlocui prin diferenţele divizate

$$A(x, h)e_i := [F(x + h_i e_i) - F(x)]/h_i, \qquad i = \overline{1, n}, \tag{6.11.1}$$

unde $e_i \in \mathbb{R}^n$ este al $i$-lea vector al bazei canonice şi $h_i = h_i(x)$ 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 maşinii.

## 6.11.1. Interpolare liniară

La interpolare fiecare dintre planele tangente se înlocuieşte cu un (hiper)plan care interpolează funcţiile 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 $a^{(i)}$ şi scalarii $\alpha_i$, astfel încât pentru

$$L_i(x) = \alpha_i + a^{(i)T}x, \qquad i = \overline{1, n} \tag{6.11.2}$$

are loc

$$L_i(x^{k,j}) = F_i(x^{k,j}), \qquad i = \overline{1, n}, \quad j = \overline{0, n}.$$

Următoarea aproximaţie $x^{(k+1)}$ se obţine ca punct de intersecţie între cele $n$ hiperplane (6.11.2) din $\mathbb{R}^{n+1}$ cu hiperplanul $y = 0$. $x^{(k+1)}$ rezultă ca soluţie a sistemului de ecuaţii liniare

$$L_i(x) = 0, \qquad i = \overline{1, n}. \tag{6.11.3}$$

În funcţie de alegerea punctelor de interpolare se obţin diferite metode, dintre care cele mai cunoscute sunt metoda lui Brown şi metoda lui Brent. Metoda lui Brown combină aproximarea lui $F'$ şi rezolvarea sistemului prin eliminare gaussiană. În metoda lui Brent se întrebuinţează la rezolvarea sistemului metoda QR. Ambele metode aparţin unei clase, care, la fel ca metoda lui Newton converg pătratic, dar au nevoie doar de $(n^2 + 3n)/2$ evaluări de funcţii pe iteraţie.

Într-un studiu comparativ, Moré şi Cosnard [28] au ajuns la concluzia că metoda Brent este adeseori de preferat metodei lui Brown şi că pentru sisteme de ecuaţii neliniare, la care evaluarea lui $F$ necesită un efort mai mic, metoda lui Newton discretizată este cea mai eficientă metodă de rezolvare.

## 6.11.2. Metode de modificare

Din punct de vedere al efortului de calcul, sunt deosebit de convenabile metodele în care la fiecare pas se întrebuinţează o aproximare $A_k$ a lui $F'(x^{(k)})$, care se obţine din $A_{k-1}$ printr-o modificare de rang 1, adică prin adăugarea unei matrice de rang 1:

$$A_{k+1} := A_k + u^{(k)} \left[v^{(k)}\right]^T, \qquad u^{(k)}, v^{(k)} \in \mathbb{R}^n,\ k = 0, 1, 2, \ldots$$

Pe baza formulei Sherman-Morrison (vezi [14])

$$\left(A + uv^T\right)^{-1} = A^{-1} - \frac{1}{1 + v^T A^{-1} u} A^{-1} u v^T A^{-1},$$

pentru $B_{k+1} := A_{k+1}^{-1}$ are loc relaţia de recurenţă

$$B_{k+1} = B_k - \frac{B_k u^{(k)} \left[v^{(k)}\right]^T B_k}{1 + \left[v^{(k)}\right]^T B_k u^{(k)}}, \qquad k = 0, 1, 2, \ldots,$$

atât timp cât $1 + \left[v^{(k)}\right]^T B_k u^{(k)} \neq 0$. Astfel, nu mai este necesară rezolvarea unui sistem liniar la fiecare pas; ea se înlocuieşte cu înmulţiri 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 convergentă pătratică ca la metoda lui Newton, ci doar una superliniară:

$$\lim_{k \to \infty} \frac{\|x^{(k+1)} - \alpha\|}{\|x^{(k)} - \alpha\|} = 0. \tag{6.11.4}$$

În metoda lui Broyden alegerea vectorilor $u^{(k)}$ şi $v^{(k)}$ are loc după principiul aproximaţiei secantei. În cazul scalar aproximarea $a_k \approx f'(x^{(k)})$ se face unic prin

$$a_{k+1}(x^{(k+1)} - x^{(k)}) = f(x^{(k+1)} - f(x^{(k)}).$$

Pentru $n > 1$, din contră, aproximarea

$$A_{k+1}(x^{(k+1)} - x^{(k)}) = F(x^{(k+1)}) - F(x^{(k)}) \tag{6.11.5}$$

(aşa numita ecuaţie quasi-Newton) nu mai este unic determinată; orice altă matrice de forma

$$\bar{A}_{k+1} := A_{k+1} + pq^T$$

cu $p, q \in \mathbb{R}^n$ şi $q^T(x^{(k+1)} - x^{(k)}) = 0$ verifică de asemenea ecuaţia (6.11.5). Pe de altă parte,

$$y_k := F(x^{(k)}) - F(x^{(k-1)}) \text{ şi } s_k := x^{(k)} - x^{(k-1)}$$

conţin numai informaţii despre derivata parţială a lui $F$ în direcţia $s_k$, dar nici o informaţie în direcţii ortogonale pe $s_k$. Pe această direcţie trebuie ca efectul lui $A_{k+1}$ şi $A_k$ să coincidă

$$A_{k+1}q = A_k q, \qquad \forall q \in \{v : v \neq 0, v^T s_k = 0\}. \tag{6.11.6}$$

Pornind de la prima aproximare $A_0 \approx F'(x^{(0)})$, se generează şirul $A_1$, $A_2$, ... utilizând formulele (6.11.5) şi (6.11.6) (Broyden [7], Dennis şi Moré [14]).

Pentru şirul $B_0 = A_0^{-1} \approx [F(x^{(0)})]^{-1}$, $B_1$, $B_2$, ... cu ajutorul formulei Sherman-Morisson se obţine relaţia de recurenţă

$$B_{k+1} := B_k + \frac{(s_{k+1} - B_k y_{k+1}) s_{k+1}^T B_k}{s_{k+1}^T B_k y_{k+1}}, \qquad k = 0, 1, 2, \ldots$$

care necesită doar înmulţiri matrice vector şi a cărei complexitate este doar $O(n^2)$. Cu ajutorul matricelor $B_k$ se poate defini metoda lui Broyden prin

$$x^{(k+1)} := x^{(k)} - B_k F(x^{(k)}), \qquad k = 0, 1, 2, \ldots$$

Această metodă converge superliniar în sensul lui (6.11.4), dacă paşii $s_k$ se apropie asimptotic (când $k \to \infty$) de vectorii de actualizare (corecţie) ai metodei lui Newton. Se poate recunoaşte în aceasta semnificaţia centrală a principiului linearizării locale la rezolvarea ecuaţiilor neliniare.

Metoda lui Broyden este descrisă în algoritmul 6.4.

**Algoritmul 6.4** Metoda lui Broyden pentru sisteme de ecuaţii neliniare

**Intrare:** Funcţia $F$, vectorul de pornire $x^{(0)}$, numarul maxim de iteraţii, $Nmax$, informaţii de toleranţă $tol$

**Ieşire:** O aproximaţie a rădăcinii sau un mesaj de eroare

1: $B_0 := F'(x^{(0)}); \quad v := F(x); \quad B := B_0^{-1};$
2: $s := -Bv; \quad x := x + s;$
3: **for** $k := 1$ **to** $Nmax$ **do**
4: $\quad w := v; \quad v := F(x); \quad y := v - w;$
5: $\quad z := -By;$ $\{z = -B_{k-1}y_k\}$
6: $\quad p := -s^T z;$ $\{p = s_k^T B_{k-1} y_k\}$
7: $\quad C := pI + (s + z)s^T;$ $\{C = s_k^T B_{k-1}^{-1} y_k I + (s_k + B_{k-1}y_k)s_k^T\}$
8: $\quad B := (1/p)CB;$ $\{B = B_k\}$
9: $\quad s := -Bv;$ $\{s = -B_k F(x^{(k)})\}$
10: $\quad x := x + s;$
11: $\quad$ **if** $crit\_oprire(tol)$ **then**
12: $\quad\quad$ **return** $x$; {succes}
13: $\quad$ **end if**
14: **end for**
15: $error$("S-a depăşit numărul maxim de iteraţii")

# CAPITOLUL 7

# Vectori şi valori proprii

## Cuprins



În acest capitol ne ocupăm de determinarea valorilor (şi vectorilor) proprii ale unei matrice pătratice $A \in \mathbb{R}^{n\times n}$, adică a valorilor $\lambda \in \mathbb{C}$ şi vectorilor $x \in \mathbb{C}^n$ pentru care

$$Ax = \lambda x. \tag{7.0.1}$$

**Definiţia 7.0.1** *Numărul $\lambda \in \mathbb{C}$ se numeşte* valoare proprie *a matricei $A \in \mathbb{R}^{n\times n}$, când există un vector $x \in \mathbb{C}^n \setminus \{0\}$ numit* vector propriu *astfel încât $Ax = \lambda x$.*

**Observaţia 7.0.2.**

1. Cerinţa $x \neq 0$ este importantă, căci vectorul nul este un vector propriu corespunzător oricărei valori proprii.
2. Chiar dacă $A$ este reală, ea poate avea valori proprii complexe. În acest caz ele apar în perechi conjugate. ♢

# 7.1. Valori proprii şi rădăcini ale polinoamelor

Orice problemă de calcul al valorilor proprii se poate reduce la calculul zerourilor unui polinom: valorile proprii ale unei matrice $A \in \mathbb{R}^{n\times n}$ sunt rădăcinile *polinomului caracteristic*

$$p_A\lambda = \det(A - \lambda I), \qquad \lambda \in \mathbb{C},$$

căci determinatul este nul exact atunci când sistemul $(A - \lambda I)x = 0$ are o soluţie nebanală, adică atunci când $\lambda$ este o valoare proprie.

O metodă de rezolvare a problemelor proprii ar putea fi calculul polinomului caracteristic şi apoi determinarea rădăcinilor. Natural, calculul unui determinant *în general* fiind o problemă complexă şi instabilă, transformarea matricei ar fi mai potrivită. Reciproc, problema găsirii rădăcinilor unui polinom poate fi formulată ca o problemă de determinare a valorilor proprii. Fie $p \in \mathbb{P}_n$ un polinom cu coeficienţi reali, pe care îl putem scrie (cu ajutorul rădăcinilor sale, $z_1, \ldots, z_n$, eventual complexe) sub forma

$$p(x) = a_n x^n + \cdots + a_0 = a_n(x - z_1)\ldots(x - z_n), \quad a_n \in \mathbb{R}, \quad a_n \neq 0.$$

Pe spaţiul vectorial $\mathbb{P}_{n-1}$ „înmulţirea modulo $p$”

$$\mathbb{P}_{n-1} \ni q \to r \qquad xq(x) = \alpha p(x) + r(x), \qquad r \in \mathbb{P}_n \tag{7.1.1}$$

este o *transformare liniară* şi deoarece

$$x^n = \frac{1}{a_n}p(x) - \sum_{j=0}^{n-1}\frac{a_j}{a_n}x^j, \qquad x \in \mathbb{R},$$

vom reprezenta $p$ relativ la baza $1, x, \ldots, x^{n-1}$ prin aşa-numita matrice *companion* a lui Frobenius (de dimensiune $n \times n$)

$$M = \begin{bmatrix} 0 & & & & -\frac{a_0}{a_n} \\ 1 & 0 & & & -\frac{a_1}{a_n} \\ & \ddots & \ddots & & \vdots \\ & & 1 & 0 & -\frac{a_{n-2}}{a_n} \\ & & & 1 & -\frac{a_{n-1}}{a_n} \end{bmatrix}, \tag{7.1.2}$$

Fie $v_j = (v_{jk} : k = \overline{1,n}) \in \mathbb{C}^n$, $j = \overline{1,n}$ alese astfel ca

$$\ell_j(x) = \frac{p(x)}{x - z_j} = a_n\prod_{k\neq j}(x - z_k) = \sum_{k=1}^{n} v_{jk}x^{k-1}, \qquad j = \overline{1,n},$$

atunci

$$\sum_{k=1}^{n}(Mv_j - z_j v_j)_k x^{k-1} = x\ell_j(x) - z_j\ell_j(x) = (x - z_j)\ell_j(x) = p(x) \approx 0,$$

şi cu aceasta $Mv_j = z_j v_j, \quad j = \overline{1,n}$.

*Valorile proprii ale lui $M$ sunt deci rădăcini ale lui $p$.*

Matricea Frobenius dată de (7.1.2) este doar o modalitate dintre multe altele prin care se poate reprezenta „înmulţirea" din (7.1.1); orice altă bază a lui $\mathbb{P}_{n-1}$ furnizează o matrice $M$ ale cărei valori proprii să fie rădăcini ale lui $p$. Singurul mijloc auxiliar de manipulare a polinoamelor de care avem nevoie este realizarea unei „împărţiri cu rest".

## 7.2. Terminologie şi descompunere Schur

Asa cum ne arată exemplul

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \qquad p_A(\lambda) = \lambda^2 + 1 = (\lambda + i)(\lambda - i),$$

o matrice reală poate avea valori proprii complexe. De aceea (cel puţin în teorie) este avantajos să ne ocupăm de matrice complexe $A \in \mathbb{C}^{n\times n}$.

**Definiţia 7.2.1** *Două matrice $A, B \in \mathbb{C}^{n\times n}$ se numesc* similare *dacă există o matrice $T \in \mathbb{C}^{n\times n}$, astfel încât*

$$A = TBT^{-1}.$$

**Lema 7.2.2** *Dacă $A, B \in \mathbb{C}^{n\times n}$sunt similare, ele au aceleaşi valori proprii.*

*Demonstraţie.* Fie $\lambda \in \mathbb{C}$ o valoare proprie a lui $A = TBT^{-1}$ şi $x \in \mathbb{C}^n$ vectorul propriu corespunzător. Atunci avem

$$B(T^{-1}x) = T^{-1}ATT^{-1}x = T^{-1}Ax = \lambda T^{-1}x$$

şi deci, $\lambda$ este, de asemenea, valoare proprie a lui $B$. □

Din algebra liniară se cunoaşte următorul rezultat important.

**Teorema 7.2.3 (Forma normală Jordan)** *Orice matrice $A \in \mathbb{C}^{n\times n}$ este similară cu o matrice*

$$J = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_k \end{bmatrix}, \quad J_\ell = \begin{bmatrix} \lambda_\ell & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_\ell \end{bmatrix} \in \mathbb{C}^{n_\ell \times n_\ell}, \qquad \sum_{\ell=1}^{k} n_\ell = n,$$

*numită* forma normală Jordan *a lui $A$.*

**Definiţia 7.2.4** *O matrice se numeşte* diagonalizabilă, *dacă toate blocurile sale Jordan $J_\ell$ au dimensiunea 1, adică $n_\ell = 1$, $\ell = \overline{1,n}$. O matrice se numeşte* nederogatorie *dacă pentru fiecare valoare proprie $\lambda_\ell$ există exact un bloc Jordan în care ea apare pe diagonală.*

**Observaţia 7.2.5.** Dacă o matrice $A \in \mathbb{R}^{n\times n}$ are $n$ valori proprii simple, atunci ea este diagonalizabilă şi de asemenea nederogatorie şi potrivită pentru a fi tratată numeric. $\diamondsuit$

**Teorema 7.2.6 (Descompunere Schur)** *Pentru orice matrice $A \in \mathbb{C}^{n\times n}$ există o matrice unitară $U \in \mathbb{C}^{n\times n}$ şi o matrice triunghiulară superior*

$$R = \begin{bmatrix} \lambda_1 & * & \dots & * \\ & \ddots & \ddots & \vdots \\ & & \ddots & * \\ & & & \lambda_n \end{bmatrix} \in \mathbb{C}^{n\times n},$$

*astfel încât $A = URU^*$.*

**Observaţia 7.2.7.**

1. Elementele diagonale ale lui $R$ sunt, natural, valorile proprii ale lui $A$. Deoarece $A$ şi $R$ sunt similare, ele au aceleaşi valori proprii.
2. Între $A$ şi $R$ are loc o formă mai puternică de similaritate: ele sunt *unitar-similare*. $\diamondsuit$

*Demonstraţia teoremei 7.2.6.* Demonstraţia se face prin inducţie. Cazul $n = 1$ este trivial. Presupunem teorema adevărată pentru $n \in \mathbb{N}$ şi fie $A \in \mathbb{C}^{(n+1)\times(n+1)}$. Fie $\lambda \in \mathbb{C}$ o valoare proprie a lui $A$ şi $x \in \mathbb{C}^{n+1}$, $\|x\|_2 = 1$, vectorul propriu corespunzător. Luăm vectorul $u_1 = x$ şi alegem $u_2, \dots, u_{n+1}$ astfel încât $u_1, \dots, u_{n+1}$ să formeze o bază ortonormală a lui $\mathbb{C}^{n+1}$, sau echivalent, matricea $U = [u_1, \dots, u_{n+1}]$ să fie unitară. Aşadar,

$$U^*AUe_1 = U^*Au_1 = U^*Ax = \lambda U^*x = \lambda e_1,$$

adică

$$U^*AU = \begin{bmatrix} \lambda & * \\ 0 & B \end{bmatrix}, \qquad B \in \mathbb{C}^{n\times n}.$$

Conform ipotezei inducţiei există o matrice unitară $V \in \mathbb{C}^{n\times n}$, astfel încât $B = VSV^*$, unde $S \in \mathbb{C}^{n\times n}$ este o matrice triunghiulară superior. De aceea

$$A = U \begin{bmatrix} \lambda_1 & * \\ 0 & VSV^* \end{bmatrix} U^* = \underbrace{U \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}}_{=:U} \underbrace{\begin{bmatrix} \lambda_1 & * \\ 0 & S \end{bmatrix}}_{=:R} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & V^* \end{bmatrix} U^*}_{=U^*}$$

şi demonstraţie este completă. $\square$

Să dăm acum două consecinţe nemijlocite ale descompunerii Schur.

**Corolarul 7.2.8** *Oricărei matrice hermitiene $A \in \mathbb{C}^{n\times n}$ îi corespunde o matrice ortogonală $U \in \mathbb{C}^{n\times n}$ astfel încât*

$$A = U \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} U^*, \qquad \lambda_j \in \mathbb{R}, \qquad j = \overline{1,n}.$$

*Demonstraţie.* Matricea $R$ din teorema 7.2.6 verifică $R = U^*AU$. Deoarece

$$R^* = (U^*AU) = U^*A^*U = U^*AU = R,$$

$R$ trebuie să fie diagonală, cu elemente reale pe diagonală (fiind hermitiană). □

Altfel spus, corolarul 7.2.8 ne asigură că orice matrice hermitiană este unitar diagonalizabilă şi posedă o bază formată din vectori proprii ortonormali. În plus, toate valorile proprii ale unei matrice hermitiene sunt reale. Este interesant, nu numai din punct de vedere teoretic, ce matrice sunt unitar diagonalizabile.

**Teorema 7.2.9** *O matrice $A \in \mathbb{C}^{n\times n}$ este unitar diagonalizabilă, adică există o matrice unitară $U \in \mathbb{C}^{n\times n}$ astfel încât*

$$A = U \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} U^*, \qquad \lambda_j \in \mathbb{R}, \qquad j = 1, \ldots, n. \tag{7.2.1}$$

*dacă şi numai dacă $A$ este normală, adică*

$$AA^* = A^*A. \tag{7.2.2}$$

*Demonstraţie.* Punem $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. Conform lui (7.2.1), $A$ are forma $A = U\Lambda U^*$, deci

$$AA^* = U\Lambda U^*U\Lambda^*U^* = U|\Lambda|^2U^* \text{ şi } A^*A = U^*\Lambda^*U^*U\Lambda U^* = U|\Lambda|^2U^*,$$

adică (7.2.2). Pentru reciprocă, folosim descompunerea Schur a lui $A$ sub forma $R = U^*AU$. Atunci

$$|\lambda_1|^2 = (R^*R)_{11} = (RR^*)_{11} = |\lambda_1|^2 + \sum_{k=2}^{n} |r_{1k}|^2,$$

de unde rezultă $r_{12} = \cdots = r_{1n} = 0$. Prin inducţie, se vede că pentru $j = \overline{2,n}$

$$(R^*R)_{jj} = |\lambda_j|^2 + \sum_{k=1}^{j-1} |r_{kj}|^2 = (RR^*_{jj} = |\lambda_j|^2 + \sum_{k=j+1}^{n} |r_{jk}|^2,$$

din care cauză $R$ trebuie să fie diagonală. □

Pentru matricea reale *descompunerea Schur reală* este un pic mai complicată.

**Teorema 7.2.10** *Pentru orice matrice $A \in \mathbb{R}^{n\times n}$ există o matrice ortogonală $U \in \mathbb{R}^{n\times n}$ astfel încât*

$$A = U \begin{bmatrix} R_1 & * & \dots & * \\ & \ddots & \ddots & \vdots \\ & & \ddots & * \\ & & & R_k \end{bmatrix} U^*, \tag{7.2.3}$$

*în care fie $R_j \in \mathbb{R}^{1\times 1}$, fie $R_j \in \mathbb{R}^{2\times 2}$, cu două valori proprii complexe conjugate, $j = \overline{1,k}$.*

Descompunerea Schur reală transformă $A$ într-o matrice Hessenberg superioară

$$U^T A U = \begin{bmatrix} * & \dots & \dots & * \\ * & \ddots & & \vdots \\ & \ddots & \ddots & \vdots \\ & & * & * \end{bmatrix}.$$

*Demonstraţie.* Dacă $A$ are doar valori proprii reale, atunci se procedează ca la descompunerea Schur complexă. Altfel, fie $\lambda = \alpha + i\beta$, $\beta \neq 0$, o valoare proprie complexă a lui $A$ şi $x + iy$ vectorul propriu corespunzător. Atunci

$$A(x+iy) = \lambda(x+iy) = (\alpha + i\beta)(x+iy) = (\alpha x - \beta y) + i(\beta x + \alpha y)$$

sau matricial

$$A \underbrace{[x\ y]}_{\in \mathbb{R}^{n\times 2}} = [x\ y] \underbrace{\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}}_{:=R}.$$

Deoarece $R = \alpha^2 + \beta^2 > 0$, căci $\beta \neq 0$, $\operatorname{span}\{x, y\}$ este un subspaţiu bidimensional $A$-invariant al lui $\mathbb{R}^n$. Atunci alegem $u_1$, $u_2$ astfel încât să formeze o bază a acestui spaţiu, completată cu $u_3, \dots, u_n$ la o bază ortonormală a lui $\mathbb{R}^n$ şi raţionând analog cu cazul complex obţinem

$$U^T A U = \begin{bmatrix} R & * \\ 0 & B \end{bmatrix}$$

şi inducţia se derulează ca la descompunerea Schur complexă. □

# 7.3. Iteraţia vectorială

Iteraţia vectorială (numită şi metoda puterii) este cea mai simplă metodă atunci când dorim o valoare proprie şi vectorul propriu corespunzător.

Pornind de la un vector $y^{(0)} \in \mathbb{C}^n$ se construieşte şirul $y^k$, $k \in \mathbb{N}$ prin intermediul iteraţiei

$$\begin{aligned} z^{(k)} &= Ay^{(k-1)}, \\ y^{(k)} &= \frac{z^{(k)}}{z_{j*}^{(k)}} \frac{|z_{j*}^{(k)}|}{\|z^{(k)}\|_\infty}, \quad j = \min\left\{1 \le j \le n : \left|z_j^{(k)}\right| \ge \left(1 - \frac{1}{k}\right) \|z^{(k)}\|_\infty\right\} \end{aligned} \tag{7.3.1}$$

şi se afirmă că, în condiţii determinate, acest şir converge către vectorul propriu dominant.

**Propoziţia 7.3.1** *Fie $A \in \mathbb{C}^{n\times n}$ o matrice diagonalizabilă ale cărei valori proprii $\lambda_1, \ldots \lambda_n$ verifică condiţia*

$$|\lambda_1| > |\lambda_2| \ge \ldots |\lambda_n|.$$

*Atunci şirul $y^{(k)}$, $k \in \mathbb{N}$, converge către un multiplu al vectorului normat $x_1$ corespunzător valorii proprii $\lambda_1$, pentru aproape orice $y^{(0)}$.*

*Demonstraţie.* Fie $x_1, \ldots, x_n$ vectorii proprii ortonormali ai lui $A$ corespunzători valorilor proprii $\lambda_1, \ldots, \lambda_n$ – existenţa lor rezultă din diagonalizabilitatea lui $A$. Scriem

$$y^{(0)} = \sum_{j=1}^{n} \alpha_j x_j, \qquad \alpha_j \in \mathbb{C}, \qquad j = \overline{1,n},$$

şi afirmăm că

$$A^k y^{(0)} = \sum_{j=1}^{n} \alpha_j A^k x_j = \sum_{j=1}^{n} \alpha_j \lambda^k x_j = \lambda_1^k \sum_{j=1}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1}\right)^k x_j.$$

De aici rezultă, deoarece $|\lambda_1| > |\lambda_j|$, $j = \overline{2,n}$ că

$$\lim_{k\to\infty} \lambda_1^{-k} A^k y^{(0)} = \alpha_1 x_1 + \lim_{k\to\infty} \sum_{j=2}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1}\right)^k x_k = \alpha_1 x_1,$$

precum şi

$$\lim_{k\to\infty} |\lambda_1|^{-k} \left\|A^k y^{(0)}\right\|_\infty = \left\|\sum_{j=1}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1}\right)^k \alpha_j x_j\right\| = |\alpha_1| \|x_1\|_\infty .$$

Dacă $\alpha_1 = 0$ şi deci $y^{(0)}$ aparţine hiperplanului

$$x_1^+ = \{x \in \mathbb{C}^n, x^* x_1 = 0\},$$

atunci ambele limite sunt nule şi nu se poate spune nimic despre convergenţa şirului $y^k$, $k \in \mathbb{N}$; acest hiperplan este o mulţime de măsură nulă, aşa că în continuare vom presupune că $\alpha_1 \neq 0$.

Datorită lui (7.3.1), $y^k = \gamma_k A^k y^{(0)}$, $\gamma_k \in \mathbb{C}$ şi pe lângă aceasta $\|y^k\|_\infty = 1$, aşadar

$$\lim_{k\to\infty} |\lambda_1|^k |\gamma_k| = \lim_{k\to\infty} \frac{1}{|\lambda_1|^{-1}\|A^{(k)}y^{(0)}\|} = \frac{1}{|\alpha_1|\|x_1\|_\infty}.$$

Astfel

$$y^{(k)} = \gamma_k A^k y^{(0)} = \underbrace{\frac{\gamma_k \lambda_1^k}{|\gamma_k \lambda_1^k|}}_{=:e^{-2\pi i\theta_k}} \underbrace{\frac{\alpha_1 x_1}{|\alpha_1|\|x_1\|_\infty}}_{=:\alpha x_1} + O\left(\frac{|\lambda_2|^k}{|\lambda_1|^k}\right), \qquad k \in \mathbb{N}, \tag{7.3.2}$$

unde $\theta_k \in [0,1]$. Aici intervine normarea mai ciudată din (7.3.1): fie $j$ indicele minim pentru care $|(\alpha x_1)_j| = \|\alpha x_1\|_\infty\|$; atunci conform lui (7.3.2) pentru un $k$ suficient de mare de asemenea în (7.3.1) $j^* = j$. Deci are loc

$$\lim_{k\to\infty} y_j^{(k)} = 1 \Rightarrow \lim_{k\to\infty} e^{2\pi i\theta_k} = \lim_{k\to\infty} \frac{y_j^{(k)}}{(\alpha x_1)_j} = \frac{1}{(\alpha x_1)_j}.$$

Înlocuid aceasta în (7.3.2) obţinem convergenţa şirului $y^{(k)}$, $k \in \mathbb{N}$. □

Se poate aplica de asemenea iteraţia vectorială pentru a găsi toate valorile proprii şi toţi vectorii proprii, în măsura în care valorile proprii ale lui $A$ sunt diferite în modul. Pentru aceasta se determină cea mai mare în modul valoare proprie $\lambda_1$ a lui $A$ şi vectorul propriu corespunzător $x_1$ şi se continuă cu

$$A^{(1)} = A - \lambda_1 x_1 x_1^T.$$

Matricea diagonalizabilă $A^{(1)}$ are aceeaşi vectori proprii ortonormali ca şi $A$, doar că $x_1$ este vectorul propriu corespunzător valorii proprii 0 şi nu mai joacă nici un rol în iteraţie, atât timp cât nu se porneşte chiar cu un multiplu al lui $x_1$. Aplicând încă odată iteraţia vectorială lui $A^{(1)}$ se obţine a doua valoare proprie ca mărime în modul $\lambda_2$ şi vectorul propriu corespunzător; iteraţia

$$A^{(j)} = A^{(j-1)} - \lambda_j x_j x_j^T, \qquad j = \overline{1,n}, \quad A^{(0)} = A$$

calculează succesiv toate valorile proprii şi toţi vectorii proprii ai lui $A$, presupunând că valorile proprii sunt diferite în modul.

**Observaţia 7.3.2 (Dezavantajele iteraţiei vectoriale).** 1. Metoda funcţionează în general doar când există un vector propriu dominant, adică când există pentru

o valoare proprie dominantă exact un vector propriu. Dacă se consideră, de exemplu, matricea

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

atunci ea transformă vectorul $[x_1\ x_2]^T$ în vectorul $[x_2\ x_1]^T$ şi convergenţa are loc exact atunci când iteraţia porneşte cu unul din vectorii proprii.

2. Metoda dă rezultate numai pentru vectori de pornire „potriviţi”. Sună nemaipomenit că toţi vectorii de pornire cu excepţia celor dintr-un hiperplan sunt potriviţi, dar nu este chiar aşa de simplu. Dacă valoarea proprie cea mai mare în modul a unei matrice reale este complexă atunci se poate itera la nesfârşit cu valori de pornire reale, căci oricum nu se va găsi vectorul propriu.

3. Ar trebui făcute tot timpul calcule în complex, ceea ce ar ridica serios complexitatea (la adunare un număr dublu de calcule, la înmulţire de şase ori mai multe).

4. Viteza de convergenţă depinde de raportul

$$\frac{|\lambda_2|}{|\lambda_1|} < 1$$

care poate fi oricât de aproape de 1. Dacă dominanţa vectorului propriu dominant nu este bine reliefată, atunci convergenţa poate fi extrem de lentă. ♢

Ţinând cont de toate acestea, conchidem că iteraţia vectorială nu este o metodă prea bună pentru problemele de valori proprii.

## 7.4. Metoda QR – teoria

Metoda practică pentru tratarea problemelor de valori proprii este în zilele noastre metoda QR descoperită de Francis [18] şi Kublanovskaya [27], o extensie unitară a metodei LR a lui Rutishauser [32]. Vom începe cu cazul complex.

Metoda este una iterativă extrem de simplă: se porneşte cu $A^{(0)} = A$ şi se calculează iterativ cu ajutorul descompunerii QR,

$$A^{(k)} = Q_k R_k, \qquad A^{(k+1)} = R_k Q_k, \qquad k \in \mathbb{N}_0. \tag{7.4.1}$$

Cu puţin noroc, sau aşa cum spun matematicienii, în anumite ipoteze, acest şir va converge către o matrice ale cărei elemente diagonale sunt valorile proprii ale lui $A$.

**Lema 7.4.1** *Matricele $A^{(k)}$ construite prin* (7.4.1)*, $k \in \mathbb{N}$, sunt similar ortogonale cu $A$ şi au aceleaşi valori proprii ca $A$.*

*Demonstraţie.* Are loc

$$A^{(k+1)} = Q_k^* Q_k R_k Q_k = Q_k^* A^{(k)} Q_k = \cdots = \underbrace{Q_k^* \ldots Q_0^*}_{=:U_k^*} A \underbrace{Q_0 \ldots Q_k}_{=:U_k}.$$

□

Pentru a arăta convergenţa, vom interpreta iteraţia QR ca o generalizare a iteraţiei vectoriale (7.3.1) (fără normarea ciudată) la spaţii vectoriale. Pentru aceasta vom scrie baza ortonormală $u_1, \ldots, u_m \in \mathbb{C}^n$ a unui subspaţiu $m$-dimensional $U \subset \mathbb{C}^n$, $m \leq n$, ca vectori coloane ai unei matrice unitare $U \in \mathbb{R}^{n\times m}$ şi iterăm subspaţiul vectorial (respectiv matricele) peste descompunerea QR

$$U_{k+1} R_k = A U_k, \quad k \in \mathbb{N}_0, \quad U_0 \in \mathbb{C}^n. \tag{7.4.2}$$

De aici rezultă imediat că

$$U_{k+1}(R_k \ldots R_0) = A U_k (R_{k-1} \ldots R_0) = A^2 U_{k-1}(R_{k-2} \ldots R_0) = \ldots = A^{k+1} U_0. \tag{7.4.3}$$

Dacă definim acum, pentru $m = n$, $A^{(k)} = U_k^* A U_k$, atunci conform lui (7.4.2) au loc relaţiile

$$\begin{aligned} A^{(k)} &= U_k^* A U_k = U_k^* U_{k+1} R_k \\ A^{(k+1)} &= U_{k+1}^* A U_{k+1} = U_{k+1}^* A U_k U_k^* U_{k+1} \end{aligned}$$

şi punând $Q_k := U_k^* U_{k+1}$, obţinem regula de iterare (7.4.1). Ca matrice de pornire putem alege $U_0 = I$.

Pentru a obţine rezultate despre metoda QR mai avem nevoie de un tip de matrice.

**Definiţia 7.4.2** *O* matrice de fază $\Theta \in \mathbb{C}^{n\times n}$ *este o matrice diagonală de forma*

$$\Theta = \begin{bmatrix} e^{-i\theta_1} & & \\ & \ddots & \\ & & e^{-i\theta_n} \end{bmatrix}, \qquad \theta_j \in [0, 2\pi), \qquad j = \overline{1, n}.$$

**Propoziţia 7.4.3** *Presupunem că matricea $A \in \mathbb{C}^{n\times n}$ are valori proprii distincte în modul, $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$. Dacă matricea $X^{-1}$ din forma normală Jordan $A = X\Lambda X^{-1}$ a lui $A$ are o descompunere LU*

$$X^{-1} = ST, \qquad S = \begin{bmatrix} 1 & & & \\ * & 1 & & \\ \vdots & \ddots & \ddots & \\ * & \ldots & * & 1 \end{bmatrix}, \qquad T = \begin{bmatrix} * & \ldots & * \\ & \ddots & \vdots \\ & & * \end{bmatrix},$$

*atunci există matricele de fază $\Theta_k$, $k \in \mathbb{N}_0$, astfel încât şirul de matrice $(\Theta_k U_k$, $k \in \mathbb{N}$ să conveargă.*

**Observaţia 7.4.4 (La propoziţia 7.4.3).** 1. Convergenţa şirului $(\Theta_k U_k)$, înseamnă mai ales, că dacă bazele ortonormale aparţinătoare converg către o bază ortonormală a lui $\mathbb{C}^n$, avem de asemenea convergenţa spaţiilor vectoriale.

2. Existenţa descompunerii LU a lui $X^{-1}$ este fără nici o limitare: deoarece $X^{-1}$ este inversabilă în mod trivial, există întotdeauna o permutare $P$, astfel încât $X^{-1}P^T = (PX)^{-1} = LU$ şi $PX$ este într-adevăr o matrice inversabilă. Aceasta înseamnă că matricea $\widehat{A} = P^T AP$, în care $A$ se transformă prin permutarea liniilor şi coloanelor are aceleşi valori proprii ca şi $A$, verifică ipoteza propoziţiei 7.4.3.

3. Demonstraţia propoziţiei 7.4.3 este o modificare a demonstraţiei din [40, pag. 54–56] pentru convergenţa metodei LR, care îşi are originea în lucrarea lui Wilkinson [1] [46]. Ce este de fapt metoda LR? Se procedează ca la metoda QR, dar lui $A$ i se aplică eliminarea gausiană, $A^{(k)} = L_k R_k$ şi apoi se construieşte $A^{(k+1)} = R_k L_k$. De asemenea, această metodă converge în anumite condiţii către o matrice triunghiulară superior. $\diamondsuit$

Înainte de a demonstra propoziţia 7.4.3 să vedem întâi de ce convergenţa şirului $(U_k)$ atrage convergenţa metodei QR. Şi anume, dacă avem $\|U_{k+1} - U_k\| \leq \varepsilon$ sau echivalent

$$U_{k+1} = U_k + E, \qquad \|E\|_2 \leq \varepsilon,$$

atunci

$$Q_k = U_{k+1}^* U_k = (U_k + E)^* U_k = I + E^* U_k = I + F, \quad \|F\|_2 \leq \|E\|_2 \underbrace{\|U_k\|_2}_{=1} \leq \varepsilon,$$

şi cu aceasta

$$A^{(k+1)} = R_k Q_k = R_k(I + F) = R_k + G, \qquad \|G\|_2 \leq \varepsilon \|R_k\|_2,$$

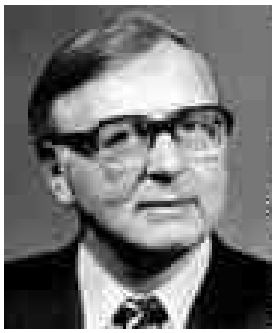
deci şirul $A^{(k)}$, $k \in \mathbb{N}$, converge, de asemenea, către o matrice triunghiulară superior, numai dacă normele lui $R_k$, $k \in \mathbb{N}$ sunt uniform mărginite. Chiar aşa se întâmplă, căci

$$\|R_k\|_2 = \|Q_k^* A^{(k)}\|_2 = \|A^{(k)}\|_2 = \|Q_{k-1}^* \dots Q_0^* A Q_0 \dots Q_{k-1}\| = \|A\|_2.$$

Mai avem nevoie de un rezultat ajutător despre ,,unicitatea" descompunerii QR.

---

[1] James Hardy Wilkinson (1919-1986), matematician englez. Contribuţii importante în domeniul Analizei numerice, Algebrei liniare numerice şi Informaticii. Membru al Royal Society, laureat al premiului Turing al ACM. Pe lângă numeroasele sale lucrări în domeniul Analizei numerice, a lucrat şi la dezvoltarea de biblioteci de rutine numerice. Grupul NAG (Numerical Algorithms Group) şi-a început activitatea în 1970 şi multe dintre rutinele de algebră liniară numerică s-au datorat lui Wilkinson.



**Lema 7.4.5** *Fie $U, V \in \mathbb{C}^{n\times n}$ matrice unitare şi fie $R, S \in \mathbb{C}^{n\times n}$ matrice triunghiulare superior inversabile. Atunci are loc $UR = VS$ dacă şi numai dacă există o matrice de fază*

$$\Theta = \begin{bmatrix} e^{-i\theta_1} & & \\ & \ddots & \\ & & e^{-i\theta_n} \end{bmatrix}, \qquad \theta_j \in [0, 2\pi), j = \overline{1, n},$$

*astfel încât $V = V\Theta^*, \quad R = \Theta S$.*

*Demonstraţie.* Deoarece $UR = V\Theta^*\Theta S = VS$, "$\Leftarrow$" este trivială. Pentru necesitate, din $UR = VS$ rezultă că $V^*U = SR^{-1}$ trebuie să fie o matrice triunghiulară superior astfel încât $(V^*U)^* = U^*V = RS^{-1}$. Aşadar $\Theta = V^*U$ este o matrice diagonală unitară şi are loc $U = VV^*U = V\Theta$. □

*Demonstraţia propoziţiei 7.4.3.* Fie $A = X\Lambda X^{-1}$ forma normală Jordan a lui $A$, unde $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. Pentru $U_0 = I$ şi $k \in \mathbb{N}_0$

$$U_k \left( \prod_{j=k-1}^{0} R_j \right) = \left(X^{-1}\Lambda X\right)^k = X\Lambda^k X^{-1} = X\Lambda^k ST = X \underbrace{(\Lambda^k S \Lambda^{-k})}_{=:L_k} \Lambda^k T,$$

în care $L_k$ este o matrice triunghiulară inferior cu elementele

$$(L_k)_{jm} = \left(\frac{\lambda_j}{\lambda_m}\right)^k, \qquad 1 \le m \le j \le n \tag{7.4.4}$$

astfel încât pentru $k \in \mathbb{N}$

$$|L_k - I| \le \left( \max_{1\le m<j\le n} |s_{jm}| \right) \left( \max_{1\le m<j\le n} \left|\frac{\lambda_j}{\lambda_m}\right| \right)^k \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ \vdots & \ddots & \ddots & \\ 1 & \ldots & 1 & 0 \end{bmatrix}, \quad (k \in \mathbb{N}). \tag{7.4.5}$$

Fie $\widehat{U}_k\widehat{R}_k = XL_k$ descompunerea $QR$ a lui $XL_k$, care datorită lui (7.4.5) şi lemei 7.4.5 converge până la o matrice de fază către descompunerea QR $X = UR$ a lui $X$. Dacă aplicăm acum lema 7.4.5 identităţii

$$U_k \left( \prod_{j=k-1}^{0} R_j \right) \widehat{Q}_k \widehat{R}_k \Lambda^k T,$$

atunci există matricele de fază $\Theta_k$, astfel încât

$$U_k = \widehat{Q}_k \Theta_k^* \text{ şi } \left( \prod_{j=k-1}^{0} R_j \right) = \Theta_k \widehat{R}_k \Lambda^k T,$$

deci există matricele de fază $\widehat{\Theta}_k$, astfel încât $U_k\widehat{\Theta}_k \to U$, când $k \to \infty$. □

Merită să aruncăm o scurtă privire asupra „termenului de eroare” din (7.4.4), ale cărui elemente subdiagonale verifică relaţia

$$|L_k|_{jm} \leq \left(\frac{|\lambda_j|}{|\lambda_m|}\right) |s_{jm}, \qquad 1 \leq m < j \leq n.$$

Aşadar are loc

*Convergenţa unui element subdiagonal către 0 este cu atât mai rapidă cu cât elementul este mai îndepărtat de diagonală.*

# 7.5. Metoda QR – practica

## 7.5.1. Metoda QR clasică

Am văzut că metoda QR generează un şir de matrice $A^{(k)}$ care în condiţii determinate trebuie să conveargă către o matrice triunghiulară superior care are pe diagonală valorile proprii. Putem întrebuinţa această metodă pentru matrice reale.

**Exemplul 7.5.1.** Fie

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{bmatrix}.$$

Această matrice are valorile proprii

$$\lambda_1 \approx 4.56155, \qquad \lambda_2 = -1, \qquad \lambda_3 \approx 0.43845.$$

Iterând prin metoda QR se obţin pentru elementele subdiagonale rezultatele din tabela 7.1 (o implementare MATLAB brută). Se poate vedea că, după $k$ iteraţii, elementele

| #iteraţii | $a_{21}$ | $a_{31}$ | $a_{32}$ |
|---|---|---|---|
| 10 | 6.64251e-007 | -2.26011e-009 | 0.00339953 |
| 20 | 1.70342e-013 | -1.52207e-019 | 8.9354e-007 |
| 30 | 4.36711e-020 | -1.02443e-029 | 2.34578e-010 |
| 40 | 1.11961e-026 | -6.89489e-040 | 6.15829e-014 |

Tabela 7.1: Rezultate pentru exemplul 7.5.1

$a_{m\ell}^{(k)}$, $\ell < m$, se apropie de 0 la fel ca $|\lambda_\ell/\lambda_k|$. ♢

**Exemplul 7.5.2.** Matricea

$$\begin{bmatrix} 1 & 5 & 7 \\ 3 & 0 & 6 \\ 4 & 3 & 1 \end{bmatrix}$$

are valorile proprii

$$\lambda_1 \approx 9.7407, \qquad \lambda_2 \approx -3.8703 + 0.6480i, \qquad \lambda_2 \approx -3.8703 - 0.6480i.$$

În acest caz nu ne putem aştepta ca metoda QR să conveargă către o matrice triunghiulară superioară, căci atunci toate valorile proprii ale lui $A$ ar fi reale. De fapt, după 100 de iteraţii se obţine matricea

$$A^{(100)} \approx \begin{bmatrix} 9.7407 & -4.3355 & 0.94726 \\ 8.552e-039 & -4.2645 & 0.7236 \\ 3.3746e-039 & -0.79491 & -3.4762 \end{bmatrix},$$

care ne furnizează corect valoarea proprie reală. Pe lângă aceasta, matricea $2 \times 2$ din colţul din dreapta jos ne furnizează valorile proprii complexe $-3.8703 \pm 0.6480i$. $\diamondsuit$

Al doilea exemplu recomandă următoarea strategie: dacă elementele situate sub diagonală nu vor să dispară, ar fi indicat să privim mai atent matricea $2 \times 2$ corespunzătoare.

**Definiţia 7.5.3** *Dacă $A \in \mathbb{R}^{n\times n}$ are descompunerea QR $A = QR$, atunci* transformarea RQ *a lui $A$ se defineşte prin $A_* = RQ$.*

Ce probleme apar la realizarea practică a metodei QR? Deoarece complexitatea descompunerii QR este $O(n^3)$, nu este prea înţelept să utilizăm o metodă care se bazează pe un astfel de pas iterativ. Pentru a evita problema, vom converti matricele iniţiale în matrice similare a căror descompunere QR poate fi calculată mai repede. Astfel de matrice sunt matricele Hessenberg superioare a căror descompunere QR se poate calcula cu $n$ iteraţii Givens, deci un total de $O(n^2)$ operaţii: de vreme ce numai elementele $h_{j-1,j}$, $j = \overline{2,n}$ trebuie eliminate, vom determina unghiurile $\phi_2, \dots \phi_n$, astfel ca

$$G(n-1, n; \phi_n) \dots G(1, 2; \phi_2)H = R$$

şi are loc

$$H_* = RG^T(1, 2; \phi_2) \dots G^T(n-1, n; \phi_n). \tag{7.5.1}$$

Această este ideea algoritmului 7.1. După [35], motto-ul trebuie să fie „odată Hessenberg, întotdeauna Hessenberg”.

**Lema 7.5.4** *Dacă $H \in \mathbb{R}^{n\times n}$ este o matrice Hessenberg superioară, atunci $H_*$ este de asemenea matrice Hessenberg superioară.*

*Demonstraţie.* Rezultă direct din reprezentarea (7.5.1). Înmulţirea la dreapta cu o matrice Givens $G^T(j, j+1, \phi_{j+1})$, $j = \overline{1, n-1}$ înseamnă o combinaţie a *coloanelor* $j$ şi $j+1$ şi crează valori diferite de zero doar în prima subdiagonală – $R$ este triunghiulară superior. $\square$

**Algoritmul 7.1** Transformarea RQ a unei matrice Hessenberg H, adică $H_* = RQ$ unde $H = QR$ este descompunerea QR a lui H

**for** $k := 1$ **to** $n-1$ **do**
$[c_k, s_k] := \mathbf{givens}(H_{kk}, H_{k+1,k})$;
$H_{k:k+1,k:n} := \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix}^T H_{k:k+1,k:n}$;
**end for**
**for** $k := 1$ **to** $n-1$ **do**
$H_{1:k+1,k:k+1} := H_{1:k+1,k:k+1} \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix}$;
**end for**

Să vedem cum aducem matricea iniţială la forma Hessenberg. În acest scop vom utiliza pentru variaţie transformările Householder. Să presupunem că am găsit deja o matrice $Q_k$, astfel încât primele $k$ coloane ale matricei transformate să aibă deja forma Hessenberg, adică

$$Q_k A Q_k^T = \left[\begin{array}{ccc|c|ccc} * & \dots & * & * & * & \dots & * \\ * & \dots & * & * & * & \dots & * \\ & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & * & * & * & \dots & * \\ \hline & & & a_1^{(k)} & * & \dots & * \\ & & & \vdots & \vdots & \ddots & \vdots \\ & & & a_{n-k-1}^{(k)} & * & \dots & * \end{array}\right].$$

Apoi determinăm $\widehat{y} \in \mathbb{R}^{n-k-1}$ şi $\alpha \in \mathbb{R}$ (care rezultă automat), astfel ca

$$H(\widehat{y}) \begin{bmatrix} a_1^{(k)} \\ \vdots \\ \vdots \\ a_{n-k-1}^{(k)} \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Rightarrow U_{k+1} := \begin{bmatrix} I_{k+1} & \\ & H(\widehat{y}) \end{bmatrix}$$

şi obţinem că

$$\underbrace{U_{k+1}Q_k}_{=:Q_{k+1}} A \underbrace{Q_k U_{k+1}}_{=Q_{k+1}^T} = \left[\begin{array}{ccc|c|ccc} * & \dots & * & * & * & \dots & * \\ * & \dots & * & * & * & \dots & * \\ & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & * & * & * & \dots & * \\ \hline & & & \alpha & * & \dots & * \\ & & & 0 & * & \dots & * \\ & & & \vdots & \ddots & \vdots & \\ & & & 0 & * & \dots & * \end{array}\right] U_{k+1};$$

matricea unitate $I_{k+1}$ din stânga sus în matricea $U_{k+1}$ are grijă să avem în primele $k+1$ coloane o structură Hessenberg. Algoritmul 7.2 dă metoda de trecere a unei matrice la forma Hessenberg. În concluzie, metoda noastră QR va fi o metodă în două faze:

1. Transformă pe $A$ în forma Hessenberg utilizând o transformare ortogonală:

$$H^{(0)} = QAQ^T, \qquad Q^TQ = QQ^T = I.$$

2. Execută iteraţiile QR

$$H^{(k+1)} = H_*^{(k)}, \qquad k \in \mathbb{N}_0,$$

în speranţa că elementele de pe prima subdiagonală converg toate către zero.

**Algoritmul 7.2** Reducere la forma Hessenberg superioară

**Intrare:** Matricea $A \in \mathbb{R}^{n\times n}$
**Ieşire:** $H$ forma Hessenberg a lui $A$ şi dacă se doreşte $Q$ astfel încât $H = QAQ^T$

```
for i := 1 to n − 2 do
    u_i := House(A_{i+1:n,i});
    P_i := I − 2u_i u_i^T; {Q_i = diag(I_i, P_i)}
    A_{i+1:n,i:n} := P_i A_{i+1:n,i:n};
    A_{1:n,i+1:n} := A_{1:n,i+1:n} P_i;
end for
if se doreşte Q then
    Q := I;
    for i := 1 to n − 2 do
        Q_{i+1:n,i+1:n} := P_i Q_{i+1:n,i+1:n};
    end for
end if
```

| $\varepsilon$ | #iteraţii | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ |
|---|---|---|---|---|
| $10^{-3}$ | 11 | 4.56155 | -0.999834 | 0.438281 |
| $10^{-4}$ | 14 | 4.56155 | -1.00001 | 0.438461 |
| $10^{-5}$ | 17 | 4.56155 | -0.999999 | 0.438446 |
| $10^{-10}$ | 31 | 4.56155 | -1 | 0.438447 |

Tabela 7.2: Rezultatele pentru exemplul 7.5.5

Deoarece elementele subdiagonale converg cel mai încet, putem folosi drept criteriu de oprire maximul modulului. Aceasta ne conduce la *metoda QR simplă*, vezi algoritmul 7.3. Desigur, la apariţia unor valori proprii complexe această metodă iterează la nesfârşit.

**Algoritmul 7.3** Metoda QR simplă

**Intrare:** Matricea $A$, toleranţa $tol$

**Ieşire:** Vectorul valorilor proprii $\lambda$ şi numărul de iteraţii $it$

$H := \mathbf{Hessenberg}(A)$; {Forma Hessenberg a lui $A$}
$it := 0$;
**while** $\|\mathrm{diag}(H, -1)\|_\infty > tol$ **do**
  $H := H_*$; {Transformarea RQ a lui $H$}
  $it := it + 1$;
**end while**
$\lambda := \mathrm{diag}(H)$;

**Exemplul 7.5.5.** Aplicăm noua metodă matricei din exemplul 7.5.1. Pentru diverse toleranţe $\varepsilon$ date obţinem rezultatele din tabela 7.2. Se observă că la fiecare trei iteraţii se câştigă o zecimală la elementele vecine cu diagonala. ♢

*Putem încerca să accelerăm metoda descompunând problema în subprobleme*. Dacă avem o matrice Hessenberg de forma

$$H = \left[\begin{array}{cccc|cccc} * & \dots & \dots & * & & & & \\ * & \ddots & & \vdots & & * & & \\ & \ddots & \ddots & \vdots & & & & \\ & & * & * & & & & \\ \hline & & & & * & \dots & \dots & * \\ & & & & * & \ddots & & \vdots \\ & & & & & \ddots & \ddots & \vdots \\ & & & & & & * & * \end{array}\right] = \begin{bmatrix} H_1 & * \\ & H_2 \end{bmatrix},$$

atunci problema de valori proprii referitoare la $H$ se poate descompune într-o problemă de valori proprii referitoare la $H_1$ şi una referitoare la $H_2$.

Conform lui [24], un element subdiagonal $h_{j+1,j}$ este considerat „suficient de mic" dacă

$$|h_{j+1,j}| \leq \text{eps}\left(|h_{jj}| + |h_{j+1,j+1}|\right). \tag{7.5.2}$$

Aici vom proceda mai simplu şi vom descompune o matrice dacă cel mai mic element în modul situat pe prima subdiagonală devine mai mic decât o toleranţă dată. Procedeul este după cum urmează: funcţia de determinare a valorilor proprii determină cu ajutorul unei iteraţii QR o descompunere în matricele $H_1$ şi $H_2$ şi se apelează recursiv pe sine pentru fiecare din aceste matrice.

Dacă una dintre matrice este $1 \times 1$, valoarea proprie este automat determinată, iar dacă este $2 \times 2$, atunci polinomul său caracteristic este

$$\begin{aligned} p_A(x) &= \det(A - xI) = x^2 - \text{trace}(A)x + \det(A) \\ &= x^2 \underbrace{(a_{11} + a_{22})}_{=:b} x + \underbrace{(a_{11}a_{22} - a_{12}a_{21})}_{=:c}. \end{aligned}$$

Dacă discriminantul $b^2 - 4c$ este pozitiv, $A$ are două valori proprii reale şi distincte

$$x_1 = \frac{1}{2}\left(-b - \text{sgn}(b)\sqrt{b^2 - 4c}\right) \text{ şi } x_2 = \frac{c}{x_1},$$

altfel valorile proprii sunt complexe şi anume

$$\frac{1}{2}\left(-b \pm i\sqrt{4c - b^2}\right);$$

astfel avem sub control şi cazul valorilor proprii complexe. Presupunem că funcţia **Eigen2x2** returnează valorile proprii ale unei matrice $2 \times 2$. Ideea este implementată în algoritmul 7.4. Iteraţiile QR efective sunt date de algoritmul 7.5.

**Exemplul 7.5.6.** Să considerăm din nou matricea din exemplul 7.5.2 căreia îi aplicăm algoritmul 7.4. Rezultatele apar în tabela 7.3. ♢

## 7.5.2. Deplasare spectrală

Utilizarea matricelor Hessenberg ne permite, într-adevăr, să executăm fiecare pas de iteraţie într-un timp mai scurt. Vom încerca acum să micşorăm numărul de iteraţii, aşadar să creştem viteza de convergenţă deoarece

*Rata de convergenţă a elementelor subdiagonale $h_{j+1,j}$ are ordinul de mărime*

$$\left(\frac{\lambda_{j+1}}{\lambda_j}\right)^k, \qquad j = \overline{1, n-1}.$$

**Algoritmul 7.4** $\mathbf{QRSplit1a}$ – metoda QR cu partiţionare şi tratarea cazurilor $2 \times 2$

**Intrare:** Matricea $A \in \mathbb{R}^{n \times n}$ şi toleranţa $tol$
**Ieşire:** Valorile proprii $\lambda$ şi numărul de iteraţii $it$

```
if n = 1 then
    it := 0; λ := A;
    return;
else
    if n = 2 then
        it := 0;
        λ := Eigen2x2(A);
        return;
    else
        H := Hessenberg(A); {Forma Hessenberg}
        [H1, H2, it] := QRIter(H, tol);
        [λ1, it1] := QRSplit1a(H1, tol);{apeluri recursive}
        [λ2, it2] := QRSplit1a(H2, tol);
        it := it + it1 + it2;
        λ := [λ1, λ2];
    end if
end if
```

**Algoritmul 7.5** Iteraţie QR pe o matrice Hessenberg; utilizat de algoritmul 7.4 – apel $[H_1, H2, it] = \mathbf{QRIter}(H, t)$

**Intrare:** Matricea Hessenberg $H$, toleranţa $tol$
**Ieşire:** Matricele $H_1$, $H_2$ reprezentând descompunerea lui $H$ după elementul subdiagonal minim în modul şi $It$ numărul de iteraţii

```
it := 0;
Determină minimul m al modulelor elementelor de pe prima subdiagonală a lui H
şi poziţia sa j;
while m > tol do
    it := it + 1;
    H := H*; {Transformarea RQ a matricei H}
    Determină minimul m al modulelor elementelor de pe prima subdiagonală a lui
    H şi poziţia sa j;
end while
H1 := H1:j,1:j;
H2 := Hj+1:n,j+1:n;
```

| $\varepsilon$ | #iteraţii | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ |
|---|---|---|---|---|
| $10^{-3}$ | 12 | 9.7406 | $-3.8703 + 0.6479i$ | $-3.8703 - 0.6479i$ |
| $10^{-4}$ | 14 | 9.7407 | $-3.8703 + 0.6479i$ | $-3.8703 - 0.6479i$ |
| $10^{-5}$ | 17 | 9.7407 | $-3.8703 + 0.6480i$ | $-3.8703 - 0.6480i$ |
| $10^{-5}$ | 19 | 9.7407 | $-3.8703 + 0.6480i$ | $-3.8703 - 0.6480i$ |
| $10^{-5}$ | 22 | 9.7407 | $-3.8703 + 0.6480i$ | $-3.8703 - 0.6480i$ |

Tabela 7.3: Rezultate pentru exemplul 7.5.6

Cuvântul de ordine este aici *deplasare (translaţie) spectrală*. Se observă că, pentru $\mu \in \mathbb{R}$, matricea $A - \mu I$ are valorile proprii $\lambda_1 - \mu, \ldots, \lambda_n - \mu$. Pentru o matrice oarecare inversabilă, $B$, matricea $B(A - \mu I)B^{-1} + \mu I$ are valorile proprii $\lambda_1, \ldots, \lambda_n$ – se poate aşadar deplasa spectrul matricei înainte şi înapoi printr-o transformare de similaritate. Se ordonează valorile proprii $\mu_1, \ldots, \mu_n$ astfel ca

$$|\mu_1 - \mu| > |\mu_2 - \mu| > \cdots > |\mu_n - \mu|, \qquad \{\mu_1, \ldots, \mu_1\} = \{\lambda_1, \ldots, \lambda_n\}$$

şi dacă $\mu$ este apropiat de $\mu_n$, atunci dacă metoda QR începe cu $H^0 = A - \mu I$, elementul subdiagonal $h^{(n)}_{n-1,n}$ converge foarte repede către zero. Mai bine este dacă deplasarea spectrală se realizează la fiecare pas individual. Pe lângă aceasta, putem lua ca aproximaţie pentru $\mu$ (în mod euristic) valoarea $h^{(k)}_{nn}$. Se obţine astfel următoarea schemă iterativă

$$H^{(k+1)} = \left(H^{(k)} - \mu_k I\right)_* + \mu_k I, \qquad \mu_k := h^{(k)}_{nn}, \quad k \in \mathbb{N}_0,$$

cu matricea de pornire $H^0 = QAQ^T$. Algoritmul 7.6 dă o variantă a metodei care tratează şi valorile proprii complexe. El utilizează algoritmul 7.7. În ultimul algoritm, $(H - H_{n,n}I_n)_*$ din linia 6 înseamnă transformarea RQ a matricei $H - H_{n,n}I_n$

**Observaţia 7.5.7.** Dacă valoarea de deplasare $\mu$ este suficient de apropiată de o valoare proprie $\lambda$, atunci matricea se descompune într-un singur pas iterativ. ♢

## 7.5.3. Metoda QR cu pas dublu

Se poate arăta că metoda QR cu deplasare spectrală converge *pătratic*, eroarea fiind pentru un $\rho < 1$ doar

$$O(\rho^{2k}) \text{ în loc de } O(\rho^k).$$

Oricât de frumoasă ar fi această idee ea funcţionează bine doar pentru valori proprii reale, în cazul valorilor proprii complexe fiind problematică. Cu toate acestea, putem exploata faptul că valorile proprii apar în perechi. Aceasta ne conduce la ideea de „metode cu pas dublu”:

**Algoritmul 7.6** Metoda QR cu deplasare spectrală, partiţionare şi tratarea valorilor proprii complexe

**Intrare:** Matricea $A \in \mathbb{R}^{n \times n}$ şi toleranţa $tol$
**Ieşire:** Valorile proprii $\lambda$ ale lui $A$ şi numărul de iteraţii $It$

$It := 0$;
**if** $n = 1$ **then**
  $\lambda := A$;
  **return**
**else if** $n = 2$ **then**
  $\lambda := \mathbf{Eigen2x2}(A)$;
  **return**
**else**
  $H := \mathbf{Hessenberg}(A)$; {aducere la forma Hessenberg}
  $[H_1, H_2, It] := \mathbf{QRIter2}(H, t)$
  $[\lambda_1, It_1] := \mathbf{QRSplit2}(H_1, tol)$ {apel recursiv}
  $[\lambda_2, It_2] := \mathbf{QRSplit2}(H_2, tol)$ {apel recursiv}
  $It := It + It_1 + It_2$;
  $\lambda = [\lambda_1, \lambda_2]$;
**end if**

**Algoritmul 7.7** Iteraţie QR şi partiţionare

$It := 0$;
Determină minimul $m$ al modulelor elementelor de pe prima subdiagonală a lui $H$ şi poziţia sa $j$;
**while** $m > tol$ **do**
  $It := It + 1$;
  $H := (H - H_{n,n} I_n)_* + H_{n,n} I_n$;
  Determină minimul $m$ al modulelor elementelor de pe prima subdiagonală a lui $H$ şi poziţia sa $j$;
**end while**
$H_1 := H_{1:j,1:j}$;
$H_2 := H_{j+1:n,j+1:n}$;

*în loc să deplasăm spectrul cu o valoare proprie, aproximată în mod euristic cu $h_{n,n}^{(k)}$, se execută două deplasări într-un pas, şi anume cu valorile proprii ale lui*

$$B = \begin{bmatrix} h_{n-1,n-1}^{(k)} & h_{n-1,n}^{(k)} \\ h_{n-1,n}^{(k)} & h_{n,n}^{(k)} \end{bmatrix}.$$

Există două posibilităţi: fie ambele valori $\mu$ şi $\mu'$ ale lui $B$ sunt reale şi atunci se procedează ca mai sus, fie sunt complexe conjugate şi avem valorile proprii $\mu$ şi $\bar{\mu}$. Aşa cum vom vedea, al doilea caz se poate de asemenea trata în aritmetica reală. Fie $Q_k$, $Q'_k \in \mathbb{C}^{n\times n}$ şi $R_k$, $R'_k \in \mathbb{C}^{n\times n}$ matricele descompunerii QR complexe

$$\begin{aligned} Q_k R_k &= H^{(k)} - \mu I, \\ Q'_k R'_k &= R_k Q_k + (\mu - \bar{\mu}) I. \end{aligned}$$

Atunci are loc

$$\begin{aligned} H^{(k+1)} &:= R'_k Q'_k + \mu I = (Q'_k)^*(R_k Q_k + (\mu - \bar{\mu})I)Q'_k + \bar{\mu} I \\ &= Q_k^* R_k Q_k Q'_k + \mu I = (Q'_k)^* Q_k^* (H^{(k)} - \mu I) Q_k Q'_k + \mu I \\ &= \underbrace{(Q_k Q'_k)^*}_{=U^*} H^{(k)} \underbrace{Q_k Q'_k}_{=U}. \end{aligned}$$

Utilizând matricea $S = R'_k R_k$ avem

$$\begin{aligned} US &= Q_k Q'_k R'_k R_k = Q_k (R_k Q_k + (\mu - \bar{\mu})I) R_k \\ &= Q_k R_k Q_k R_k + (\mu - \bar{\mu}) Q_k R_k = (H^{(k)} - \mu I)^2 + (\mu - \bar{\mu})(H^{(k)} - \mu I) \\ &= (H^{(k)})^2 - 2\mu H^{(k)} + \mu^2 I + (\mu - \bar{\mu}) H^{(k)} - (\mu^2 - \mu\bar{\mu}) I \\ &= (H^{(k)})^2 - (\mu + \bar{\mu}) H^{(k)} + \mu\bar{\mu} I =: X \end{aligned} \tag{7.5.3}$$

Dacă $\mu = \alpha + i\beta$, atunci $\mu + \bar{\mu} = 2\alpha$ şi $\mu\bar{\mu} = |\mu|^2 = \alpha^2 + \beta^2$, aşadar matricea $X$ din membrul drept al lui (7.5.3), deci are o descompunere QR $X = QR$ reală şi conform lemei 7.4.5 există o matrice de fază $\Theta \in \mathbb{C}^{n\times n}$ astfel încât $U = \Theta Q$. Dacă vom itera în real mai departe, vom obţine *metoda QR cu pas dublu*

$$\begin{aligned} Q_k R_k &= (H^{(k)})^2 - (h_{n-1,n-1}^{(k)} + h_{n,n}^{(k)}) H^{(k)} \\ &\quad + \left( (h_{n-1,n-1}^{(k)} h_{n,n}^{(k)} - h_{n-1,n}^{(k)} H_{n,n-1}^{(k)} \right) I, \\ H^{(k+1)} &= Q_k^T H^{(k)} Q_k. \end{aligned} \tag{7.5.4}$$

**Observaţia 7.5.8 (Metoda QR cu pas dublu).** 1. Matricea $X$ din (7.5.3) nu mai este o matrice Hessenberg, ea având o diagonală suplimentară. Cu toate acestea se poate calcula descompunerea QR a lui $X$ destul de simplu, cu doar $2n-3$ rotaţii Jacobi, în loc de $n-1$, cât este necesar pentru o matrice Hessenberg.

2. Datorită complexităţii sale ridicate, înmulţirea $Q_k^T H^{(k)} Q_k$ nu mai este o metodă efectivă de iteraţie; acest lucru se poate remedia, a se vedea de exemplu [18] sau [36, pag. 272–278].

3. Natural, $H^{(k+1)}$ poate fi adusă la forma Hessenberg.

4. Metoda cu dublu pas se aplică numai atunci când matricea $A$ are valori proprii complexe; din contră, la matrice simetrice nu este avantajoasă. ♢

Metoda QR cu pas dublu, partiţionare şi tratarea matricelor $2 \times 2$ este dată în algoritmul 7.8. El apelează algoritmul 7.9.

**Algoritmul 7.8** Metoda QR cu dublu pas, partiţionare şi tratarea matricelor $2 \times 2$

**Intrare:** Matricea $A \in \mathbb{R}^{n \times n}$ şi toleranţa $tol$
**Ieşire:** Valorile proprii $\lambda$ ale lui $A$ şi numărul de iteraţii $It$

$It := 0$;
**if** $n = 1$ **then**
  $\lambda := A$;
  **return**
**else if** $n = 2$ **then**
  $\lambda := \mathbf{Eigen2x2}(A)$;
  **return**
**else**
  $H := \mathbf{Hessenberg}(A)$; {aducere la forma Hessenberg}
  $[H_1, H_2, It] := \mathbf{QRDouble}(H, t)$
  $[\lambda_1, It_1] := \mathbf{QRSplit2}(H_1, tol)$ {apel recursiv}
  $[\lambda_2, It_2] := \mathbf{QRSplit2}(H_2, tol)$ {apel recursiv}
  $It := It + It_1 + It_2$;
  $\lambda = [\lambda_1, \lambda_2]$;
**end if**

**Exemplul 7.5.9.** Aplicăm algoritmii 7.6 şi 7.8 matricelor din exemplele 7.5.1 şi 7.5.2. Se obţin rezultatele din tabela 7.4. Buna comportare a metodei pasului dublu se justifică prin aceea că obţine două valori proprii dintr-o dată. ♢

**Algoritmul 7.9** Iteraţie QR cu dublu pas şi transformare Hessenberg

$It := 0$;
Determină minimul $m$ al modulelor elementelor de pe prima subdiagonală a lui $H$ şi poziţia sa $j$;
**while** $m > tol$ **do**
  $It := It + 1$;
  $X := H^2 - (H_{n-1,n-1} + H_{n,n})H + (H_{n-1,n-1}H_{n,n} - H_{n,n-1}H_{n-1,n})I_n$;
  Determină descompunerea $X = QR$ a lui $X$;
  $H := \mathbf{Hessenberg}(Q^T HQ)$;
  Determină minimul $m$ al modulelor elementelor de pe prima subdiagonală a lui $H$ şi poziţia sa $j$;
**end while**
$H_1 := H_{1:j,1:j}$;
$H_2 := H_{j+1:n,j+1:n}$;

| | #iteraţii în real | | #iteraţii în complex | |
|---|---|---|---|---|
| $\varepsilon$ | alg. 7.6 | alg. 7.8 | alg. 7.6 | alg. 7.8 |
| 1e-010 | 1 | 1 | 9 | 4 |
| 1e-020 | 9 | 2 | 17 | 5 |
| 1e-030 | 26 | 3 | 45 | 5 |

Tabela 7.4: Comparaţiile din exemplul 7.5.9

CAPITOLUL 8

# Rezolvarea numerică a ecuaţiilor diferenţiale ordinare

## Cuprins

# 8.1. Ecuaţii diferenţiale

Considerăm problema cu valori iniţiale sau problema Cauchy [1]: să se determine o funcţie cu valori vectoriale $y \in C^1[a,b]$, $y : [a,b] \to \mathbb{R}^d$, astfel încât

$$(PC) \quad \begin{cases} \dfrac{dy}{dx} = f(x,y),\ x \in [a,b] \\ y(a) = y_0 \end{cases} \tag{8.1.1}$$

Vom evidenţia două clase importante de astfel de probleme:

(i) pentru $d = 1$ avem o singură ecuţie diferenţială scalară de ordinul I

$$\begin{cases} y' = f(x,y) \\ y(a) = y_0 \end{cases}$$

(ii) pentru $d > 1$ avem un sistem de ecuaţii diferenţiale ordinare de ordinul I

$$\begin{cases} \frac{dy^i}{dx} = f^i(x, y^1, y^2, \ldots, y^d), \quad i = \overline{1,d} \\ y^i(a) = y_0^i,\ i = \overline{1,d} \end{cases}$$

Reamintim următoarea teoremă clasică referitoare la existenţă şi unicitate.

**Teorema 8.1.1** *Presupunem că $f(x,y)$ este continuă în prima variabilă pentru $x \in [a,b]$ şi în raport cu cea de-a doua variabilă satisface o condiţie Lipschitz uniformă*

$$\|f(x,y) - f(x,y^*)\| \le L\|y - y^*\|, \quad y, y^* \in \mathbb{R}^d, \tag{8.1.2}$$

*unde $\|\cdot\|$ este o anumită normă vectorială. Atunci problema Cauchy (PC) are o soluţie unică $y(x)$, $a \le x \le b$, $\forall\, y_0 \in \mathbb{R}^d$. Mai mult, $y(x)$ depinde continuu de $a$ şi $y_0$.*

[1] Augustin Louis Cauchy (1789-1857), matematician francez, considerat părintele analizei moderne. A fundamentat solid analiza pe baza conceptului riguros de limită. Este de asemenea creatorul analizei complexe, în care „formula lui Cauchy" ocupă un loc central. Numele său este legat şi de contribuţii de pionierat în domeniul ecuaţiilor diferenţiale şi cu derivate parţiale, în particular legate de problema existenţei şi unicităţii. La fel ca în cazul multor mari matematicieni din secolelele al optsprezecelea şi al nouasprezecelea, lucrările sale au tratat probleme din geometrie, algebră, teoria numerelor, mecanică, dar şi fizică teoretică.

Condiţia Lipschitz (8.1.2) are singur loc dacă toate funcţiile $\frac{\partial f^i}{\partial y^j}(x,y)$, $i,j = \overline{1,d}$ sunt continue în raport cu variabilele $y$ şi sunt mărginite pe $[a,b] \times \mathbb{R}^d$. Aceasta este situaţia în cazul sistemelor de ecuaţii diferenţiale ordinare liniare, unde

$$f^i(x,y) = \sum_{j=1}^{d} a_{ij}(x)y^j + b_i(x), \quad i = \overline{1,d}$$

şi $a_{ij}(x)$, $b_i(x)$ sunt funcţii continue pe $[a,b]$.

De multe ori condiţia Lipschitz (8.1.2) are loc într-o vecinătate a lui $x_0$ astfel încât $y(x)$ să rămână într-un compact $D$.

## 8.2. Metode numerice

Se face distincţie între *metode de aproximare analitice* şi *metode discrete*. În cadrul primei categorii se încearcă să se găsească aproximaţii $y_a(x) \approx y(x)$ ale soluţiei exacte, valabile pentru orice $x \in [a,b]$. Acestea de obicei au forma unei dezvoltări într-o serie trunchiată, fie după puterile lui $x$, fie în polinoame Cebâşev, fie într-un alt sistem de funcţii de bază. În cazul metodelor discrete, se încearcă să se găsească aproximaţii $u_n \in \mathbb{R}^d$ ale lui $y(x_n)$ pe o grilă de puncte $x_n \in [a,b]$. Abscisele $x_n$ pot fi predeterminate (de exemplu puncte echidistante pe $[a,b]$), sau mai convenabil sunt generate dinamic ca parte a procesului de integrare.

Dacă se doreşte, se pot obţine din aceste aproximante discrete $\{u_n\}$ aproximante $y_n(x)$ definite pe întreg intervalul $[a,b]$, fie prin interpolare, sau mai natural, printr-un mecanism de continuare conţinut în metoda de aproximare însăşi. Ne vom ocupa numai de metode discrete cu un pas, adică metode în care $u_{n+1}$ este determinat cunoscând numai $x_n, u_n$ şi pasul $h$ pentru a trece de la $x_n$ la $x_{n+1} = x_n + h$. Într-o metodă cu $k$ paşi $(k > 1)$ este necesară cunoaşterea a $k-1$ puncte adiţionale $(x_{n-j}, u_{n-j})$, $j = 1, 2, \ldots, k-1$ pentru a obţine o nouă componentă a soluţiei.

Când se descrie o metodă cu un pas este suficient să arătăm cum se trece de la un punct generic $(x,y)$, $x \in [a,b]$, $y \in \mathbb{R}^d$ la punctul următor $(x+h, y_{next})$. Ne vom referi la aceasta ca fiind *descrierea locală* a unei metode cu un pas. Aceasta include de asemenea o discuţie a preciziei locale, adică cât de apropiat este $y_{next}$ de soluţie în $x+h$. O metodă cu un pas pentru rezolvarea problemei Cauchy (8.1.1) generează efectiv o funcţie grilă $\{u_n\}$, $u_n \in \mathbb{R}^d$, pe o grilă $a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b$ ce acoperă intervalul $[a,b]$, prin care se intenţionează ca $u_n$ să aproximeze soluţia exactă $y(x)$ în $x = x_n$. Punctul $(x_{n+1}, u_{n+1})$ se obţine din punctul $(x_n, u_n)$ aplicând o metodă cu un pas, având un pas $h_n = x_{n+1} - x_n$ adecvat ales. Ne vom referi la aceasta ca *descrierea globală* a unei metode cu un pas. Chestiunile de interes aici sunt comportarea erorii globale $u_n - y(x_n)$, în particular stabilitatea, convergenţa şi alegerea lui $h_n$ pentru a trece de la un punct al grilei $x_n$, la următorul, $x_{n+1} = x_n + h_n$.

## 8.3. Descrierea locală a metodelor cu un pas

Dându-se un punct generic $x \in [a,b]$, $y \in \mathbb{R}^d$, definim un pas al metodei cu un pas prin

$$y_{next} = y + h\Phi(x,y;h), \quad h > 0. \tag{8.3.1}$$

Funcţia $\Phi : [a,b] \times \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ poate fi gândită ca un increment aproximativ pe unitatea de pas sau ca o aproximare a diferenţei divizate şi ea defineşte metoda. Împreună cu (8.3.1) considerăm soluţia $u(t)$ a ecuaţiei diferenţiale (8.1.1) ce trece prin punctul $(x,y)$, adică problema locală cu valoarea iniţială

$$\begin{cases} \frac{du}{dt} = f(t,u) \\ u(t) = y \quad t \in [t, t+h] \end{cases} \tag{8.3.2}$$

Vom numi $u(t)$ *soluţie de referinţă*. Se intenţionează ca vectorul $y_{next}$ din (8.3.1) să aproximeze $u(x+h)$. Cu cât succes se realizează aceasta se măsoară prin eroarea locală de trunchiere, definită după cum urmează.

**Definiţia 8.3.1** Eroarea de trunchiere *a metodei* $\Phi$ *în punctul* $(x,y)$ *este definită prin*

$$T(x,y;h) = \frac{1}{h}[y_{next} - u(x+h)]. \tag{8.3.3}$$

Eroarea de trunchiere este o funcţie cu valori vectoriale de $d+2$ variabile. Utilizând (8.3.1) şi (8.3.2) o putem scrie sub forma

$$T(x,y;h) = \Phi(x,y;h) - \frac{1}{h}[u(x+h) - u(x)], \tag{8.3.4}$$

ceea ce arată că $T$ este diferenţa între incrementul aproximativ şi cel exact pe unitatea de pas.

**Definiţia 8.3.2** *Metoda* $\Phi$ *se numeşte* consistentă *dacă*

$$T(x,y;h) \to 0 \text{ } \textit{când} \text{ } h \to 0, \tag{8.3.5}$$

*uniform pentru* $(x,y) \in [a,b] \times \mathbb{R}^d$.

Conform lui (8.3.4) şi (8.3.3) avem consistenţă dacă şi numai dacă

$$\Phi(x,y;0) = f(x,y), \quad x \in [a,b], \quad y \in \mathbb{R}^d. \tag{8.3.6}$$

O descriere mai fină a preciziei locale este furnizată de definiţia următoare, bazată pe conceptul de eroare de trunchiere.

**Definiţia 8.3.3** *Spunem că metoda $\Phi$ are* ordinul $p$ *dacă pentru o anumită normă vectorială $\|\cdot\|$*

$$\|T(x,y;h)\| \leq Ch^p, \tag{8.3.7}$$

*uniform pe $[a,b]\times\mathbb{R}^d$, cu constanta $C$ independentă de $x, y$ şi $h$.*

Această proprietate se mai poate exprima sub forma

$$T(x,y;h) = O(h^p), \quad h \to 0. \tag{8.3.8}$$

De notat că $p > 0$ implică consistenţa. De obicei $p \in \mathbb{N}^*$. El se numeşte *ordin exact,* dacă (8.3.7) nu are loc pentru nici un $p$ mai mare.

**Definiţia 8.3.4** *O funcţie $\tau : [a,b]\times\mathbb{R}^d \to \mathbb{R}^d$ care satisface $\tau(x,y) \not\equiv 0$ şi*

$$T(x,y;h) = \tau(x,y)h^p + O(h^{p+1}), \quad h \to 0 \tag{8.3.9}$$

*se numeşte* funcţie de eroare principală .

Funcţia de eroare principală determină termenul principal (dominant) al erorii de trunchiere. Numărul $p$ din (8.3.9) este ordinul exact al metodei deoarece $\tau \not\equiv 0$.

Toate definiţiile precedente sunt formulate în ideea că $h > 0$ este un număr mic. Cu cât $p$ este mai mare, cu atât metoda este mai precisă.

# 8.4. Exemple de metode cu un pas

Unele dintre metodele cele mai vechi sunt motivate prin consideraţii geometrice simple asupra pantei definite de membrul drept al ecuaţiei diferenţiale. În această categorie intră metoda lui Euler şi metoda lui Euler modificată. Alte metode mai precise şi mai sofisticate se bazează pe dezvoltarea Taylor.

## 8.4.1. Metoda lui Euler

Euler a propus metoda sa în 1768, la începutul istoriei calculului diferenţial şi integral. Ea constă pur şi simplu în a urma panta în punctul generic $(x,y)$ pe un interval de lungime $h$

$$y_{next} = y + hf(x,y). \tag{8.4.1}$$

(vezi figura 8.1).

Astfel $\Phi(x,y;h) = f(x,y)$ nu depinde de $h$ şi conform lui (8.3.6) metoda este consistentă. Pentru eroarea de trunchiere avem conform lui (8.3.3)

$$T(x,y;h) = f(x,y) - \frac{1}{h}[u(x+h) - u(x)], \tag{8.4.2}$$

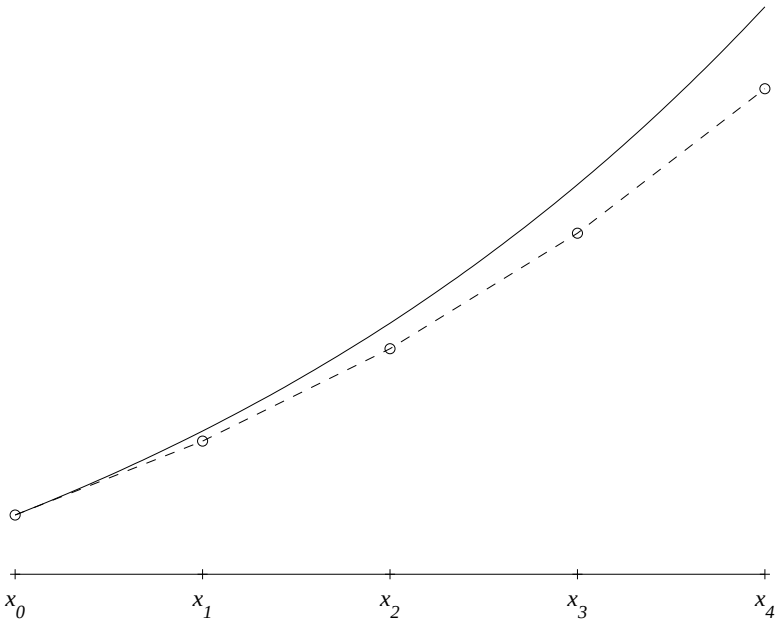



Figura 8.1: Metoda lui Euler – soluţia exactă (linie continuă) şi soluţia aproximativă (linie punctată)

unde $u(t)$ este soluţia de referinţă definită de (8.3.2). Deoarece $u' = f(x, u(x)) = f(x, y)$, putem scrie, utilizând formula lui Taylor

$$\begin{aligned} T(x, y; h) &= u'(x) - \frac{1}{h}[u(x+h) - u(x)] = \\ &= u'(x) - \frac{1}{h}[u(x) + hu'(x) + \frac{1}{2}h^2u''(\xi) - u(x)] = \\ &= -\frac{1}{2}hu''(\xi), \quad \xi \in (x, x+h), \end{aligned} \tag{8.4.3}$$

presupunând că $u \in C^r[x, x+h]$. Aceasta este adevărată dacă $f \in C^1([a, b] \times \mathbb{R}^d)$. Diferenţiind acum total (8.3.2) în raport cu $t$ şi făcând $t = \xi$, suntem conduşi la

$$T(x, y; h) = -\frac{1}{2}h[f_x + f_y f](\xi, u(\xi)), \tag{8.4.4}$$

unde $f_x$ este derivata parţială a lui $f$ în raport cu $x$ şi $f_y$ este jacobianul lui $f$ în raport cu variabila $y$. Dacă în spiritul teoremei 8.1.1, presupunem că $f$ şi toate derivatele sale parţiale de ordinul I sunt uniform mărginite în $[a, b] \times \mathbb{R}^d$, există o constantă $C$, independentă de $x, y$ şi $h$ astfel încât

$$\|T(x, y; h)\| \leq Ch. \tag{8.4.5}$$

Astfel, metoda lui Euler are ordinul $p = 1$. Dacă facem aceeaşi presupunere şi despre derivatele parţiale de ordinul doi ale lui $f$ avem

$$u''(\xi) = u''(x) + O(h)$$

şi de aceea din (8.4.3) rezultă

$$T(x, y; h) = -\frac{1}{2}h[f_x + f_y f](x, y) + O(h^2), \quad h \to 0, \tag{8.4.6}$$

arătând că funcţia eroare principală este dată de

$$\tau(x, y) = -\frac{1}{2}[f_x + f_y f](x, y). \tag{8.4.7}$$

Exceptând situaţia când $f_x + f_y f \equiv 0$, ordinul exact al metodei lui Euler este $p = 1$.

## 8.4.2. Metoda dezvoltării Taylor

Am văzut că metoda lui Euler se bazează pe trunchierea dezvoltării Taylor a soluţiei de referinţă după cel de-al doilea termen. Este o idee naturală, propusă încă de Euler, de a utiliza mai mulţi termeni din dezvoltarea Taylor. Aceasta necesită calculul succesiv al „derivatelor totale“ ale lui $f$,

$$\begin{aligned} &f^{[0]}(x, y) = f(x, y) \\ &f^{[k+1]}(x, y) = f_x^{[k]}(x, y) + f_y^{[k]}(x, y) f(x, y), \quad k = 0, 1, 2, \ldots \end{aligned} \tag{8.4.8}$$

care determină derivatele succesive ale soluţiei de referinţă $u(t)$ a lui (8.3.2) în virtutea relaţiei

$$u^{(k+1)}(t) = f^{[k]}(t, u(t)), \quad k = 0, 1, 2, \ldots \tag{8.4.9}$$

Acestea, pentru $t = x$ devin

$$u^{(k+1)}(x) = f^{[k]}(x, y), \quad k = 0, 1, 2, \ldots \tag{8.4.10}$$

şi sunt utilizate pentru a scrie dezvoltarea Taylor conform cu

$$y_{next} = y + h\left[f^{[0]}(x, y) + \frac{1}{2}hf^{[1]}(x, y) + \cdots + \frac{1}{p!}h^{p-1}f^{[p-1]}(x, y)\right], \tag{8.4.11}$$

adică

$$\Phi(x, y; h) = f^{[0]}(x, y) + \frac{1}{2}hf^{[1]}(x, y) + \cdots + \frac{1}{p!}h^{p-1}f^{[p-1]}(x, y). \tag{8.4.12}$$

Pentru eroarea de trunchiere, utilizând (8.4.10) şi (8.4.12) şi presupunând că $f \in C^p([a,b] \times \mathbb{R}^d)$ se obţine din teorema lui Taylor

$$\begin{aligned} T(x,y;h) &= \Phi(x,y;h) - \frac{1}{h}[u(x+h) - u(x)] = \\ &= \Phi(x,y;h) - \sum_{k=0}^{p-1} u^{(k+1)}(x)\frac{h^k}{(k+1)!} - u^{(p+1)}(\xi)\frac{h^p}{(p+1)!} = \\ &= -u^{(p+1)}(\xi)\frac{h^p}{(p+1)!}, \quad \xi \in (x, x+h), \end{aligned}$$

aşa că

$$\|T(x,y;h)\| \leq \frac{C_p}{(p+1)!}h^p,$$

unde $C_p$ este o margine a derivatei totale de ordin $p$ a lui $f$. Astfel metoda are ordinul exact $p$ (în afară de cazul când $f^{[p]}(x,y) \equiv 0$) şi funcţia eroare principală este

$$\tau(x,y) = -\frac{1}{(p+1)!}f^{[p]}(x,y). \tag{8.4.13}$$

Necesitatea calculului multor derivate parţiale în (8.4.8) a fost un factor descurajant în trecut, când se făcea cu mâna. Dar în zilele noastre această sarcină poate cădea în seama calculatorului, astfel încât metoda a devenit din nou o opţiune viabilă.

### 8.4.3. Metode de tip Euler îmbunătăţite

Există prea multă inerţie în metoda lui Euler: nu se urmează aceeaşi pantă pe întreg intervalul de lungime $h$, deoarece de-a lungul acestui segment de dreaptă panta definită de ecuaţia diferenţială se schimbă. Aceasta sugerează mai multe alternative. De exemplu am putea să reevaluăm panta la mijlocul segmentului – să luăm pulsul ecuaţiei diferenţiale – şi apoi să urmăm panta actualizată pe întreg intervalul (vezi figura 8.2). Formula este

$$y_{next} = y + hf\left(x + \frac{1}{2}h, y + \frac{1}{2}hf(x,y)\right) \tag{8.4.14}$$

sau

$$\Phi(x,y;h) = f\left(x + \frac{1}{2}h, y + \frac{1}{2}hf(x,y)\right) \tag{8.4.15}$$

Observaţi „imbricarea” necesară aici. Pentru programarea acestei metode este indicat să se scrie

$$\begin{aligned} &K_1(x,y) = f(x,y) \\ &K_2(x,y;h) = f\left(x + \tfrac{1}{2}h, u + \tfrac{1}{2}hK_1\right) \\ &y_{next} = y + hK_2 \end{aligned} \tag{8.4.16}$$

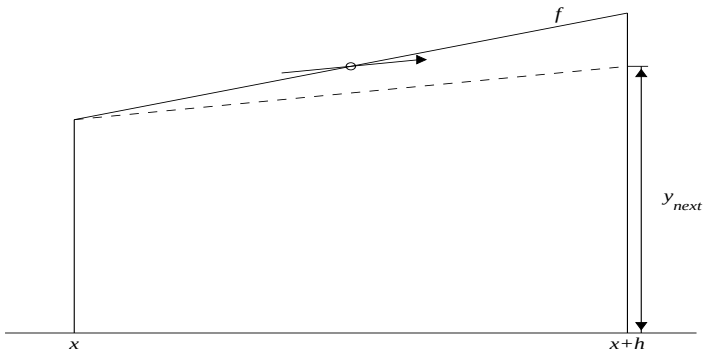



Figura 8.2: Metoda lui Euler modificată

Cu alte cuvinte, încercăm să luăm două pante de test $K_1$ şi $K_2$, una în punctul iniţial şi a doua în apropiere şi apoi s-o alegem pe ultima ca pantă finală. Metoda se numeşte *metoda lui Euler modificată* .

Putem la fel de bine să luăm o a doua pantă de încercare $(x+h, y+hf(x,y))$, dar atunci, deoarece trebuie să aşteptăm prea mult înainte de a reevalua panta, luăm ca pantă finală media celor două pante

$$\begin{aligned} K_1(x,y) &= f(x,y) \\ K_2(x,y;h) &= f(x+h, y+hK_1) \\ y_{next} &= y + \frac{1}{2}h(K_1+K_2). \end{aligned} \tag{8.4.17}$$

Această metodă se numeşte *metoda lui Heun*. Efectul ambelor modificări este creşterea ordinului cu 1, aşa cum se va vedea în continuare.

# 8.5. Metode Runge-Kutta

Se caută $\Phi$ de forma:

$$\begin{aligned} \Phi(x,y;h) &= \sum_{s=1}^{r} \alpha_s K_s \\ K_1(x,y) &= f(x,y) \\ K_s(x,y) &= f\left(x+\mu_s h, y+h\sum_{j=1}^{s-1}\lambda_{sj}K_j\right), \qquad s=2,3,\ldots,r \end{aligned} \tag{8.5.1}$$

Este natural să impunem în (8.5.1) condiţiile

$$\mu_s = \sum_{j=1}^{s-1} \lambda_{sj}, \quad s = 2, 3, \ldots, r, \quad \sum_{s=1}^{r} \alpha_s = 1, \tag{8.5.2}$$

unde primul set de condiţii este echivalent cu

$$K_s(x, y; h) = u'(x + \mu_s h) + O(h^2), \quad s \geq 2,$$

iar a doua este condiţia de consistenţă (8.3.6) (adică $\Phi(x, y; h) = f(x, y)$).

Vom numi metoda (8.5.1) *metodă Runge-Kutta explicită în $r$ stadii* deoarece necesită $r$ evaluări ale funcţiei $f$ din membrul drept al ecuaţiei diferenţiale. Condiţiile (8.5.2) conduc la un sistem neliniar. Fie $p^*(r)$ ordinul maxim pentru o metodă Runge-Kutta explicită în $r$ stadii. Kutta [2] a arătat în 1901 că

$$p^*(r) = r, \quad r = \overline{1, 4.}$$

Se pot considera *metode Runge-Kutta implicite* cu $r$ stadii

$$\begin{aligned} &\Phi(x, y; h) = \sum_{s=1}^{r} \alpha_s K_s(x, y; h), \\ &K_s = f\left(x + \mu_s h, y + \sum_{j=1}^{r} \lambda_{sj} K_j\right), \qquad s = \overline{1, r}, \end{aligned} \tag{8.5.3}$$

în care ultimele $r$ ecuaţii formează un sistem de ecuaţii (în general neliniar) cu necunoscutele $K_1$, $K_2$, ..., $K_r$. Deoarece fiecare dintre necunoscute este un vector din $\mathbb{R}^d$, înainte de construirea incrementului aproximativ $\Phi$ trebuie să rezolvăm un sistem de $rd$ ecuaţii cu $rd$ necunoscute. *Metodele Runge-Kutta semiimplicite*, la care limitele de sumare merg de la $j = 1$ la $j = s$ necesită un efort mai mic. Se ajunge la un sistem de $r$ ecuaţii, fiecare având $d$ necunoscute, componentele lui $K_s$. Volumul considerabil de calcule necesar în metodele implicite şi semiimplicite se justifică numai în împrejurări

[2] Wilhelm Martin Kutta (1867-1944), matematician german, cu preocupări în domeniul matematicilor aplicate. Cunoscut pentru lucrările sale în domeniul rezolvării numerice a ecuaţiilor diferenţiale ordinare, a avut şi contribuţii al aplicarea transformărilor conforme la probleme de hidro şi aerodinamică (formula Kutta-Jukovski).

speciale, de exemplu la rezolvarea problemelor stiff. Motivul este acela că metodele implicite pot avea ordin mai mare şi proprietăţi de stabilitate mai bune.

Parametrii se aleg astfel ca ordinul să fie cât mai mare posibil.

**Exemplul 8.5.1.** Fie

$$\Phi(x, y; h) = \alpha_1 K_1 + \alpha_2 K_2 \qquad \diamondsuit$$

unde

$$\begin{aligned} K_1(x, y) &= f(x, y) \\ K_2(x, y; h) &= f(x + \mu_2 h, y + \lambda_{21} h K_1) \\ \lambda_{21} &= \mu_2 \end{aligned}$$

Avem deci 3 parametri $\alpha_1$, $\alpha_2$, $\mu$. Un mod sistematic de a determina ordinul maxim $p$ este de a dezvolta atât $\Phi(x, y; h)$ cât şi $h^{-1}[u(x + h) - u(x)]$ după puterile lui $h$ şi să impunem coincidenţa a cât mai multor termeni posibili, fără a impune restricţii asupra lui $f$. Pentru a dezvolta $\Phi$ avem nevoie de dezvoltarea Taylor a unei funcţii vectoriale de mai multe variabile

$$\begin{aligned} f(x + \Delta x, y + \Delta y) = f + f_x \Delta x + f_y \Delta y + \\ + \frac{1}{2}[f_{xx}(\Delta x)^2 + 2 f_{xy} \Delta x \Delta y + (\Delta y)^T f_{yy} (\Delta y)] + \cdots , \end{aligned} \tag{8.5.4}$$

unde $f_y$ este jacobianul lui $f$, iar $f_{yy} = [f^i_{yy}]$ este vectorul matricelor hessiene ale lui $f$. În formula de mai sus toate funcţiile şi derivatele parţiale se evaluează în $(x, y)$. Punând $\Delta x = \mu h$, $\Delta y = \mu h f$ obţinem

$$\begin{aligned} K_2(x, y; h) = f + \mu h (f_x + f_y f) \\ + \frac{1}{2} \mu^2 h^2 (f_{xx} + 2 f_{xy} f + f^T f_{yy} f) + O(h^3) \end{aligned} \tag{8.5.5}$$

$$\frac{1}{h}[u(x + h) - u(x)] = u'(x) + \frac{1}{2} h u''(x) + \frac{1}{6} u'''(x) + O(h^3) \tag{8.5.6}$$

unde

$$\begin{aligned} u'(x) &= f \\ u''(x) &= f^{[1]} = f_x + f_y f \\ u'''(x) &= f^{[2]} = f^{[1]}_x + f^{[1]}_y f = f_{xx} + f_x f_y f + f_y f_x + (f_{xy} + (f_y f)_y) f = \\ &= f_{xx} + 2 f_{xy} f + f^T f_{yy} f + f_y (f_x + f_y) f \end{aligned}$$

unde în ultima ecuaţie s-a utilizat

$$(f_y f)_y f = f^T f_{yy} f + f_y^2 f$$

Avem

$$T(x,y;h)=\alpha_1K_1+\alpha_2K_2-\frac{1}{h}[u(x+h)-u(x)]$$

în care înlocuim (8.5.5) şi (8.5.6). Găsim

$$T(x,y;h)=(\alpha_1+\alpha_2-1)f+\left(\alpha_2\mu-\frac{1}{2}\right)h(f_x+f_yf)+$$
$$+\frac{1}{2}h^2\left[\left(\alpha_2\mu^2-\frac{1}{3}\right)(f_{xx}+2f_{xy}f+f^Tf_{yy}f)-\frac{1}{3}f_y(f_x+f_yf)\right]+O(h^3) \quad (8.5.7)$$

Nu putem impune asupra coeficientul lui $h^2$ condiţia ca el să fie zero decât dacă impunem restricţii severe asupra lui $f$. Astfel ordinul maxim este 2 şi el se obţine pentru

$$\begin{cases}\alpha_1+\alpha_2=1\\ \alpha_2\mu=\frac{1}{2}\end{cases}$$

Soluţia

$$\alpha_1=1-\alpha_2$$
$$\mu=\frac{1}{2\alpha_2}$$

depinde de un parametru, $\alpha_2\neq 0$, arbitrar.

Pentru $\alpha_2=1$ avem metoda lui Euler modificată, iar pentru $\alpha_2=\frac{1}{2}$ metoda lui Heun.

Vom menţiona formula Runge-Kutta clasică de ordin $p=4$.

$$\begin{array}{l}\Phi(x,y;h)=\frac{1}{6}(K_1+2K_2+2K_3+K_4)\\ K_1(x,y;h)=f(x,y)\\ K_2(x,y;h)=f\left(x+\frac{1}{2}h,y+\frac{1}{2}hK_1\right)\\ K_3(x,y;h)=f\left(x+\frac{1}{2}h,y+\frac{1}{2}hK_2\right)\\ K_4(x,y;h)=f(x+h,y+hK_3)\end{array} \quad (8.5.8)$$

Dacă $f$ nu depinde de $y$, atunci (8.5.8) coincide cu formula lui Simpson. Runge [3] a avut ideea de a generaliza formula lui Simpson la ecuaţii diferenţiale ordinare. El a reuşit

[3] Carle David Tolmé Runge (1856-1927) matematician german, membru al şcolii matematice de la Götingen şi unul dintre pionierii matematicii numerice. Este cunoscut pentru metodele Runge-Kutta din domeniul rezolvării numerice a ecuaţiilor diferenţiale ordinare, ale căror idei de bază i se datorează. A avut contribuţii notabile şi în domeniul aproximărilor în planul complex.

doar parţial, formula sa având $r = 4$ şi $p = 3$. Metoda (8.5.8) a fost descoperită de Kutta în 1901 printr-o căutare sistematică.

În cazul când $f$ nu depinde de $y$, atunci (8.5.8) se reduce la formula lui Simpson. Metoda Runge-Kutta clasică de ordinul 4 pentru o grilă de $N + 1$ puncte echidistante este dată de algoritmul 8.1.

**Algoritmul 8.1** Metoda Runge-Kutta de ordinul 4

**Intrare:** capetele $a, b$ ale intervalului; întregul $N$; valoarea iniţială $\alpha$.
**Ieşire:** $N + 1$ abscise $t$ şi aproximantele $w$ ale lui valorilor lui $y$ în $t$.

$h := (b - a)/N$;
$t_0 := a$;
$w_0 := \alpha$;
**for** $i := 0$ **to** $N - 1$ **do**
  $K_1 := hf(t_i, w_i)$;
  $K_2 := hf(t_i + h/2, w_i + K_1/2)$;
  $K_3 := hf(t_i + h/2, w_i + K_2/2)$;
  $K_4 := hf(t_i + h, w_i + K_3)$;
  $w_{i+1} := w_i + (K_1 + 2 * K_2 + 2 * K_3 + K_4)$;
  $t_{i+1} := t_i + h$;
**end for**

**Exemplul 8.5.2.** Utilizând metoda Runge-Kutta de ordinul 4 pentru a aproxima soluţia problemei Cauchy

$$\begin{aligned} &y' = -y + t + 1, \quad t \in [0, 1] \\ &y(0) = 1, \end{aligned}$$

cu $h = 0.1$, $N = 10$ şi $t_i = 0.1i$ se obţin rezultatele din tabelul 8.1. ♢

Se obişnuieşte să se asocieze unei metode Runge-Kutta cu $r$ stadii (8.5.3) tabloul

$$\begin{array}{c|cccc} \mu_1 & \lambda_{11} & \lambda_{12} & \dots & \lambda_{1r} \\ \mu_2 & \lambda_{21} & \lambda_{22} & \dots & \lambda_{2r} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \mu_r & \lambda_{r1} & \lambda_{r2} & \dots & \lambda_{rr} \\ \hline & \alpha_1 & \alpha_2 & \dots & \alpha_r \end{array} \qquad \left( \text{în formă matricială } \begin{array}{c|c} \mu & \Lambda \\ \hline & \alpha^T \end{array} \right)$$

numit *tabelă Butcher*. Pentru o metodă explicită $\mu_1 = 0$ şi $\Lambda$ este triunghiulară inferior, cu zerouri pe diagonală. Putem asocia primelor $r$ linii ale tabelei Butcher o formulă de cuadratură $\int_0^{\mu_s} u(t)\ \mathrm{d}t \approx \sum_{j=1}^{r} \lambda_{sj} u(\mu_j)$, $s = \overline{1, r}$ şi ultimei linii formula de cuadratură

| $t_i$ | Aproximante | Valori exacte | Eroarea |
|---|---|---|---|
| 0.0 | 1 | 1 | 0 |
| 0.1 | 1.00483750000 | 1.00483741804 | 8.19640e-008 |
| 0.2 | 1.01873090141 | 1.01873075308 | 1.48328e-007 |
| 0.3 | 1.04081842200 | 1.04081822068 | 2.01319e-007 |
| 0.4 | 1.07032028892 | 1.07032004604 | 2.42882e-007 |
| 0.5 | 1.10653093442 | 1.10653065971 | 2.74711e-007 |
| 0.6 | 1.14881193438 | 1.14881163609 | 2.98282e-007 |
| 0.7 | 1.19658561867 | 1.19658530379 | 3.14880e-007 |
| 0.8 | 1.24932928973 | 1.24932896412 | 3.25617e-007 |
| 0.9 | 1.30656999120 | 1.30656965974 | 3.31459e-007 |
| 1.0 | 1.36787977441 | 1.36787944117 | 3.33241e-007 |

Tabela 8.1: Rezultate numerice pentru exemplul 8.5.2

$\int_0^1 u(t)\,\mathrm{d}t \approx \sum_{s=1}^{r} \alpha_s u(\mu_j)$. Dacă gradele de exactitate respective sunt $d_s = q_s - 1$, $1 \leq s \leq r+1$ ($d_s = \infty$ dacă $\mu_s = 0$ şi toţi $\lambda_{sj} = 0$) atunci din teorema lui Peano rezultă că în reprezentarea restului apar derivatele de ordinul $q_s$ ale lui $u$ şi deci punând $u(t) = y'(x+th)$ se obţine

$$\frac{y(x+\mu_s h) - y(x)}{h} - \sum_{j=1}^{r} \lambda_{sj} y'(x+\mu_j h) = O\left(h^{q_s}\right), \qquad s = \overline{1,r}$$

şi

$$\frac{y(x+h) - y(x)}{h} - \sum_{s=1}^{r} \alpha_s y'(x+\mu_s h) = O\left(h^{q_r+1}\right).$$

Pentru metoda Runge-Kutta clasică de ordinul patru (8.5.8) tabela Butcher este:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

# 8.6. Descrierea globală a metodelor cu un pas

Descrierea metodelor se face cel mai bine în termeni de grile şi funcţii grilă.

O *grilă* pe intervalul $[a, b]$ este o mulţime de puncte $\{x_n\}_{n=0}^N$ astfel încât

$$a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b, \tag{8.6.1}$$

cu lungimile grilei definite prin

$$h_n = x_{n+1} - x_n, \quad n = 0, 1, \ldots, N-1. \tag{8.6.2}$$

Fineţea grilei este măsurată prin

$$|h| = \max_{0 \le n \le N-1} h_n. \tag{8.6.3}$$

Vom utiliza litera $h$ pentru a desemna colecţia de lungimi $h = \{h_n\}$. Dacă $h_1 = h_2 = \cdots = h_N = (b-a)/N$, grila se numeşte *uniformă*, iar în caz contrar *neuniformă*. O funcţie cu valori vectoriale $v = \{v_n\}$, $v_n \in \mathbb{R}^d$, definită pe grila (8.6.1) se numeşte *funcţie grilă*. Astfel, $v_n$ este valoarea funcţiei $v$ în punctul $x_n$ al grilei. Orice funcţie $v(x)$ definită pe $[a, b]$ induce o funcţie grilă prin restricţie. Vom desemna mulţimea funcţiilor grilă pe $[a, b]$ prin $\Gamma_h[a, b]$ şi pentru fiecare funcţie grilă $v = \{v_n\}$ definim norma sa prin

$$\|v\|_\infty = \max_{0 \le n \le N} \|v_n\|, \quad v \in \Gamma_h[a, b] \tag{8.6.4}$$

O metodă cu un pas – de fapt orice metodă discretă – este o metodă care produce o funcţie grilă $u = \{u_n\}$ astfel încât $u \approx y$, unde $y = \{y_n\}$ este funcţia grilă indusă de soluţia exactă a problemei Cauchy. Fie metoda

$$\begin{aligned} x_{n+1} &= x_n + h_n \\ u_{n+1} &= u_n + h_n \Phi(x_n, u_n; h_n) \end{aligned} \tag{8.6.5}$$

unde $x_0 = a$, $u_0 = y_0$.

Pentru a clarifica analogia dintre (8.1.1) şi (8.6.5) vom introduce operatorii $R$ şi $R_h$ care acţionează pe $C^1[a, b]$ şi respectiv pe $\Gamma_h[a, b]$. Aceştia sunt operatorii reziduali

$$(Rv)(x) := v'(x) - f(x, v(x)), \quad v \in C^1[a, b] \tag{8.6.6}$$

$$(R_h v)_n := \frac{1}{h_n}(v_{n+1} - v_n) - \Phi(x_n, v_n; h_n), \quad n = 0, 1, \ldots, N-1, \tag{8.6.7}$$

unde $v = \{v_n\} \in \Gamma_h[a, b]$. (Funcţia grilă $\{(R_h v)_n\}$ nu este definită pentru $n = N$, dar putem lua arbitrar $(R_h v)_N = (R_h v)_{N-1}$). Atunci problema Cauchy şi analogul său discret (8.6.5) se pot scrie transparent ca

$$Ry = 0 \text{ pe } [a, b], \quad y(a) = y_0 \tag{8.6.8}$$

$$R_h u = 0 \text{ pe } [a, b], \quad u_0 = y_0 \tag{8.6.9}$$

De notat că operatorul rezidual (8.6.7) este strâns înrudit cu eroarea locală de trunchiere (8.3.3) când aplicăm operatorul într-un punct $(x_n, y(x_n))$ pe traiectoria soluţiei exacte. Atunci, într-adevăr, soluţia de referinţă $u(t)$ coincide cu soluţia $y(t)$ şi

$$(R_h y)_n = \frac{1}{h_n}[y(x_{n+1}) - y(x_n)] - \Phi(x_n, y(x_n); h_n) = \\ = -T(x_n, y(x_n); h_n). \tag{8.6.10}$$

## 8.6.1. Stabilitatea

Stabilitatea este o proprietate numai a schemei numerice (8.6.5) şi nu are nimic de-a face apriori cu puterea de aproximare. Ea caracterizează robusteţea schemei în raport cu perturbaţii mici. Totuşi stabilitatea combinată cu consistenţa conduce la convergenţa soluţiei numerice către soluţia adevărată. Definim stabilitatea în termeni de operatori reziduali discreţi $R_h$ în (8.6.7). Ca de obicei, presupunem că $\Phi(x, y; h)$ este definită pe $[a, b] \times \mathbb{R}^d \times [0, h_0]$, unde $h_0 > 0$ este un număr pozitiv adecvat.

**Definiţia 8.6.1** *Metoda (8.6.5) se numeşte* stabilă *pe $[a, b]$ dacă există o constantă $K > 0$ care nu depinde de $h$ astfel încât pentru o grilă arbitrară $h$ pe $[a, b]$ şi pentru două funcţii grilă arbitrare $v, w \in \Gamma_h[a, b]$ are loc*

$$\|v - w\|_\infty \le K \left( \|v_0 - w_0\|_\infty + \|R_h v - R_h w\|_\infty \right), \quad v, w \in \Gamma_h[a, b] \tag{8.6.11}$$

*pentru orice $h$ cu $|h|$ suficient de mică. În (8.6.11) norma este definită prin (8.6.4).*

Vom numi în continuare (8.6.11) *inegalitatea de stabilitate*. Motivaţia ei este următoarea. Să presupunem că avem două funcţii grilă $u, w$ ce satisfac

$$R_h u = 0, \qquad u_0 = y_0 \tag{8.6.12}$$

$$R_h w = \varepsilon, \qquad w_0 = y_0 + \eta_0, \tag{8.6.13}$$

unde $\varepsilon = \{\varepsilon_n\} \in \Gamma_h[a, b]$ este o funcţie grilă cu $\|\varepsilon_n\|$ mic şi $\|\eta_0\|$ este de asemenea mic. Putem interpreta $u \in \Gamma_h[a, b]$ ca fiind rezultatul aplicării schemei numerice (8.6.5) cu precizie infinită, în timp ce $w \in \Gamma_h[a, b]$ ar putea fi soluţia lui (8.6.5) în aritmetica în virgulă flotantă. Atunci, dacă stabilitatea are loc, avem

$$\|u - w\|_\infty \le K(\|\eta_0\|_\infty + \|\varepsilon\|_\infty), \tag{8.6.14}$$

adică, schimbarea locală în $u$ este de acelaşi ordin de mărime ca şi eroarea reziduală locală $\{\varepsilon_n\}$ şi eroarea iniţială $\eta_0$. Trebuie apreciat, totuşi, că prima ecuaţie în (8.6.13) spune

$$w_{n+1} - w_n - h_n \Phi(x_n, w_n, h_n) = h_n \varepsilon_n,$$

însemnând că erorile de rotunjire trebuie să tindă către zero atunci când $|h| \to \infty$.

Interesant, pentru stabilitate este necesară doar o condiţie Lipschitz asupra lui $\Phi$.

**Teorema 8.6.2** *Dacă $\Phi(x, y; h)$ satisface o condiţie Lipschitz în raport cu variabila $y$,*

$$\|\Phi(x, y; h) - \Phi(x, y^*; h)\| \leq M\|y - y^*\| \text{ pe } [a, b] \times \mathbb{R}^d \times [0, h_0], \tag{8.6.15}$$

*atunci metoda (8.6.5) este stabilă.*

Vom pregăti demonstraţia prin următoarea lemă utilă.

**Lema 8.6.3** *Fie $\{e_n\}$ o secvenţă de numere, $e_n \in \mathbb{R}$, ce satisface*

$$e_{n+1} \leq a_n e_n + b_n, \quad n = 0, 1, \ldots, N-1 \tag{8.6.16}$$

*unde $a_n > 0$ şi $b_n \in \mathbb{R}$. Atunci*

$$e_n \leq E_n, \quad E_n = \left(\prod_{k=0}^{n-1} a_k\right) e_0 + \sum_{k=0}^{n-1} \left(\prod_{\ell=k+1}^{n-1} a_\ell\right) b_k, \quad n = 0, 1, \ldots, N \tag{8.6.17}$$

Adoptăm aici convenţia uzuală că un produs vid are valoarea 1 şi o sumă vidă are valoarea 0.

*Demonstraţia lemei 8.6.3.* Se verifică că

$$E_{n+1} = a_n E_n + b_n, \quad n = 0, 1, \ldots, N-1, \quad E_0 = e_0.$$

Scăzând aceasta din (8.6.16) se obţine

$$e_{n+1} - E_{n+1} \leq a_n(e_n - E_n), \quad n = 0, 1, \ldots, N-1.$$

Acum, $e_0 - E_0 = 0$, aşa că $e_1 - E_1 \leq 0$, căci $a_0 > 0$. Prin inducţie, mai general, $e_n - E_n \leq 0$, deoarece $a_{n-1} > 0$. □

*Demonstraţia teoremei 8.6.2.* Fie $h = \{h_n\}$ o grilă arbitrară pe $[a, b]$ şi $v, w \in \Gamma_h[a, b]$ două funcţii grilă cu valori vectoriale arbitrare. Din definiţia lui $R_h$ putem scrie

$$v_{n+1} = v_n + h_n \Phi(x_n, v_n; h_n) + h_n (R_h v)_n, \quad n = 0, 1, \ldots, N-1$$

şi similar pentru $w_{n+1}$. Scăzându-le obţinem

$$\begin{aligned} v_{n+1} - w_{n+1} = v_n - w_n + h_n[\Phi(x_n, v_n; h_n) - \Phi(x_n, w_n; h_n)] + \\ + h_n[(R_h v)_n - (R_h w)_n], \quad n = 0, 1, \ldots, N-1. \end{aligned} \tag{8.6.18}$$

Definim acum

$$e_n = \|v_n - w_n\|, \quad d_n = \|(R_h v)_n - (R_h w)_n\|, \quad \delta = \|d_n\|_\infty. \tag{8.6.19}$$

Utilizând inegalitatea triunghiului în (8.6.18) şi condiţia Lipschitz (8.6.19) pentru $\Phi$ obţinem

$$e_{n+1} \le (1 + h_n M)e_n + h_n\delta, \quad n = 0, 1, \ldots, N-1 \tag{8.6.20}$$

Aceasta este inegalitatea (8.6.16) cu $a_n = 1 + h_n M$, $b_n = h_n\delta$. Deoarece $k = 0, 1, \ldots, n-1$, $n \le N$ avem

$$\prod_{\ell=k+1}^{n-1} a_\ell \le \prod_{\ell=0}^{n-1} a_\ell = \prod_{\ell=0}^{N-1}(1 + h_\ell M) \le \prod_{\ell=0}^{N-1} e^{h_\ell M}$$
$$= e^{(h_0+h_1+\cdots+h_{N-1})M} = e^{(b-a)M},$$

unde în a doua inegalitate a fost utilizată inegalitatea clasică $1 + x \le e^x$, din lema 8.6.3 se obţine că

$$e_n \le e^{(b-a)M} e_0 + e^{(b-a)M} \sum_{k=0}^{n-1} h_k\delta \le$$
$$\le e^{(b-a)M}(e_0 + (b-a)\delta), \quad n = 0, 1, \ldots, N-1.$$

De aceea

$$\|e\|_\infty = \|v - w\|_\infty \le e^{(b-a)M}(\|v_0 - w_0\| + (b-a)\|R_h v - R_h w\|_\infty),$$

care este (8.6.11) cu $K = e^{(b-a)M}\max\{1, b-a\}$. $\square$

Am demonstrat de fapt stabilitatea pentru orice $|h| \le h_0$, nu numai pentru $h$ suficient de mic.

Toate metodele cu un pas utilizate în practică satisfac o condiţie Lipschitz dacă $f$ satisface o astfel de condiţie şi constanta $M$ pentru $\Phi$ poate fi aproximată cu ajutorul constantei $L$ pentru $f$. Este evident pentru metoda lui Euler şi nu este dificil de demonstrat pentru celelalte. Este util de observat că $\Phi$ nu este nevoie să fie continuă în $x$; continuitatea pe porţiuni fiind suficientă atât timp cât (8.6.15) are loc pentru orice $x \in [a, b]$, luând limitele laterale în punctele de discontinuitate.

Vom folosi următoarea aplicaţie a lemei 8.6.3, relativă la o funcţie grilă $v \in \Gamma_h[a, b]$ ce satisface

$$v_{n+1} = v_n + h_n(A_n v_n + b_n), \qquad n = 0, 1, ..., N-1, \tag{8.6.21}$$

unde $A_n \in \mathbb{R}^{d\times d}$, $b_n \in \mathbb{R}^d$, şi $h_n$ este o grila arbitrară pe $[a, b]$.

**Lema 8.6.4** *Presupunem că în* (8.6.21)

$$\|A_n\| \le M, \qquad \|b_n\| \le \delta, \qquad n = 0, 1, \ldots, N-1, \tag{8.6.22}$$

*unde constantele $M$, $\delta$ nu depind de $h$. Atunci, există o constantă $K > 0$, independentă de $h$, dar depinzând de $\|v_0\|$, astfel încât*

$$\|v\|_\infty \le K. \tag{8.6.23}$$

*Demonstraţie.* Lemma rezultă observând că

$$\|v_{n+1}\|\| \leq (1 + h_n M)\|v_n\| + h_n\delta, \quad n = 0, 1, \ldots, N-1,$$

care este chiar inegalitatea (8.6.19) din demonstraţia teoremei 8.6.2, deci

$$\|v_n\| \leq e^{(b-a)M}\{\|v_0\| + (b-a)\delta\}. \tag{8.6.24}$$

□

## 8.6.2. Convergenţă

Stabilitatea este un concept puternic. Ea implică aproape imediat convergenţa şi este un instrument de deducere a estimării erorii globale. Vom începe prin a defini precis ce înseamnă convergenţa.

**Definiţia 8.6.5** *Fie $a = x_0 < x_1 < x_2 < \cdots < x_N = b$ o grilă pe $[a, b]$ cu lungimea grilei $|h| = \max\limits_{1\leq n\leq N}(x_n - x_{n-1})$. Fie $u = \{u_n\}$ o funcţie grilă definită aplicând metoda (8.6.5) pe $[a, b]$ şi $y = \{y_n\}$ funcţia grilă indusă de soluţia exactă a problemei Cauchy. Metoda (8.6.5) se numeşte* convergentă *pe $[a, b]$ dacă are loc*

$$\|u - y\|_\infty \to 0, \text{ când } |h| \to 0 \tag{8.6.25}$$

**Teorema 8.6.6** *Dacă metoda (8.6.5) este consistentă şi stabilă pe $[a, b]$, atunci ea converge. Mai mult, dacă $\Phi$ are ordinul $p$, atunci*

$$\|u - y\|_\infty = O(|h|^p) \text{ când } |h| \to 0. \tag{8.6.26}$$

*Demonstraţie.* Din inegalitatea de stabilitate (8.6.11) aplicată funcţiilor grilă $v = h$ şi $w = y$ din definiţia 8.6.5 avem pentru $|h|$ suficient de mic

$$\|u - y\|_\infty \leq K(\|u_0 - y(x_0)\| + \|R_h u - R_h y\|_\infty) = K\|R_h y\| \tag{8.6.27}$$

deoarece $u_0 = y(x_0)$ şi $R_h u = 0$ conform (8.6.5). Dar din (8.6.10)

$$\|R_h y\|_\infty = \|T(\cdot, y; h)\|_\infty \tag{8.6.28}$$

unde $T$ este eroare de trunchiere a metodei $\Phi$. Din definiţia consistenţei

$$\|T(\cdot, y; h)\|_\infty \to 0, \text{ când } |h| \to 0,$$

ceea ce demonstrează prima parte a teoremei. Partea a doua rezultă imediat din (8.6.27) şi (8.6.28), deoarece ordinul $p$, prin definiţia înseamnă că

$$\|T(\cdot, y; h)\|_\infty = O(|h|^p), \text{ când } |h| \to 0. \tag{8.6.29}$$

□

### 8.6.3. Asimptotica erorii globale

Deoarece funcţia eroare principală descrie contribuţia termenului principal al erorii locale de trunchiere este de interes să identificăm termenul dominant în eroarea globală $u_n - y(x_n)$. Pentru a simplifica lucrurile vom presupune că avem o grilă de lungime constantă $h$, deşi nu este dificil să lucrăm cu o grilă de lungime variabilă de forma $h_n = \vartheta(x_n)h$, unde $\vartheta(x)$ este o funcţie continuă pe porţiuni şi $0 < \vartheta(x) < \theta$ pentru $a \leq x \leq b$. Astfel, considerăm că metoda cu un pas având forma

$$\begin{aligned} &x_{n+1} = x_n + h \\ &u_{n+1} = u_n + h\Phi(x_n, u_n; h); \qquad n = 0, 1, \ldots, N-1 \\ &x_0 = a, \qquad u_0 = y_0, \end{aligned} \tag{8.6.30}$$

defineşte o funcţie grilă $u = \{u_n\}$ pe o grilă uniformă pe $[a, b]$. Suntem interesaţi în comportarea asimptotică a lui $u_n - y(x_n)$ când $h \to 0$, unde $y(x)$ este soluţia exactă a problemei Cauchy

$$\begin{cases} \dfrac{dy}{dx} = f(x, y) \quad x \in [a, b] \\ y(a) = y_0 \end{cases} \tag{8.6.31}$$

**Teorema 8.6.7** *Presupunem că*

*(1)* $\Phi(x, y, h) \in C^2\left([a, b] \times \mathbb{R}^d \times [0, h_0]\right)$;

*(2)* $\Phi$ *este o metodă de ordin* $p \geq 1$ *ce admite o funcţie de eroare principală* $\tau(x, y) \in C\left([a, b] \times \mathbb{R}^d\right)$*;*

*(3)* $e(x)$ *este soluţia problemei Cauchy*

$$\begin{cases} \frac{de}{dx} = f_y(x, y(x))e + \tau(x, y(x)), \quad a \leq x \leq b \\ e(a) = 0 \end{cases} \tag{8.6.32}$$

*Atunci, pentru* $n = \overline{0, N}$,

$$u_n - y(x_n) = e(x_n)h^p + O(h^{p+1}), \quad \text{când } h \to 0. \tag{8.6.33}$$

Înainte de a demonstra teorema, facem următoarele observaţii:

1. Semnificaţia precisă a lui (8.6.33) este

$$\|u - y - h^p e\|_\infty = O(h^{p+1}),$$

unde $u$, $y$, $e$ sunt funcţiile grilă $u = \{u_n\}$, $y = \{y(x_n)\}$ şi $e = \{e(x_n)\}$.

2. Datorită consistenţei $\Phi(x,y;0) = f(x,y)$, condiţia (1) din enunţ implică $f \in C^2([a,b]\times\mathbb{R}^d)$, ceea ce este mai mult decât suficient pentru a garanta existenţa şi unicitatea soluţiei $e(x)$ a lui (8.6.32) pe întreg intervalul $[a,b]$.

3. Faptul că anumite componente, dar nu toate ale lui $\tau(x,y)$ ar putea să fie identic nule nu implică faptul că $e(x)$ se anulează de asemenea, deoarece (8.6.32) este un sistem cuplat de ecuaţii diferenţiale.

*Demonstraţia teoremei 8.6.7.* Vom începe cu un calcul ajutător, estimarea lui

$$\Phi(x_n,u_n;h) - \Phi(x_n,y(x_n);h). \tag{8.6.34}$$

Conform teoremei lui Taylor (pentru funcţii de mai multe variabile), aplicată celei de-a $i$-a componente a lui (8.6.34), avem

$$\begin{aligned}\Phi^i(x_n,u_n;h)-\Phi^i(x_n,y(x_n);h) = \sum_{j=1}^{d} \Phi^i y^j\,(x_n,y(x_n);h)\left[u_n^j - y^j(x_n)\right]\\ + \frac{1}{2}\sum_{j,k=1}^{d} \Phi^i y^j y^k\,(x_n,\overline{u}_n;h)\left[u_n^j - y^j(x_n)\right]\left[u_n^k - y^k(x_n)\right],\end{aligned} \tag{8.6.35}$$

unde $\overline{u}_n$ este pe segmentul ce uneşte $u_n$ şi $y(x_n)$. Utilizând teorema lui Taylor încă o dată, în variabila $h$, putem scrie

$$\Phi^i_{y^j}(x_n,y(x_n);h) = \Phi^i_{y^j}(x_n,y(x_n);0) + h\Phi^i_{y^jh}\left(x_n,y(x_n);\bar{h}\right),$$

unde $0<\bar{h}<h$. Deoarece, conform consistenţei, $\Phi(x,y;0)\equiv f(x,y)$ pe $[a,b]\times\mathbb{R}^d$, avem

$$\Phi^i_{y^j}(x,y;0) = f^i_{y^j}(x,y), \qquad x\in[a,b], \quad y\in\mathbb{R}^d,$$

şi condiţia (1) ne permite să scriem

$$\Phi^i_{y^j}(x_n,y(x_n);h) = f^i_{y^j}(x_n,y(x_n)) + O(h), \quad h\to 0. \tag{8.6.36}$$

Observând acum că $u_n - y(x_n) = O(h^p)$, în virtutea teoremei 8.6.6 şi utilizând (8.6.36) şi (8.6.35), obţinem conform ipotezei (1),

$$\begin{aligned}\Phi^i(x_n,u_n;h) - \Phi^i(x_n,y(x_n);h) = \sum_{j=1}^{d} f^i_{y^j}\,(x_n,y(x_n))\left[u_n^j - y^j(x_n)\right] +\\ O(h^{p+1}) + O(h^{2p}).\end{aligned}$$

Dar $O(h^{2p})$ este de ordinul $O(h^{p+1})$, căci $p\geq 1$. Astfel, în notaţie vectorială,

$$\Phi(x_n,u_n;h) - \Phi(x_n,y(x_n);h) = f_y\,(x_n,y(x_n))\left[u_n - y(x_n)\right] + O(h^{p+1}). \tag{8.6.37}$$

Acum, pentru a evidenţia termenul dominant în eroarea globală, definim funcţia grilă $r = \{r_n\}$ prin

$$r = h^{-p}(u - y). \tag{8.6.38}$$

Atunci

$$\begin{aligned}\frac{1}{h}(r_{n+1} - r_n) &= \frac{1}{h}\left[h^{-p}\left(u_{n+1} - y(x_{n+1})\right) - h^{-p}\left(u_n - y(x_n)\right)\right] = \\ &= h^p\left[\frac{1}{h}(u_{n+1} - u_n) - \frac{1}{h}\left(y(x_{n+1}) - y(x_n)\right)\right] = \\ &= h^{-p}\left\{\Phi(x_n, u_n; h) - [\Phi(x_n, y(x_n); h) - T(x_n, y(x_n); h)]\right\},\end{aligned}$$

unde am utilizat (8.6.30) şi relaţia (8.6.10) pentru eroarea de trunchiere $T$. De aceea, exprimând $T$ cu ajutorul funcţiei eroare principală $\tau$, obţinem

$$\begin{aligned}\frac{1}{h}(r_{n+1} - r_n) = h^{-p}\big[&\Phi(x_n, u_n; h) - \Phi(x_n, y(x_n); h) + \tau(x_n, y(x_n))h^p \\ &+ O(h^{p+1})\big]\end{aligned}$$

Pentru primii doi termeni din paranteză utilizăm (8.6.37) şi definiţia lui $r$ (8.6.38) pentru a obţine

$$\begin{gathered}\tfrac{1}{h}(r_{n+1} - r_n) = f_y\left(x_n, y(x_n)\right) r_n + \tau(x_n, y(x_n)) + O(h), \quad n = \overline{0, N-1} \\ r_0 = 0.\end{gathered} \tag{8.6.39}$$

Acum punând

$$g(x, y) = f_y(x, y(x))y + \tau(x, y(x)) \tag{8.6.40}$$

interpretăm (8.6.39) scriind

$$\left(R_h^{Euler,g} r\right)_n = \varepsilon_n, \quad n = \overline{0, N-1}, \quad \varepsilon_n = O(h),$$

unde $R_h^{Euler,g}$ este operatorul rezidual discret (8.6.7) corespunzător metodei lui Euler aplicată lui $e' = g(x, e)$, $e(a) = 0$. Deoarece metoda lui Euler este stabilă pe $[a, b]$ şi $g$ fiind liniară în $y$ satisface o condiţie Lipschitz uniformă, avem conform inegalităţii de stabilitate (8.6.11)

$$\|r - e\|_\infty = O(h),$$

şi conform lui (8.6.38)

$$\|u - y - h^p e\|_\infty = O(h^{p+1}),$$

aşa cum trebuia arătat. □

# 8.7. Monitorizarea erorilor şi controlul pasului

Vom încerca să realizăm monitorizarea erorilor globale, cel puţin asimptotic, implementând rezultatul din teorema 8.6.7. Aceasta necesită evaluarea matricei jacobiene $f_y(x, y)$ de-a lungul sau în apropierea traiectoriei soluţiei; acest lucru este natural, deoarece $f_y(x, y)$ guvernează, într-o primă aproximare, efectul perturbaţiilor prin ecuaţia diferenţială variaţională (8.6.32). În această ecuaţie tonul este dat de funcţia de eroare principală, evaluată de-a lungul traiectoriei, aşa că estimarea erorii locale de trunchiere (mai exact a funcţiei de eroare principală) este de asemenea necesară în această abordare. Pentru simplitate vom presupune că grila are o lungime constantă.

## 8.7.1. Estimarea erorii globale

Ideea estimării este de a integra „ecuaţia variaţională“ (8.6.32) împreună cu ecuaţia principală (8.6.31). Deoarece avem nevoie de $e(x)$ în (8.6.31) numai în limita unei erori de $O(h)$ (orice termen de eroare $O(h)$ din $e(x_n)$, înmulţit cu $h^p$, fiind absorbit de termenul $O(h^{p-1})$), putem utiliza în acest scop metoda lui Euler, care va furniza aproximaţia dorită $v_n \approx e(x_n)$.

**Teorema 8.7.1** *Presupunem că*

*(1)* $\Phi(x, y; h) \in C^2\left([a, b] \times \mathbb{R}^d \times [0, h_0]\right)$;

*(2)* $\Phi$ *este o metodă de ordin* $p \geq 1$ *ce admite o funcţie de eroare principală* $\tau(x, y) \in C^1\left([a, b] \times \mathbb{R}^d\right)$;

*(3) este disponibilă o estimaţie* $r(x, y; h)$ *pentru funcţie de eroare principală ce satisface*

$$r(x, y; h) = \tau(x, y) + O(h), \quad h \to 0, \tag{8.7.1}$$

*uniform pe* $[a, b] \times \mathbb{R}^d$;

*(4) odată cu funcţia grilă* $u = \{u_n\}$ *generăm funcţia grilă* $v = \{v_n\}$ *în modul următor*

$$\begin{aligned} &x_{n+1} = x_n + h; \\ &u_{n+1} = u_n + h\Phi(x_n, u_n; h) \\ &v_{n+1} = v_n + h\left[f_y(x_n, v_n)v_n + r(x_n, u_n; h)\right] \\ &x_0 = a, \quad u_0 = y_0, \quad v_0 = 0. \end{aligned} \tag{8.7.2}$$

*Atunci pentru orice* $n = \overline{0, N-1}$,

$$u_n - y(x_n) = v_n h^p + O(h^{p+1}), \quad \text{când } h \to 0. \tag{8.7.3}$$

*Demonstraţie.* Demonstaţia începe cu stabilirea următoarelor estimări

$$f_y(x_n, u_n) = f_y(x_n, y(x_n)) + O(h), \tag{8.7.4}$$
$$r(x_n, u_n; h) = \tau(x_n, y(x_n)) + O(h). \tag{8.7.5}$$

Din ipoteza (1) observăm pe baza consistenţei $f(x,y) = \Phi(x,y;0)$ că $f(x,y) \in C^2\left([a,b] \times \mathbb{R}^d\right)$. Ţinând cont de teorema 8.6.6, avem $u_n = y(x_n) + O(h^p)$, şi deci

$$f_y(x_n, u_n) = f_y(x_n, y_n) + O(h^p),$$

relaţie ce implică (8.7.4), deoarece $p \geq 1$. În continuare, deoarece $\tau(x,y) \in C^1\left([a,b] \times \mathbb{R}^d\right)$, conform ipotezei (2) avem

$$\begin{aligned}\tau(x_n, u_n) &= \tau(x_n, y(x_n)) + \tau_y(x_n, \bar{u}_n)(u_n - y(x_n)) \\ &= \tau(x_n, y(x_n)) + O(h^p)\end{aligned}$$

şi aplicând apoi ipoteza (3) obţinem

$$r(x_n, u_n; h) = \tau(x_n, u_n) + O(h) = \tau(x_n, y(x_n)) + O(h^p) + O(h),$$

din care rezultă imediat (8.7.5).

Fie (a se compara cu (8.6.40))

$$g(x,y) = f_y(x, y(x))y + \tau(x, y(x)). \tag{8.7.6}$$

Ecuaţia pentru $v_{n+1}$ în (8.7.2) are forma

$$v_{n+1} = v_n + h(A_n v_n + b_n),$$

unde $A_n$ sunt matrice mărginite şi $b_n$ sunt vectori mărginiţi. Conform lemei 8.6.6, avem mărginirea lui $v_n$,

$$v_n = O(1), \qquad h \to 0. \tag{8.7.7}$$

Înlocuind (8.7.4) şi (8.7.5) în ecuaţia lui $v_{n+1}$ şi ţinând cont de (8.7.7) obţinem

$$\begin{aligned}v_{n+1} &= v_n + h\left[f_y(x_n, y(x_n))v_n + \tau(x_n, y(x_n)) + O(h)\right] \\ &= v_n + hg(x_n, v_n) + O(h^2).\end{aligned}$$

Astfel, cu notaţia utilizată în demonstraţia teoremei 8.6.7

$$\left(R_h^{Euler,g} v\right)_n = O(h), \quad v_0 = 0.$$

Deoarece metoda lui Euler este stabilă,

$$v_n - e(x_n) = O(h),$$

unde $e(x)$ este, ca mai înainte, soluţia ecuaţiei

$$e' = g(x, e)$$
$$e(a) = 0$$

Deci, conform lui (8.6.33)

$$u_n - y(x_n) = e(x_n)h^p + O(h^{p+1}).$$

□

## 8.7.2. Estimarea erorii de trunchiere

Pentru a aplica teorema 8.7.1 avem nevoie de estimări $r(x, y; h)$ ale funcţiei de eroare principală $\tau(x, y)$ care să aibă precizia $O(h)$. Vom descrie două dintre ele, în ordinea crescătoare a eficienţei.

### Extrapolare Richardson la zero

Aceasta funcţionează pentru orice metodă cu un pas $\Phi$, dar de obicei este considerată prea costisitoare. Dacă $\Phi$ are ordinul $p$, procedura este următoarea

$$\begin{aligned} y_h &= y + h\Phi(x, y; h), \\ y_{h/2} &= y + \frac{1}{2}h\Phi\left(x, y; \frac{1}{2}h\right), \\ y_h^* &= y_{h/2} + \frac{1}{2}h\Phi\left(x + \frac{1}{2}h, y_{h/2}; \frac{1}{2}h\right), \\ r(x, y; h) &= \frac{1}{1 - 2^{-p}} \frac{1}{h^{p+1}} \left(y_h - y_h^*\right). \end{aligned} \tag{8.7.8}$$

De notat că $y_h^*$ este rezultatul aplicării lui $\Phi$ pentru doi paşi consecutivi, fiecare de lungime $h/2$, pe câtă vreme $y_h$ este rezultatul aplicării lui $\Phi$ pentru un pas de lungime $h$.

Să verificăm acum că $r(x, y; h)$ dat grupul de formule (8.7.8) este o estimaţie acceptabilă. Pentru aceasta trebuie să presupunem că $\tau(x, y) \in C^1\left([a, b] \times \mathbb{R}^d\right)$. Conform lui (8.3.4) şi (8.3.8), utilizând soluţia de referinţa $u(t)$ ce trece prin $(x, y)$ avem

$$\Phi(x, y; h) = \frac{1}{h}[u(x + h) - u(x)] + \tau(x, y)h^p + O(h^{p+1}). \tag{8.7.9}$$

Mai mult,

$$\begin{aligned} \frac{1}{h}(y_h - y_h^*) &= \frac{1}{h}(y_h - y_{h/2}) + \Phi(x, y; h) - \frac{1}{2}h\Phi\left(x + \frac{1}{2}h, y_{h/2}; \frac{1}{2}h\right) \\ &= \Phi(x, y; h) - \frac{1}{2}\Phi\left(x, y; \frac{1}{2}h\right) - \frac{1}{2}h\Phi\left(x + \frac{1}{2}h, y_{h/2}; \frac{1}{2}h\right). \end{aligned}$$

Aplicând (8.7.9) fiecărui termen din membrul drept găsim

$$\begin{aligned}\frac{1}{h}(y_h - y_h^*) = \frac{1}{h}[u(x+h) - u(x)] + \tau(x,y)h^p + O(h^{p+1}) \\ - \frac{1}{2}\frac{1}{h/2}\left[u\left(x+\frac{1}{2}h\right) - u(x)\right] - \frac{1}{2}\tau(x,y)\left(\frac{1}{2}h^p\right) + O(h^{p+1}) \\ - \frac{1}{2}\frac{1}{h/2}\left[u\left(x+h\right) - u\left(x+\frac{1}{2}h\right)\right] - \frac{1}{2}\tau\left(x+\frac{1}{2}h, y+O(h)\right)\left(\frac{1}{2}h^p\right) \\ + O(h^{p+1}) = \tau(x,y)(1-2^{-p})h^p + O(h^{p+1}).\end{aligned}$$

În consecinţă

$$\frac{1}{1-2^{-p}}\frac{1}{h}(y_h - y_h^*) = \tau(x,y)h^p + O(h^{p+1}), \tag{8.7.10}$$

aşa cum s-a dorit.

Scăzând (8.7.10) din (8.7.9), rezultă incidental că

$$\Phi^*(x,y;h) := \Phi(x,y;h) - \frac{1}{1-2^{-p}}\frac{1}{h}(y_h - y_h^*) \tag{8.7.11}$$

defineşte o metodă cu un pas de ordin $p+1$.

Procedura (8.7.8) este costisitoare. Pentru un proces Runge-Kutta de ordinul 4 sunt necesare în total 11 evaluări ale lui $f$ pe pas, aproape de trei ori mai mult decât pentru un pas Runge-Kutta. De aceea, extrapolarea Richardson este utilizată numai după fiecare doi paşi ai lui $\Phi$, adică se continuă în conformitate cu formulele

$$\begin{aligned}y_h &= y + h\Phi(x,y;h), \\ y_{2h}^* &= y_h + h\Phi(x+h, y_h; h) \\ y_{2h} &= y + 2h\Phi(x,y;2h).\end{aligned} \tag{8.7.12}$$

Atunci (8.7.10) ne dă

$$\frac{1}{2(2^p-1)}\frac{1}{h^{p+1}}(y_{2h} - y_{2h}^*) = \tau(x,y) + O(h), \tag{8.7.13}$$

aşa că expresia din membrul drept este un estimator acceptabil al lui $r(x,y;h)$. Dacă cei doi paşi din (8.7.12) conduc la o precizie acceptabilă (a se vedea subsecţiunea 8.7.3), atunci pentru un proces Runge-Kutta de ordinul 4 procedura necesită numai trei evaluări adiţionale ale lui $f$, deoarece $y_h$ şi $y_{2h}^*$ trebuie calculat oricum. Vom vedea că există scheme mai eficiente.

**Metode scufundate (imbricate)**

Ideea de bază a aceste abordări este următoarea: se consideră o metodă $\Phi$ de ordinul $p$ şi o metodă $\Phi^*$ de ordinul $p^* = p + 1$ şi se defineşte

$$r(x, y; h) = \frac{1}{h^p}[\Phi(x, y; h) - \Phi^*(x, y; h)]. \tag{8.7.14}$$

Acesta este un estimator acceptabil, aşa cum rezultă scăzând relaţiile

$$\Phi(x, y; h) - \frac{1}{h}[u(x + h) - u(x)] = \tau(x, y)h^p + O(h^{p+1})$$
$$\Phi^*(x, y; h) - \frac{1}{h}[u(x + h) - u(x)] = O(h^{p+1})$$

şi împărţind rezultatul cu $h^p$.

Cheia problemei este de a face această procedură eficientă. Urmând o idee a lui Fehlberg, putem încerca să facem aceasta incluzând un proces Runge-Kutta de ordinul $p$ în altul de ordin $p + 1$. Mai concret, fie $\Phi$ o metodă Runge-Kutta explicită în $r$ stadii

$$K_1(x, y) = f(x, y)$$
$$K_s(x, y; h) = f\left(x + \mu_s h; y + h\sum_{j=1}^{s-1} \lambda_{sj} K_j\right), \quad s = 2, 3, \ldots, r$$
$$\Phi(x, y; h) = \sum_{s=1}^{r} \alpha_s K_s$$

Atunci pentru $\Phi^*$ alegem un proces similar în $r^*$-stadii, cu $r^* > r$, astfel încât

$$\mu_s^* = \mu_s, \quad \lambda_{sj}^* = \lambda_{sj}, \text{ pentru } s = 2, 3, \ldots, r.$$

Estimarea (8.7.14) costă atunci din $r^* - r$ evaluări suplimentare ale lui $f$. Dacă $r^* = r + 1$ putem încă să mai facem economii de evaluări suplimentare, selectând (dacă este posibil)

$$\mu_r^* = 1, \quad \lambda_{rj}^* = \alpha_j \text{ pentru } j = \overline{1, r^* - 1} \quad (r^* = r + 1) \tag{8.7.15}$$

Atunci, într-adevăr, $K_r^*$ va fi identic cu $K_1$ pentru pasul următor.

Perechi de astfel de formule Runge-Kutta imbricate $(p, p + 1)$ au fost dezvoltate la sfârşitul anilor ’60 de E. Fehlberg[16, 17]. Este un grad considerabil de libertate în alegerea acestor parametri. Alegerile lui Fehlberg au fost ghidate de încercarea de a reduce mărimea coeficienţilor tututor derivatelor parţiale care intervin în funcţia de eroare principală $\tau(x, y)$ a lui $\Phi$. El a reuşit să obţină pentru parametrii $p, r, r^*$ valorile date în tabela 8.2

Pentru procesul de ordinul 3 (şi numai pentru acesta) se pot alege parametrii astfel ca să aibă loc şi (8.7.15).

| $p$ | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|
| $r$ | 4 | 5 | 6 | 8 | 11 | 15 |
| $r^*$ | 5 | 6 | 8 | 10 | 13 | 17 |

Tabela 8.2: Formule Runge-Kutta imbricate

### 8.7.3. Controlul pasului

Orice estimare $r(x, y; h)$ a funcţiei de eroare principală $\tau(x, y)$ implică o estimare

$$h^p r(x, y; h) = T(x, y; h) + O(h^{p+1}) \tag{8.7.16}$$

a erorii locale de trunchiere, care poate fi utilizată pentru a monitoriza eroarea de trunchiere în timpul procesului de integrare. Totuşi, trebuie avut în vedere faptul că eroarea locală de trunchiere este chiar diferită de eroarea globală, eroare pe care vrem de fapt să o controlăm. Pentru a obţine o mai bună cunoaştere a relaţiei dintre aceste două erori reamintim teorema următoare, care cuantifică continuitatea soluţiei problemei Cauchy în raport cu valorile iniţiale.

**Teorema 8.7.2** *Fie $f(x, y)$ continuă în $x \in [a, b]$ şi care satisface o condiţie Lipschitz cu constanta $L$, uniform pe $[a, b] \times \mathbb{R}$, adică*

$$\|f(x, y) - f(x, y^*)\| \leq L \|y - y^*\| .$$

*Atunci problema Cauchy*

$$\begin{aligned} &\frac{dy}{dx} = f(x, y), \quad x \in [a, b], \\ &y(c) = y_c \end{aligned} \tag{8.7.17}$$

*are o soluţie unică pentru orice $c \in [a, b]$ şi orice $y_c \in \mathbb{R}^d$. Fie $y(x, s)$ şi $y(x; s^*)$ soluţiile lui (8.7.17) ce corespund lui $y_c = s$ şi respectiv $y_c = s^*$. Atunci, pentru orice normă vectorială $\|.\|$,*

$$\|y(x; s) - y(x; s^*)\| \leq e^{L|x-c|} \|s - s^*\| . \tag{8.7.18}$$

Rezolvarea numerică a problemei (8.6.31) printr-o metodă cu un pas (nu neapărat constant) înseamnă în realitate că se urmăreşte o secvenţă de „piste ale soluţiei“ (expresia este din [20]) prin care în fiecare punct al grilei se sare de la o pistă la următoarea cu o cantitate egală cu eroarea de trunchiere în $x_n$ (vezi figura 8.3). Aceasta rezultă din definiţia erorii de trunchiere, soluţia de referinţă fiind una din pistele soluţiei. Mai concret, a $n$-a pistă, $n = \overline{0, N}$, este dată de soluţia problemei Cauchy

$$\begin{aligned} &\frac{dv_n}{dx} = f(x, v_n), \quad x \in [x_n, b], \\ &v_n(x_n) = u_n, \end{aligned} \tag{8.7.19}$$

şi

$$u_{n+1} = v(x_{n+1}) + h_n T(x_n, u_n; h_n), \qquad n = \overline{0, N-1.} \tag{8.7.20}$$

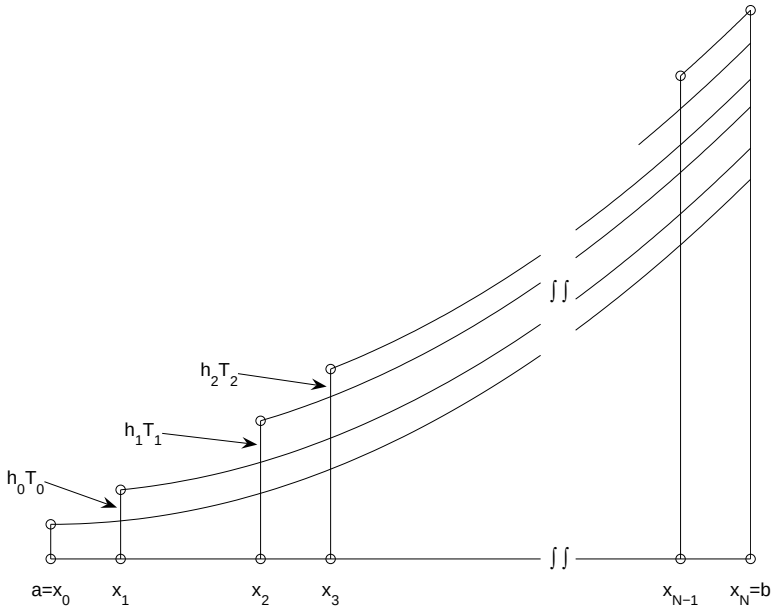



Figura 8.3: Acumularea erorilor într-o metodă cu un pas

Deoarece conform lui (8.7.19) avem $u_{n+1} = v_{n+1}(x_{n+1})$, putem aplica teorema 8.7.2 soluţiilor $v_{n+1}$ şi $v_n$, luând $c = x_{n+1}$, $s = u_{n+1}$, $s^* = u_{n+1} - h_n T(x_n, u_n; h_n)$ (conform lui (8.7.20)) şi astfel obţinem

$$\|v_{n+1}(x) - v_n(x)\| \leq h_n e^{L|x-x_n|} \|T(x_n, u_n; h_n)\|, \quad n = \overline{0, N-1}. \tag{8.7.21}$$

Acum

$$\sum_{n=0}^{N-1} [v_{n+1}(x) - v_n(x)] = v_N(x) - v_0(x) = v_N(x) - y(x), \tag{8.7.22}$$

şi deoarece $v_N(x_N) = u_N$, luând $x = x_N$, obţinem din (8.7.21) şi (8.7.22) că

$$\begin{aligned} \|u_N - y(x_N)\| &\leq \sum_{n=0}^{N-1} \|v_{n+1}(x_N) - v_n(x_N)\| \\ &\leq \sum_{n=0}^{N-1} h_n e^{L|x_N - x_{n+1}|} \|T(x_n, u_n; h_n)\| . \end{aligned}$$

De aceeea, dacă ne asigurăm că

$$\|T(x_n, u_n; h_n)\| \leq \varepsilon_T, \qquad n = \overline{0, N-1}, \tag{8.7.23}$$

atunci

$$\|u_N - y(x_N)\| \leq \varepsilon_T \sum_{n=0}^{N-1} (x_{n+1} - x_n) e^{L|x_N - x_{n+1}|}.$$

Interpretând suma din dreapta ca o sumă Riemann pentru o integrală definită, obţinem în final aproximarea

$$\|u_N - y(x_N)\| \leq \varepsilon_T \int_a^b e^{L(b-x)} dx = \frac{\varepsilon_T}{L} \left( e^{L(b-a)} - 1 \right).$$

Astfel, cunoaşterea unei estimări pentru $L$ ne va permite să găsim un $\varepsilon_T$

$$\varepsilon_T = \frac{L}{e^{L(b-a)} - 1} \varepsilon, \tag{8.7.24}$$

care să ne garanteze o eroare $\|u_N - y(x_N)\| \leq \varepsilon$. Ceea ce are loc pentru întreaga grilă pe $[a, b]$ are loc, desigur, pentru orice grilă pe subintervalul $[a, x]$, $a \leq x \leq b$. Astfel, în principiu, dându-se precizia dorită $\varepsilon$ pentru soluţia $y(x)$, putem determina un „nivel de toleranţă“ $\varepsilon_T$ (din (8.7.24)) şi putem asigura precizia dorită păstrând eroarea locală de trunchiere sub limita $\varepsilon_T$ (a se compara cu (8.7.23)). De notat că dacă $L \to 0$, avem $\varepsilon_T \to \varepsilon/(b-a)$. Această valoare limită pentru $\varepsilon_T$ ar fi adecvată pentru o problemă de cuadratură, dar nu pentru o ecuaţie diferenţială veritabilă, unde $\varepsilon_T$, în general, trebuie ales mult mai mic decât eroarea finală $\varepsilon$.

Consideraţii ca acestea motivează următorul mecanism de control al pasului: fiecare pas de integrare (de la $x_n$ la $x_{n+1} = x_n + h_n$) constă din următoarele părţi:

1. Estimăm $h_n$.

2. Se calculează $u_{n+1} = u_n + h_n \Phi(x_n, u_n; h_n)$ şi $r(x_n, u_n; h_n)$.

3. Se testează dacă $h_n^p\|r(x_n, u_n; h_n)\| \leq \varepsilon_T$. Dacă testul este satisfăcut se trece la pasul următor. Dacă nu, repetăm pasul cu un $h_n$ mai mic, până când testul este satisfăcut.

Pentru a estima $h_n$, presupunem că $n \geq 1$, astfel ca estimatorul din pasul precedent, $r(x_{n-1}, u_{n-1}; h_{n-1})$ (sau cel puţin norma sa) să fie disponibil. Atunci, neglijând termenul $O(h)$,

$$\|\tau(x_{n-1}, u_{n-1}\| \approx \|r(x_{n-1}, u_{n-1}; h_{n-1})\|,$$

şi deoarece $\tau(x_n, u_n) \approx \tau(x_{n-1}, u_{n-1})$, în plus.

$$\|\tau(x_n, u_n)\| \approx \|r(x_{n-1}, u_{n-1}; h_{n-1})\|.$$

Ceea ce dorim este

$$\|\tau(x_n, u_n)\|h_n^p \approx \theta\varepsilon_T,$$

unde $\theta$ este un factor de siguranţă (să zicem $\theta = 0.8$). Eliminând $\tau(x_n, u_n)$ găsim

$$h_n \approx \left\{ \frac{\theta\varepsilon_T}{\|r(x_{n-1}, u_{n-1}, h_{n-1})\|} \right\}^{1/p}.$$

De notat că din pasul precedent avem

$$h_{n-1}^p\|r(x_{n-1}, u_{n-1}; h_{n-1})\| \leq \varepsilon_T,$$

aşa că

$$h_n \geq \theta^{1/p} h_{n-1}$$

şi tendinţa este de creştere a pasului.

Dacă $n = 0$, procedăm la fel, alegând o valoare iniţială $h_0^{(0)}$ a lui $h_0$ şi calculăm $r(x_0, y_0; h_0^{(0)})$ pentru a obţine

$$h_0^{(1)} = \left\{ \frac{\theta\varepsilon_T}{r(x_0, y_0; h_0^{(0)})} \right\}^{1/p}.$$

Procesul se poate repeta odată sau de două ori pentru a obţine estimarea finală a lui $h_0$ şi $r(x_0, y_0; h_0^{(0)})$.

Pentru o descriere sintetică a metodelor Runge-Kutta cu pas variabil tabela Butcher se completează cu o linie suplimentară care serveşte la calculul lui $\Phi^*$ (şi deci a lui $r(x, y; h)$):

$$\begin{array}{c|cccc|c}
\mu_1 & \lambda_{11} & \lambda_{12} & \dots & \lambda_{1r} & \\
\mu_2 & \lambda_{21} & \lambda_{22} & \dots & \lambda_{2r} & \\
\vdots & \vdots & \vdots & \dots & \vdots & \\
\mu_r & \lambda_{r1} & \lambda_{r2} & \dots & \lambda_{rr} & \\
\hline
 & \alpha_1 & \alpha_2 & \dots & \alpha_r & \\
 & \alpha_1^* & \alpha_2^* & & \alpha_r^* & \alpha_{r+1}^*
\end{array}$$

Ca exemplu, în tabela 8.3 dăm tabela Butcher pentru o metodă de ordinul 2-3. Pentru deducerea elementelor tabelei a se consulta [41, paginile 451–452].

| $\mu_j$ | $\lambda_{ij}$ | | | |
|---|---|---|---|---|
| $0$ | $0$ | | | |
| $\frac{1}{4}$ | $\frac{1}{4}$ | $0$ | | |
| $\frac{27}{40}$ | $-\frac{189}{800}$ | $\frac{729}{800}$ | $0$ | |
| $1$ | $\frac{214}{891}$ | $\frac{1}{33}$ | $\frac{650}{891}$ | $0$ |
| $\alpha_i$ | $\frac{214}{891}$ | $\frac{1}{33}$ | $\frac{650}{891}$ | $0$ |
| $\alpha_i^*$ | $\frac{533}{2106}$ | $0$ | $\frac{800}{1053}$ | $-\frac{1}{78}$ |

Tabela 8.3: O pereche 2-3

Tabela 8.4 este tabela Butcher pentru metoda Bogacki-Shampine [6]. Ea stă la baza rezolvitorului `ode23` din MATLAB.

| $\mu_j$ | $\lambda_{ij}$ | | | |
|---|---|---|---|---|
| $0$ | $0$ | | | |
| $\frac{1}{2}$ | $\frac{1}{2}$ | $0$ | | |
| $\frac{3}{4}$ | $0$ | $\frac{3}{4}$ | $0$ | |
| $1$ | $\frac{2}{9}$ | $\frac{3}{9}$ | $\frac{4}{9}$ | $0$ |
| $\alpha_i$ | $\frac{2}{9}$ | $\frac{3}{9}$ | $\frac{4}{9}$ | $0$ |
| $\alpha_i^*$ | $\frac{7}{24}$ | $\frac{1}{4}$ | $\frac{1}{3}$ | $\frac{1}{8}$ |

Tabela 8.4: Tabela Butcher pentru metoda Bogacki-Shampine

Un alt exemplu important este DORPRI5 sau RK5(4)7FM, o pereche cu ordinele 4-5 şi cu 7 stadii (tabela 8.5). Aceasta este o pereche foarte eficientă, ea stând la baza rezolvitorului `ode45` din MATLAB, dar şi a altor rezolvitori importanţi.

Algoritmul 8.2 încearcă să dea sugestii pentru implementarea unei metode Runge-Kutta cu pas variabil când se cunoaşte tabela Butcher. $ttol$ este produsul dintre $tol$ si factorul de siguranţă (0.8 sau 0.9).

**Algoritmul 8.2** Fragment de pseudocod ce ilustrează implementarea unei metode RK cu pas variabil

$done := \mathbf{false}$;
**loop**
  $K_{1,:} := f(x, y)$;
  **for** $i = 2$ **to** $s$ **do**
    $w := y + hK_{:,1:i-1}\lambda^T_{i,1:i-1}$;
    $K_{:,i} := f(x + \mu_i h, w)$;
  **end for**
  $\delta := h \max\left(\left|(\alpha^* - \alpha)^T K\right|\right)$; {estimarea erorii}
  $\beta := (\delta / ttol)^{1/(1+p)}$; {raport lung. pas}
  **if** $\delta < tol$ **then**
    {acceptare pas}
    $y := y + h(K\alpha^T)$; {actualizare $y$}
    $x := x + h$;
    **if** $done$ **then**
      EXIT {terminare şi ieşire}
    **end if**
    $h := h / \max(\beta, 0.1)$; {predicţie pas următor}
    **if** $x + h > x_{end}$ **then**
      $x := x_{end} - x$; {reducere pas la capăt}
      $done := \mathbf{true}$;
    **end if**
  **else**
    {respingere pas}
    $h := h / \min(\beta, 10)$; {reducere pas}
    **if** $done$ **then**
      $done := \mathbf{false}$;
    **end if**
  **end if**
**end loop**

| $\mu_j$ | $\lambda_{ij}$ | | | | | | |
|---|---|---|---|---|---|---|---|
| $0$ | $0$ | | | | | | |
| $\frac{1}{5}$ | $\frac{1}{5}$ | $0$ | | | | | |
| $\frac{3}{10}$ | $\frac{3}{40}$ | $\frac{9}{40}$ | $0$ | | | | |
| $\frac{4}{5}$ | $\frac{44}{45}$ | $-\frac{56}{15}$ | $\frac{32}{9}$ | $0$ | | | |
| $\frac{8}{9}$ | $\frac{19372}{6561}$ | $-\frac{25360}{2187}$ | $\frac{64448}{6561}$ | $-\frac{212}{729}$ | $0$ | | |
| $1$ | $\frac{9017}{3168}$ | $-\frac{355}{33}$ | $\frac{46732}{5247}$ | $\frac{49}{176}$ | $-\frac{5103}{18656}$ | $0$ | |
| $1$ | $\frac{35}{384}$ | $0$ | $\frac{500}{1113}$ | $\frac{125}{192}$ | $-\frac{2187}{6784}$ | $\frac{11}{84}$ | $0$ |
| $\alpha_i$ | $\frac{35}{384}$ | $0$ | $\frac{500}{1113}$ | $\frac{125}{192}$ | $-\frac{2187}{6784}$ | $\frac{11}{84}$ | $0$ |
| $\alpha_i^*$ | $\frac{5179}{57600}$ | $0$ | $\frac{7571}{16695}$ | $\frac{393}{640}$ | $-\frac{92097}{339200}$ | $\frac{187}{2100}$ | $\frac{1}{40}$ |

Tabela 8.5: Perechea inclusă RK5(4)7FM (DORPRI5)

# CAPITOLUL 9

# Aproximări în mai multe variabile

## Cuprins



Problema aproximării funcţiilor de mai multe variabile şi cea a aproximării integralelor multiple sunt deosebit de importante atât din punct de vedere teoretic cât şi practic, ele intervenind atât în probleme matematice abstracte cât şi în modele ale diverselor fenomene sau procese din natură şi societate.

Vom nota cu $\mathbb{P}_m^n$ mulţimea polinoamelor în $n$ variabile şi de grad global cel mult $m$. Fie $D \in \mathbb{R}^n$, $f : D \to \mathbb{R}$. Fie de asemenea $F_1 f$, $F_2 f$, ..., $F_p f$ informaţii despre $f$ (de regulă valori ale funcţiei sau ale unor derivate parţiale ale acestora). Se pune problema ca pe baza informaţiilor $F_k$, $k = \overline{1,p}$, să se determine o funcţie $F$, astfel încât $f \approx F$, într-un sens precizat, în domeniul $D$.

De exemplu, dacă $P_i \in D$, $i = \overline{1,s}$, iar $F_i f = f(P_i)$, $i = \overline{1,s}$ se ajunge la o problemă de interpolare polinomială.

Dacă $F_i f = f(P_i)$, $i = \overline{1,s}$, iar $f \in \mathbb{P}_m^n$ se determină astfel încât

$$\sum_{i=1}^{s} [F(p_i) - f(P_i)]^2 \to \min$$

se obţine o problemă de aproximare discretă în sensul celor mai mici pătrate.

# 9.1. Aproximarea funcţiilor de mai multe variabile pe un domeniu rectangular

Ne vom ocupa în continuare de aproximarea funcţiilor definite pe un domeniu rectangular şi de extinderea unor procedee unidimensionale.

Fie $D = \prod_{k=1}^{n}[a_k, b_k] \subset \mathbb{R}^n$ şi $\mathcal{F}_n$ o mulţime de funcţii de $n$ variabile independente definite pe $D$. Notăm prin $P_i : \mathcal{F}_n \to \mathcal{G}_i$, $i = \overline{1,n}$ o mulţime de $n$ proiectori ce acţionează asupra funcţiei $f \in \mathcal{F}_n$, fiecare în raport cu variabila $x_i$, iar prin $R_i = I - P_i$, unde $I$ este operatorul identic, operatorul rest corespunzător. Operatorul $R_i : \mathcal{F}_n \to \mathcal{H}_i$ este de asemenea proiector.

Cu ajutorul operaţiei de *produs tensorial* (compunere) se pot construi operatorii $P_iP_j$, $i, j = \overline{1,n}$, care, în caz că oricare doi comută sunt de asemenea proiectori, deoarece $(P_iP_j)^2 = P_iP_j$. Deoarece suma a doi proiectori $P_i + P_j$ nu este proiector $((P_i + P_j)^2 \neq P_i + P_j)$, se lucrează cu *suma booleană* $P_i \oplus P_j = P_i + P_j - P_iP_j$, care este proiector.

Datorită proprietăţii de asociativitate a produsului şi a sumei booleene, aceste operaţii se pot extinde la trei sau mai mulţi operatori. Putem defini proiectorii

$$P := P_1P_2 \dots P_n,$$
$$S := P_1 \oplus P_2 \oplus \dots \oplus P_n = \sum_{i=1}^{n} \tilde{P}_i,$$

cu

$$\tilde{P}_1 = P_1, \qquad \tilde{P}_i = P_i - \left(\sum_{k=1}^{i-1} \tilde{P}_k\right) P_i, \qquad i = \overline{2,n},$$

adică produsul şi suma booleană a celor $n$ proiectori consideraţi. De exemplu, pentru trei proiectori avem

$$P_1P_2P_3 = P_1(P_2P_3),$$
$$P_1 \oplus P_2 \oplus P_3 = P_1 + P_2 + P_3 - P_1P_2 - P_1P_3 - P_2P_3 + P_1P_2P_3.$$

Fie $\mathcal{P}$ mulţimea formată din proiectorii $P_1, \dots, P_n$ şi toţi proiectorii obţinuţi din aceştia cu ajutorul operaţiilor de produs şi sumă booleană.

Imaginea $Qf$ a unei funcţii $f \in \mathcal{F}_n$ prin orice element $Q \in \mathcal{P}$ poate fi considerată o aproximantă a lui $f$. De exemplu, dacă $P_i$ este operatorul de interpolare Lagrange $L_{m_i}^{x_i}$ relativ la nodurile $x_{i_0}, \dots, x_{i_{m_i}}$ ce acţionează asupra variabilei $x_i$ atunci

$$\left(L_{m_i}^{x_i} f\right) = \sum_{k=0}^{m_i} \ell_k(x_i) f(x_1, \dots, x_{i-1}, x_{i_k}, x_{i+1}, \dots, x_{i_{m_i}}),$$

unde

$$\ell_k(x_i) = \frac{(x_i - x_{i_0}) \dots (x_i - x_{i_{k-1}})(x_i - x_{i_{k+1}}) \dots (x_i - x_{i_{m_i}})}{(x_{i_k} - x_{i_0}) \dots (x_{i_k} - x_{i_{k-1}})(x_{i_k} - x_{i_{k+1}}) \dots (x_{i_k} - x_{i_{m_i}})},$$

este o aproximantă a funcţiei $f$. Mai mult, putem scrie una din expresiile termenului rest din această aproximare. Astfel, utilizând forma cu diferenţe divizate pentru rest, avem

$$\left(R_{m_i}^{x_i} f\right) = u(x_i)[x_i, x_{i_0}, \dots x_{i_{m_i}}, f(x_1, \dots, x_{i-1}, \cdot, x_{i+1}, \dots, x_{i_{m_i}})].$$

Operatorul de diferenţă divizată acţionează asupra funcţiei $f$ în raport cu variabila $x_i$, iar $u(x_i) = (x_i - x_{i_0}) \dots (x_i - x_{i_{m_i}})$.

Pentru a compara aproximantele generate de elementele lui $\mathcal{P}$ se defineşte pe $\mathcal{P}$ o relaţie de ordine parţială.

**Definiţia 9.1.1** *Pentru $X, Y \in \mathcal{P}$, $X \leq Y \Leftrightarrow XY = X$.*

Dacă produsul a oricare doi proiectori este comutativ, adică $P_i P_j = P_i P_j$, $i \neq j$, atunci $(\mathcal{P}, \leq)$ este o latice distributivă. Din relaţiile

$$P = P_1 \dots P_n \leq X, \qquad X \in \mathcal{P}$$

şi

$$Y \leq S = P_1 \oplus \dots \oplus P_n, \qquad Y \in \mathcal{P},$$

rezultă că $P$ este element minimal, iar $S$ element maximal al laticei $(\mathcal{P}, \leq)$.

**Definiţia 9.1.2** *Fie $f \in \mathcal{F}_n$. Aproximanta $Pf$ a funcţiei $f$ se numeşte* algebric minimală, *iar aproximanta $Sf$ se numeşte* algebric maximală.

Pentru exemplificare, fie $n = 2$ şi $P_1 = L_r^x$, $P_2 = L_s^y$ operatorii de interpolare Lagrange unidimensionali relativ la nodurile $x_0, \dots, x_r$, respectiv $y_0, \dots, y_s$. Dacă $f \in \mathcal{F}_2$, atunci aproximanta algebric minimală este

$$\left(L_r^x L_s^y f\right)(x, y) = \sum_{i=0}^{r} \sum_{j=0}^{s} \ell_i(x) \tilde{\ell}_j(y) f(x_i, y_j), \tag{9.1.1}$$

iar cea algebric maximală este

$$\begin{aligned} \left(L_r^x \oplus L_s^y f\right)(x, y) = &\sum_{i=0}^{r} \ell_i(x) f(x_i, y) + \sum_{j=0}^{s} \tilde{\ell}_j(y) f(x, y_j) \\ &- \sum_{i=0}^{r} \sum_{j=0}^{s} \ell_i(x) \tilde{\ell}_j(y) f(x_i, y_j), \end{aligned} \tag{9.1.2}$$

unde $\ell_i(x)$ şi $\tilde{\ell}_j(y)$ sunt polinoamele fundamentale Lagrange corespunzătoare.

Prin verificare directă pentru produs se obţine

$$\left(L_r^x L_s^y f\right)(x_i, y_j) = f(x_i, y_j), \qquad i = \overline{0, r},\ j = \overline{0, s},$$

iar pentru suma booleană

$$\begin{aligned} \left(L_r^x \oplus L_s^y f\right)(x_i, y) &= f(x_i, y), && i = \overline{0, r}, \quad y \in [a_2, b_2], \\ \left(L_r^x \oplus L_s^y f\right)(x, y_j) &= f(x, y_j), && j = \overline{0, s}, \quad x \in [a_1, b_1]. \end{aligned}$$

Să considerăm grila $\{(x_i, y_j) : i = \overline{0, r}, j = \overline{0, s}\}$. Interpolarea algebric minimală

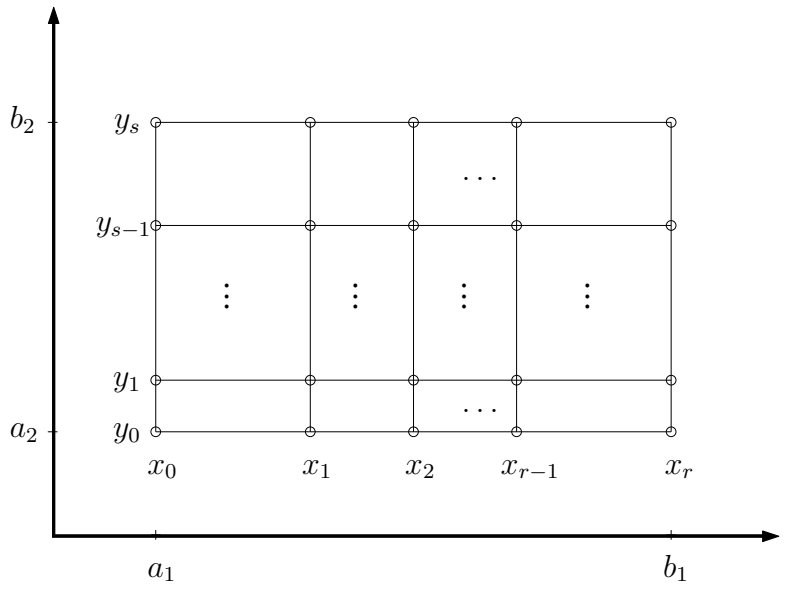



Figura 9.1: Interpretarea geometrică a interpolanţilor produs tensorial şi sumă booleană

este o interpolare discretă (punctuală); ea reproduce valorile funcţiei numai pe noduri. Interpolarea algebric maximală este o *interpolare transfinită* sau *interpolare blending*; funcţia interpolatoare reproduce valorile lui $f$ pe segmentele $\{(x_i, y) \in \mathbb{R}^2 : a_2 \leq y \leq b_2\}$, $i = \overline{0, r}$ şi $\{(x, y_j) \in \mathbb{R}^2 : a_1 \leq x \leq b_1\}$, $j = \overline{0, s}$ (figura 9.1).

Revenind la operatorii rest $R_i = I - P_i$, $i = \overline{1, n}$ se observă că mulţimea formată din $R_i$ şi operatorii obţinuţi din aceştia prin produs tensorial şi sumă booleană, formează şi ei în raport cu relaţia „$\leq$” (dată în definiţia 9.1.1) tot o latice (dacă produsul este comutativ). În această latice $R_1 R_2 \ldots R_n$ este element minimal, iar $R_1 \oplus R_2 \oplus \cdots \oplus R_n$ este element maximal. Avem de asemenea următoarele descompuneri pentru operatorul identic

$$\begin{aligned} I &= P_1 \ldots P_n + R_1 \oplus \cdots \oplus R_n \\ I &= P_1 \oplus \cdots \oplus P_n + R_1 \ldots R_n, \end{aligned} \tag{9.1.3}$$

numite *descompunere minimală* şi respectiv *maximală*. Dacă $f \in \mathcal{F}_n$, pe baza descompunerilor anterioare avem formulele de aproximare

$$f = P_1 \dots P_n f + R_1 \oplus \dots \oplus R_n f \tag{9.1.4}$$
$$f = P_1 \oplus \dots \oplus P_n f + R_1 \dots R_n f, \tag{9.1.5}$$

numite *formulă de aproximare algebric minimală* şi respectiv *formulă de aproximare algebric maximală*.

Pentru exemplificare să considerăm din nou cazul bidimensional şi operatorii de interpolare Lagrange $L_r^x$ şi $L_s^y$. Operatorii rest corespunzători vor fi $R_r^x = I - L_r^x$ şi $R_s^y = I - L_s^y$. Pentru formula algebric minimală se va obţine

$$f = L_r^x L_s^y + R_r^x \oplus R_s^y, \tag{9.1.6}$$

iar pentru cea algebric maximală

$$f = L_r^x \oplus L_s^y + R_r^x R_s^y. \tag{9.1.7}$$

Cunoscând expresiile termenului rest al formulei de interpolare Lagrange şi presupunând că există toate derivatele parţiale ale lui $f$ necesare şi că acestea satisfac condiţiile de continuitate cerute, obţinem

$$\begin{aligned}\left(R_r^x \oplus R_s^y f\right)(x,y) = &\frac{u_r(x)}{(r+1)!}\frac{\partial^{r+1}}{\partial x^{r+1}} f(\xi_1, y) + \frac{u_s(y)}{(s+1)!}\frac{\partial^{s+1}}{\partial y^{s+1}} f(x, \eta_1) \\ &- \frac{u_r(x)u_s(y)}{(r+1)!(s+1)!}\frac{\partial^{r+s+2}}{\partial x^{r+1}\partial y^{s+1}} f(\xi_2, \eta_2),\end{aligned}$$

cu $\xi_1, \xi_2 \in [\alpha_1, \alpha_2]$, $\eta_1, \eta_2 \in [\beta_1, \beta_2]$,

$$\left(R_r^x R_s^y f\right)(x,y) = \frac{u_r(x)u_s(y)}{(r+1)!(s+1)!}\frac{\partial^{r+s+2}}{\partial x^{r+1}\partial y^{s+1}} f(\xi_3, \eta_3),$$

cu $\xi_3 \in [\alpha_1, \alpha_2]$, $\eta_3 \in [\beta_1, \beta_2]$, $\alpha_1 = \min\{x, x_0, \dots, x_r\}$, $\alpha_2 = \max\{x, x_0, \dots, x_r\}$, $\beta_1 = \min\{y, y_0, \dots, y_s\}$, $\beta_2 = \max\{y, y_0, \dots, y_s\}$. De notat că termenul rest se poate exprima şi sub formă integrală şi cu ajutorul diferenţelor divizate.

**Exemplul 9.1.3.** Dacă luăm $r = s = 1$ şi nodurile 0 şi 1 după fiecare coordonată, operatorii Lagrange unidimensionali sunt:

$$L_1^x(x,y) = (1-x)f(0,y) + xf(1,y)$$
$$L_1^y(x,y) = (1-y)f(x,0) + yf(x,1),$$

iar pentru operatorii produs tensorial şi sumă booleană se obţin expresiile

$$(L_1^x L_1^y)(x,y) = (1-x)(1-y)f(0,0) + (1-x)yf(0,1) + x(1-y)f(1,0) + xyf(1,1)$$

şi respectiv

$$(L_1^x \oplus L_1^y)(x,y) = (1-x)f(0,y) + xf(1,y) + (1-y)f(x,0) + yf(x,1) - (1-x)(1-y)f(0,0) - (1-x)yf(0,1) - x(1-y)f(1,0) + xyf(1,1).$$

Prima aproximantă reproduce valorile funcţiei doar în vârfurile pătratului $[0,1]\times[0,1]$, pe când ce-a de-a doua reproduce valorile funcţiei pe frontiera pătratului. Dacă există derivatele parţiale până la ordinele necesare, expresiile corespunzătoare pentru rest sunt

$$(R_P f)(x,y) = (R_1^x R_1^y)(x,y) = \frac{x(x-1)}{2}\frac{\partial^2}{\partial x^2}f(\xi_1,y) + \frac{y(y-1)}{2}\frac{\partial^2}{\partial y^2}f(x,\eta_1) - \frac{x(x-1)}{2}\frac{y(y-1)}{2}\frac{\partial^4}{\partial x^2 y^2}f(\xi_2,\eta_2).$$

şi respectiv

$$(R_S f)(x,y) = (R_1^x \oplus R_1^y)(x,y) = \frac{x(x-1)}{2}\frac{y(y-1)}{2}\frac{\partial^4}{\partial x^2 y^2}f(\xi,\eta),$$

unde $\xi,\eta,\xi_1,\eta_1,\xi_2,\eta_2 \in (0,1)$. Dacă $f$ este de clasă $C^{2,2}$, au loc delimitările

$$\|R_P f\|_\infty \le \frac{1}{8}\left[\left\|\frac{\partial^2}{\partial x^2}f\right\|_\infty + \left\|\frac{\partial^2}{\partial y^2}f\right\|_\infty + \frac{1}{8}\left\|\frac{\partial^4}{\partial x^2 y^2}f\right\|_\infty\right],$$
$$\|R_S f\|_\infty \le \frac{1}{64}\left\|\frac{\partial^4}{\partial x^2 y^2}f\right\|_\infty. \qquad \diamondsuit$$

**Exemplul 9.1.4.** Să considerăm funcţia $f:[-2,2]\times[-2,2]\to\mathbb{R}$, $f(x,y) = xe^{-x^2-y^2}$. Graficul ei este dat în figura 9.2. Graficele aproximantelor algebric minimală, respectiv algebric maximală de tip Lagrange şi ale resturilor corespunzătoare aper în figura 9.3. S-au considerat cinci noduri echidistante după fiecare coordonată, $x_k, y_k = -2+k$, $k = \overline{0,4}$. $\diamondsuit$

În concluzie, formula de aproximare generată de produsul operatorilor $P_1, \ldots, P_n$ este algebric minimală, iar formula generată de suma booleană a acestor operatori este algebric maximală, proprietăţi datorate calităţii operatorilor produs şi sumă booleană de a fi element minimal, respectiv maximal în laticea $(\mathcal{P}, \le)$.

Se va da în continuare şi o altă interpretare a celor două formule de aproximare extremală, având în vedere ordinul de aproximare al operatorilor unidimensionali $P_i$,

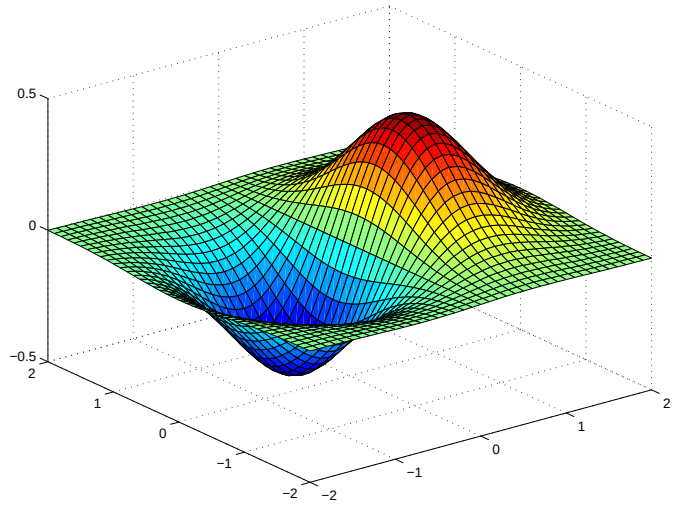
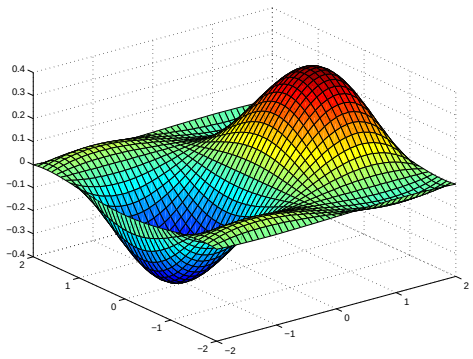
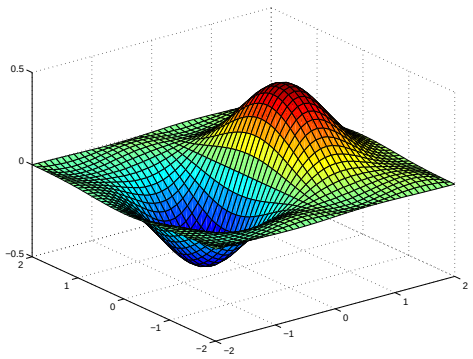


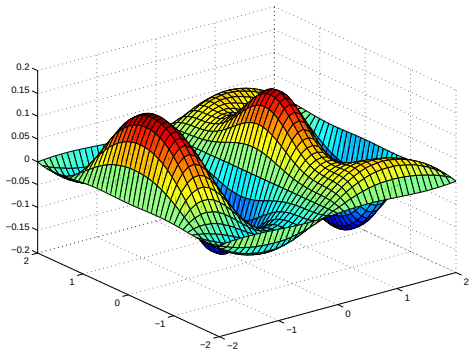

Figura 9.2: Graficul funcţiei $f : [-2,2]\times[-2,2] \to \mathbb{R}$, $f(x,y) = xe^{-x^2-y^2}$ din exemplul 9.1.4



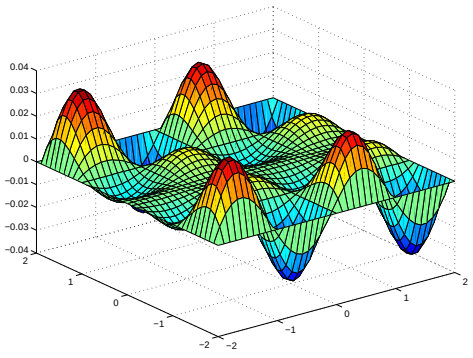
(a) produs tensorial



(b) sumă booleană



(c) rest produs tensorial



(d) rest sumă booleană

Figura 9.3: Aproximarea algebric minimală (figura 9.3(a)), cea maximală (figura 9.3(b)) şi resturile corespunzătoare (figura 9.3(c) şi respectiv 9.3(d))

$i = \overline{1,n}$ de la care se porneşte. Această interpretare va permite extinderea procedeelor de mai sus şi la alţi operatori care nu sunt proiectori.

Dacă notăm prin $m_i$ ordinul de aproximare al operatorului $P_i$, $i = \overline{1,n}$, atunci din (9.1.4) rezultă că ordinul de aproximare al operatorului produs tensorial este

$$ord(P) = \min\{ord(P_1), \dots ord(m_n)\}, \tag{9.1.8}$$

iar din (9.1.5) rezultă că ordinul de aproximare al operatorului sumă booleană este

$$ord(S) = ord(P_1) + \dots ord(P_n). \tag{9.1.9}$$

Pentru două variabile formula de aproximare algebric minimală (9.1.6) are ordinul de aproximare $\min(r,s)+1$, în timp ce ordinul formulei de interpolare algebric maximale (9.1.7) este $r+s+2$.

Prin urmare, proprietăţile de extremalitate ale celor doi operatori sunt caracterizate şi prin ordinul de aproximare, operatorul produs având ordinul de aproximare minim, iar operatorul sumă booleană având ordinul de aproximare maxim în mulţimea $\mathcal{P}$ a tuturor operatorilor generaţi din $P_1, \dots, P_n$ cu ajutorul celor două operaţii.

Deşi operatorul sumă booleană are un ordin mare de aproximare, în expresia sa

$$Sf = (P_1 + \dots + P_n - P_1P_2 - \dots - P_{n-1}P_n + \dots P_1P_2 \dots P_n)f, \tag{9.1.10}$$

vor interveni termeni care sunt funcţii de $n-1$, $n-2$, $\dots$, $1$ variabile. O astfel de formulă este aplicabilă doar atunci când se cunosc valorile funcţiei $f$ sau ale unor derivate parţiale ale ei pe hiperfeţele lui $D$ sau pe secţiuni ale domeniului $D$ paralele cu aceste hiperfeţe. Acesta este un dezavantaj major al aproximării blending. El poate fi înlăturat aplicând funcţiei $f$ pe hipersuprafeţele respective ale lui $D$ alţi operatori liniari de aproximare. Continuând acest prodeu din aproape în aproape se ajunge după cel mult $n-1$ etape la o aproximare scalară (numerică), adică la o aproximare ce conţine numai valori ale lui $f$ pe puncte ale lui $D$. Evident că restul unei astfel de formule de aproximare, obţinută din formula blending conţine mai mulţi termeni. Pentru ilustrare, să considerăm cazul unei funcţii de două variabile, definită pe $[a_1,b_1]\times[a_2,b_2]$. Considerând operatorii $P_1^1$, $P_2^1$, în care indicele superior indică numărul nivelului (etapei) de aproximare, obţinem la prima etapă

$$f = (P_1^1 + P_2^1 - P_1^1P_2^1)f + R_1^1R_2^1f, \tag{9.1.11}$$

unde $P_1^1 f$ conţine pe $x_2$ ca variabilă liberă, iar $P_2^1$ pe $x_1$. Aplicând pe al doilea nivel lui $P_1 f$ operatorul $P_2^2$, iar lui $P_2^1 f$ operatorul $P_1^2$, obţinem formula de aproximare scalară:

$$f = (P_1^1P_2^2 + P_1^2P_2^1 - P_1^1P_2^1)f + (P_2^1R_1^2 + P_1^1R_2^2 + R_1^1R_2^1)f. \tag{9.1.12}$$

Restul acestei formule conţine trei termeni. Singura proprietate a tuturor operatorilor folosiţi este comutativitatea produsului. Avem mai multe posibilităţi de alegere a operatorilor de pe al doilea nivel. Ei depind în primul rând de informaţiile disponibile asupra

lui $f$. O cale naturală de alegere este aceea ca ordinul de aproximare al formulei originale (9.1.11) să fie păstrat. O astfel de formulă de aproximare se va numi *formulă consistentă*. De exemplu, alegând în (9.1.12) pe post de $P_1^1$ şi $P_2^1$ operatorii de interpolare Lagrange $L_1^x$ şi $L_2^y$ relativi la nodurile $a_1, b_1$ şi $a_2, b_2$, iar în locul lui $P_1^2$ şi $P_2^2$ operatorii de interpolare Hermite $H_3^x$ şi $H_4^y$ relativi la nodurile duble $a_1, b_1$ şi respectiv $a_2$ triplu şi $b_2$ dublu se obţine

$$f = (L_1^x H_4^y + H_3^x L_1^y - L_1^x L_1^y)f + (L_1^y R_3^x + L_1^x R_4^y + R_1^x R_1^y)f. \tag{9.1.13}$$

Cum ordinele de aproximare ale operatorilor $L_1$, $H_3$ şi $H_4$ sunt 2, 4 şi 5, rezultă că ordinul formulei de aproximare blending

$$f = L_1^x \oplus L_2^y + R_1^x R_2^y f \tag{9.1.14}$$

este 4, iar ordinul de aproximare al lui (9.1.13) este tot 4, căci

$$\begin{aligned}\|(L_1^y R_3^x + L_1^x R_4^y + R_1^x R_1^y)f\| &\le \|R_3^x f\| + \|R_4^y f\| + \|R_1^x R_2^y f| = \\ &= c_1(b_1 - a_1)^4 + c_2(b_2 - a_2)^5 + c_3(b_1 - a_1)^2(b_2 - a_2)^2 \\ &= (c_1 + c_2 h + c_3)h^4, \qquad \text{unde } h = b_1 - a_1 = b_2 - a_2.\end{aligned}$$

Deci (9.1.13) este consistentă.

Deoarece ordinul de aproximare al formulei iniţiale nu poate fi mărit, este preferabil ca operatorii folosiţi în următoarele nivele de aproximare să fie astfel aleşi încât termenii din expresia restului formulei finale să aibă acelaşi ordin de mărime. Formula numerică astfel obţinută se va numi *omogenă*.

Formula (9.1.13) nu este omogenă deoarece $\|R_3^x f\| = O(h^4)$, $\|R_1^x R_1^y\| = O(h^4)$, dar $\|R_4^y f\| = O(h^5)$. Pentru a deveni omogenă trebuie ca operatorul $H_4^y$ folosit pe al doilea nivel să fie înlocuit printr-un operator cu ordinul de aproximare 4, de exemplu $H_3^y$. Astfel, din formula originală (9.1.14), am dedus următoarea formulă numerică omogenă

$$f = (L_1^x H_3^y + H_3^x L_1^y - L_1^x L_1^y)f + (L_1^y R_3^x + L_1^x R_3^y + R_1^x R_1^y)f.$$

## 9.2. Integrarea numerică a funcţiilor de mai multe variabile

Fie $D \subseteq \mathbb{R}^n$, funcţia $f : D \to \mathbb{R}$, punctele $P_i \in D$, $i = \overline{0, m}$ şi $w$ o funcţie pondere nenegativă, definită pe $D$.

**Definiţia 9.2.1** *Formula*

$$\int \cdots \int_D w(x_1, \ldots, x_n) f(x_1, \ldots, x_n)\, \mathrm{d}x_1 \ldots \mathrm{d}x_n = \sum_{i=0}^{m} A_i f(P_i) + R_m f \tag{9.2.1}$$

*se numeşte* formulă de integrare numerică *a funcţiei* $f$ *sau* formulă de cubatură. *Parametrii* $A_i$ *se numesc* coeficienţii *formulei*, *punctele* $P_i$ *se numesc* nodurile *ei*, *iar* $R_m$ termenul rest.

Problema construirii unei formule de cubatură constă în determinarea coeficienţilor şi a nodurilor ei, folosind condiţii corespunzătoare. În funcţie de condiţiile folosite, formulele de cubatură pot fi clasificate în deterministe şi nedeterministe, iar cele deterministe în formule de tip interpolator, de tip Gauss, optimale, etc.

O metodă eficientă de construire a formulelor de cubatură în cazul în care $D$ este un domeniu rectangular, constă în exprimarea parametrilor acesteia cu ajutorul coeficienţilor, respectiv a nodurilor unei formule de cuadratură unidimensionale. Pentru simplificare ne vom limita la cazul bidimensional. Fie $D = [a, b] \times [c, d]$, $\Delta_x$ diviziunea $a = x_0 < x_1 < \cdots < x_m = b$, $\Delta_y$ diviziunea $c = y_0 < y_1 < \cdots < y_n = d$, $w(x, y) = 1$, $\forall (x, y) \in D$.

În acest caz formula (9.2.1) devine

$$\int_a^b \int_c^d f(x, y)\,\mathrm{d}x\,\mathrm{d}y = \sum_{i=0}^{m} \sum_{j=0}^{n} A_{ij} f(x_i, y_j) + R_{m,n}(f). \tag{9.2.2}$$

Se poate obţine o astfel de formulă pornind de la formula de interpolare Lagrange (bidimensională)

$$f = L_m^x L_n^y f + R_m^x \oplus R_n^y f.$$

Dacă $f \in C^{m+1,n+1}(D)$, prin integrare termen cu termen se obţine

$$\int_a^b \int_c^d f(x, y)\,\mathrm{d}x\,\mathrm{d}y = \sum_{i=0}^{m} \sum_{j=0}^{n} A_i B_j f(x_i, y_j) + R_{mn}(f) \tag{9.2.3}$$

unde

$$A_i = \int_a^b \ell_i(x)\,\mathrm{d}x, \qquad B_j = \int_c^d \tilde{\ell}_j(y)\,\mathrm{d}y,$$

iar

$$\begin{aligned} R_{m,n}(f) &= \int_a^b \int_c^d (R_m^x \oplus R_n^y) f(x, y)\,\mathrm{d}x\,\mathrm{d}y \\ &= \frac{1}{(m+1)!} \int_a^b \int_c^d u_m(x) f^{(m+1,0)}(\xi_x, y)\,\mathrm{d}x\,\mathrm{d}y \\ &+ \frac{1}{(n+1)!} \int_a^b \int_c^d u_n(y) f^{(0,n+1)}(x, \eta_y)\,\mathrm{d}x\,\mathrm{d}y \\ &- \frac{1}{(m+1)!} \frac{1}{(n+1)!} \int_a^b \int_c^d u_m(x) u_n(y) f^{(m+1,n+1)}(\tilde{\xi}_x, \tilde{\eta}_y)\,\mathrm{d}x\,\mathrm{d}y. \end{aligned}$$

Dacă $\Delta_x$ şi $\Delta_y$ sunt diviziuni uniforme ale intervalului $[a, b]$ şi respectiv $[c, d]$, atunci formula de cubatură (9.2.3) se numeşte de tip *Newton-Cotes*.

**Cazuri particulare** Pentru $m = n = 1$ se obţine formula de cubatură a trapezului.

$$\int_a^b \int_c^d f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \frac{(b-a)(d-c)}{4}\left[f(a,c)+f(a,d)+f(b,c)+f(b,d)\right] + R_{11}(f),$$

unde

$$R_{11}(f) = -\frac{(b-a)^3(d-c)}{12}f^{(2,0)}(\xi_1,\eta_1) - \frac{(b-a)(d-c)^3}{12}f^{(0,2)}(\xi_2,\eta_2) - \frac{(b-a)^3(d-c)^3}{144}f^{(2,2)}(\xi_3,\eta_3).$$

Pentru $m = n = 2$ se obţine formula de cubatură a lui Simpson.

$$\int_a^b \int_c^d f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \frac{(b-a)(d-c)}{36}\Bigg\{f(a,c)+f(a,d)+f(b,c)+f(b,d) + 4\left[f\left(\frac{a+b}{2},c\right)+f\left(\frac{a+b}{2},d\right)+f\left(a,\frac{c+d}{2}\right)++f\left(b,\frac{c+d}{2}\right)\right] + 16f\left(\frac{a+b}{2},\frac{b+c}{2}\right)\Bigg\} + R_{22}(f),$$

unde

$$R_{22}(f) = -\frac{(b-a)^5(d-c)}{2880}f^{(4,0)}(\xi_1,\eta_1) - \frac{(b-a)(d-c)^5}{2880}f^{(0,4)}(\xi_2,\eta_2) - \frac{(b-a)^5(d-c)^5}{2880^2}f^{(4,4)}(\xi_3,\eta_3).$$

Partiţionând intervalele $[a,b]$ şi $[c,d]$ se pot obţine formule de cubatură repetate. Vom ilustra pentru formula lui Simpson. Să presupunem că $[a,b]$ este împărţit în $m$ parţi egale, iar $[c,d]$ în $n$ părţi egale, obţinându-se o diviziune cu $mn$ dreptunghiuri. Vom împărţi fiecare dreptunghi în patru părţi egale ca în figura 9.4 (vârfurile dreptunghiului sunt indicate prin cercuri negre, iar punctele intermediare cu cercuri mai albe).

Fie

$$h = \frac{b-a}{2m}, \qquad k = \frac{d-c}{2n}.$$

Punctele diviziunii vor avea coordonatele

$$x_i = x_0 + ih, \quad x_0 = a, \quad i = \overline{0,2m}$$
$$y_j = y_0 + jk, \quad y_0 = b, \quad j = \overline{0,2n}.$$

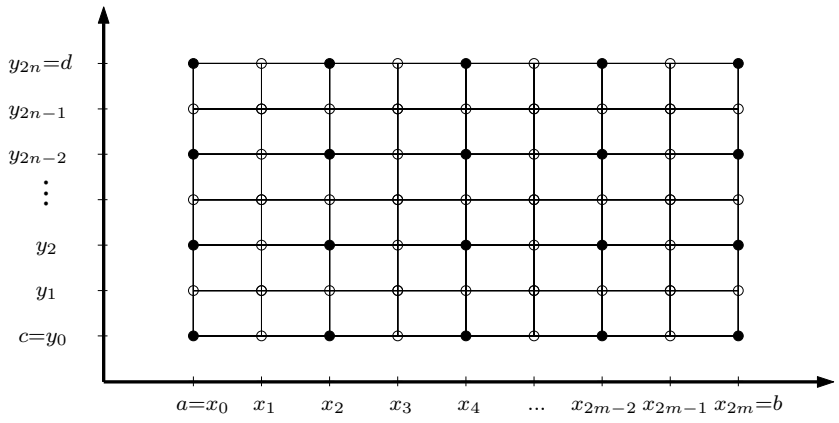



Figura 9.4: Diviziunea pentru formula repetată a lui Simpson

Introducem notaţia $f(x_i, y_j) = f_{ij}$. Aplicând formula lui Simpson fiecărui dreptunghi al diviziunii avem

$$\int_a^b \int_c^d f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \frac{hk}{9} \sum_{i=0}^{m} \sum_{j=0}^{n} \big[ f_{2i,2j} + f_{2i+2,j} + f_{2i+2,2j+2} + f_{2i,2j+2} + 4(f_{2i+1,2j} + f_{2i+2,2j+1} + f_{2i+1,2j+2} + f_{2i,2j+1}) + 16 f_{2i+1,2j+1} \big] + R_{m,n}(f).$$

Reducând termenii asemenea se obţine

$$\int_a^b \int_c^d f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \frac{hk}{9} \sum_{i=0}^{m} \sum_{j=0}^{n} \lambda_{i,j} f_{ij} + R_{m,n}(f),$$

unde $\lambda_{ij}$ sunt daţi de matricea

$$\Lambda = \begin{bmatrix} 1 & 4 & 2 & 4 & 2 & \dots & 4 & 2 & 4 & 1 \\ 4 & 16 & 8 & 16 & 8 & \dots & 16 & 8 & 16 & 4 \\ 2 & 8 & 4 & 8 & 4 & \dots & 8 & 4 & 8 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 2 & 8 & 4 & 8 & 4 & \dots & 8 & 4 & 8 & 2 \\ 4 & 16 & 8 & 16 & 8 & \dots & 16 & 8 & 16 & 4 \\ 1 & 4 & 2 & 4 & 2 & \dots & 4 & 2 & 4 & 1 \end{bmatrix}.$$

În [8, 38], pentru $f \in C^{4,4}(D)$ se dă următoarea formă a restului

$$R_{mn}(f) = -\frac{(b-a)(d-c)}{180} \left[ h^4 f^{(4,0)}(\xi_1, \eta_1) + k^4 f^{(0,4)}(\xi_2, \eta_2) \right],$$

cu $\xi_1, \eta_1, \xi_2, \eta_2 \in D$.

Se pot da formule de cubatură de tip Gauss bidimensionale. De exemplu, dacă $x_i$, $i = \overline{0, m}$ şi $y_j$, $j = \overline{0, n}$ sunt rădăcinile polinomului Legendre relativ la intervalul $[a, b]$ şi respectiv $[c, d]$, se obţine formula de cubatură Gauss-Legendre, în care

$$\begin{aligned} A_i &= \frac{[(m+1)!]^4(b-a)^{2m+3}}{[(2m+2)!]^2(x_i-a)(b-x_i)[u'(x_i)]^2}, \qquad i = \overline{0, m} \\ B_j &= \frac{[(n+1)!]^4(b-a)^{2n+3}}{[(2n+2)!]^2(y_j-c)(d-y_j)[u'(y_j)]^2}, \qquad j = \overline{0, n}, \end{aligned}$$

iar dacă $f \in C^{2m+2,2n+2}(D)$

$$\begin{aligned} R_{mn}(f) = (d-c)\lambda_m f^{(2m+2,0)}(\xi_1, \eta_1) + (b-a)\lambda_n f^{(0,2n+2)}(\xi_2, \eta_2) \\ - \lambda_m \lambda_n f^{(2m+2,2n+2)}(\xi_3, \eta_3), \end{aligned}$$

unde

$$\lambda_m = \frac{[(m+1)!]^4(b-a)^{2m+3}}{[(2m+2)!]^3(2m+3)}, \qquad \lambda_n = \frac{[(n+1)!]^4(b-a)^{2n+3}}{[(2n+2)!]^3(2n+3)}.$$

Pentru detalii a se vedea [10].

În cazul $m = n = 0$ se obţine o formulă de cubatură cu un singur nod, analoagă formulei dreptunghiului

$$\int_a^b \int_c^d f(x, y)\,\mathrm{d}x\,\mathrm{d}y = (b-a)(d-c) f\left(\frac{a+b}{2}, \frac{c+d}{2}\right) + R_{00}(f),$$

unde

$$\begin{aligned} R_{00}(f) = \frac{(b-a)^3(d-c)}{24} f^{(2,0)}(\xi_1, \eta_1) + \frac{(b-a)(d-c)^3}{24} f^{(0,2)}(\xi_2, \eta_2) \\ - \frac{(b-a)^3(d-c)^3}{576} f^{(2,2)}(\xi_3, \eta_3). \end{aligned}$$

Utilizarea metodelor de aproximare de mai sus nu este limitată la domenii rectangulare. De exemplu, tehnica de la formula de cubatură a lui Simpson se poate modifica pentru a fi aplicabilă la aproximarea unor unor integrale de forma

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\,\mathrm{d}x\,\mathrm{d}y$$

sau

$$\int_c^d \int_{a(y)}^{b(y)} f(x, y)\,\mathrm{d}x\,\mathrm{d}y.$$

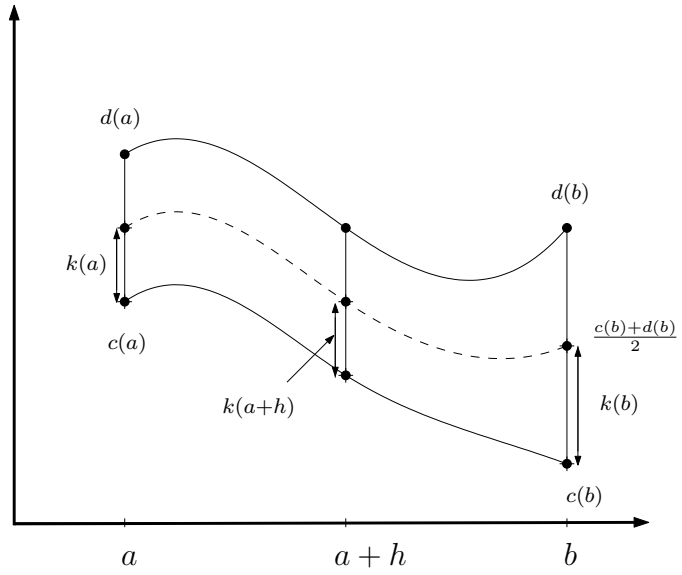



Figura 9.5: Formula lui Simpson pentru un domeniu simplu în raport cu $x$

Pentru o integrală de primul tip pasul după $x$ va fi $h = \frac{b-a}{2}$, dar cel după $y$ va varia odată cu $x$ (vezi figura 9.5)

$$k(x) = \frac{d(x) + c(x)}{2}.$$

Se obţine

$$\begin{aligned}
&\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,\mathrm{d}x\,\mathrm{d}y \approx \\
&\approx \int_a^b \frac{k(x)}{3}\left[f(x,c(x)) + 4f(x,c(x)+k(x)) + f(x,d(x))\right]\,\mathrm{d}x \\
&\approx \frac{h}{3}\Bigg\{\frac{k(a)}{3}\big[f(a,c(a)) + 4f(a,c(a)+k(a)) + f(a,d(a))\big] \\
&+ 4\frac{k(a+h)}{3}\big[f(a+h,c(a+h)) + 4f(a+h,c(a+h)+k(a+h)) \\
&+ f(a+h,d(a+h))\big] \\
&+ \frac{k(b)}{3}\left[f(b,c(b)) + 4f(b,c(b)+k(b)) + f(b,d(b))\right]\Bigg\}.
\end{aligned}$$

Dacă domeniul de integrare $D$ este curbiliniu, construim un dreptunghi $R \supset D$, ale cărui laturi să fie paralele cu axele de coordonate (figura 9.6). Se consideră funcţia auxiliară

$$f^*(x,y) = \begin{cases} f(x,y), & \text{dacă } (x,y) \in D; \\ 0, & \text{dacă } (x,y) \in R \backslash D;. \end{cases}$$

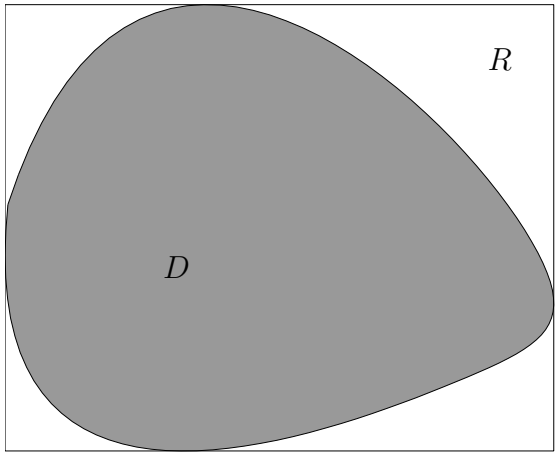



Figura 9.6: Încadrarea unui domeniu curbiliniu într-un dreptunghi

Evident

$$\iint_D f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \iint_R f^*(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

Ultima integrală se poate aproxima printr-o tehnică cunoscută.

### 9.2.1. Consideraţii de implementare

Fie domeniul de integrare

$$D = [a,b] \times [c,d].$$

Integrala de aproximat se poate scrie sub forma

$$\int_a^b \int_c^d f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int_a^b \left( \int_c^d f(x,y)\,\mathrm{d}y \right)\,\mathrm{d}x = \int_a^b F(x)\,\mathrm{d}x,$$

unde

$$F(x) = \int_c^d f(x,y)\,\mathrm{d}y.$$

Să presupunem că $adquad$ este o o rutină de cuadratură unidimensională adaptivă de forma

$$\textbf{function}\ adquad(f : functie, a, b : real,\ \varepsilon : real) : real$$

Ideea este de a folosi această această rutină pentru a calcula valorile lui $F$ definite mai sus şi de a folosi din nou rutina pentru a integra $F$. Descrierea este dată în algoritmul 9.1.

**Algoritmul 9.1** Aproximarea unei integrale duble pe dreptunghi

**Intrare:** funcţia $f$, intervalele $[a, b]$ şi $[c, d]$, rutina $adquad$, eroarea $\varepsilon$
**Ieşire:** Valoarea aproximativă a integralei

**function** $dblquad(f, a, b, c, d, adquad, \varepsilon) : real$
$Q := adquad(integint, a, b, \varepsilon, f, c, d, quadf)$;
**function** $integint(x, f, c, d, \varepsilon, adquad) : real$
{integrala interioară}
**for each** x **do**
$F(x) := adquad(f(x, \cdot), c, d, \varepsilon)$;
**end for**

# Bibliografie


[1] O. Agratini, *Positive Approximation Processes*, Hiperboreea Press, Turda, 2001.

[2] Octavian Agratini, Ioana Chiorean, Gheorghe Coman, Radu Trîmbiţaş, *Analiză numerică şi teoria aproximării*, vol. III, Presa Universitară Clujeană, Cluj-Napoca, 2002, coordonatori D. D. Stancu şi Gh. Coman.

[3] 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`.

[4] Å. Björk, *Numerical Methods for Least Squares Problem*, SIAM, Philadelphia, 1996.

[5] E. Blum, *Numerical Computing: Theory and Practice*, Addison-Wesley, 1972.

[6] P. Bogacki, L. F. Shampine, *A 3(2) pair of Runge-Kutta formulas*, Appl. Math. Lett. **2** (1989), no. 4, 321–325.

[7] C. G. Broyden, *A Class of Methods for Solving Nonlinear Simultaneous Equations*, Math. Comp. **19** (1965), 577–593.

[8] L. Burden, J. D. Faires, *Numerical Analysis*, PWS Kent, Boston, 1986.

[9] P. G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, Paris, Milan, Barcelone, Mexico, 1990.

[10] Gheorghe Coman, *Analiză numerică*, Editura Libris, Cluj-Napoca, 1995.

[11] I. Cuculescu, *Analiză numerică*, Editura Tehnică, Bucureşti, 1967.

[12] P. J. Davis, P. Rabinowitz, *Numerical Integration*, Blaisdell, Waltham, Massachusetts, 1967.

[13] James Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[14] J. E. Dennis, J. J. Moré, *Quasi-Newton Metods, Motivation and Theory*, SIAM Review **19** (1977), 46–89.

[15] J. Dormand, *Numerical Methods for Differential Equations. A Computational Approach*, CRC Press, Boca Raton New York, 1996.

[16] E. Fehlberg, *Klassische Runge-Kutta-Formeln fünfter und siebenter Ordnung mit Schrittweiten-Kontrolle*, Computing **4** (1969), 93–106, Corrigendum: *ibid.* 5, 184.

[17] E. Fehlberg, *Klassische Runge-Kutta-Formeln vierter und niedriger Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme*, Computing **6** (1970), 61–71, Corrigendum: *ibid.* 5, 184.

[18] J. G. F. Francis, *The QR transformation: A unitary analogue to the LR transformation*, Computer J. **4** (1961), 256–272, 332–345, parts I and II.

[19] W. Gander, W. Gautschi, *Adaptive quadrature - revisited*, BIT **40** (2000), 84–101.

[20] W. Gautschi, *Numerical Analysis, an Introduction*, Birkhäuser, Basel, 1997.

[21] Walther Gautschi, *Orthogonal polynomials: applications and computation*, Acta Numerica **5** (1996), 45–119.

[22] D. Goldberg, *What every computer scientist should know about floating-point arithmetic*, Computing Surveys **23** (1991), no. 1, 5–48.

[23] H. H. Goldstine, J. von Neumann, *Numerical inverting of matrices of high order*, Amer. Math. Soc. Bull. **53** (1947), 1021–1099.

[24] Gene H. Golub, Charles van Loan, *Matrix Computations*, 3rd ed., John Hopkins University Press, Baltimore and London, 1996.

[25] Nicholas J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[26] E. Isaacson, H. B. Keller, *Analysis of numerical methods*, John Wiley, New York, 1966.

[27] V. N. Kublanovskaya, *On some algorithms for the solution of the complete eigenvalue problem*, USSR Comp. Math. Phys. **3** (1961), 637–657.

[28] J. J. Moré, M. Y. Cosnard, *Numerical Solutions of Nonlinear Equations*, ACM Trans. Math. Softw. **5** (1979), 64–85.

[29] 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/`.

[30] I. A. Rus, *Ecuaţii diferenţiale, ecuaţii integrale şi sisteme dinamice*, Transilvania Press, Cluj-Napoca, 1996.

[31] I. A. Rus, P. Pavel, *Ecuaţii diferenţiale*, Editura Didactică şi Pedagogică, Bucureşti, 1982, ediţia a doua.

[32] H. Rutishauser, *Solution of the eigenvalue problems with the LR transformation*, Nat. Bur. Stand. App. Math. Ser. **49** (1958), 47–81.

[33] 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`.

[34] A. Sard, *Linear Approximation*, American Mathematical Society, Providence, RI, 1963.

[35] Thomas Sauer, *Numerische Mathematik II*, Universität Erlangen-Nurnberg, Erlangen, 2000, Vorlesungskript.

[36] R. Schwarz, H., *Numerische Mathematik*, B. G. Teubner, Stuttgart, 1988.

[37] D. D. Stancu, *Analiză numerică – Curs şi culegere de probleme*, Lito UBB, Cluj-Napoca, 1977.

[38] D. D. Stancu, G. Coman, P. Blaga, *Analiză numerică şi teoria aproximării*, vol. II, Presa Universitară Clujeană, Cluj-Napoca, 2002, D. D. Stancu, Gh. Coman, (coord.).

[39] D. D. Stancu, Gh. Coman, O. Agratini, R. Trîmbiţaş, *Analiză numerică şi Teoria aproximării*, vol. I, Presa Universitară Clujeană, Cluj-Napoca, 2001.

[40] J. Stoer, R. Burlisch, *Einfuhrung in die Numerische Mathematik*, vol. II, Springer Verlag, Berlin, Heidelberg, 1978.

[41] J. Stoer, R. Burlisch, *Introduction to Numerical Analysis*, 2nd ed., Springer Verlag, 1992.

[42] A. H. Stroud, *Approximate Calculation of Multiple Integrals*, Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

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

[44] C. Überhuber, *Computer-Numerik*, vol. 1, 2, Springer Verlag, Berlin, Heidelberg, New-York, 1995.

[45] C. Ueberhuber, *Numerical Computation. Methods, Software and Analysis*, vol. I, II, Springer Verlag, Berlin, Heidelberg, New York, 1997.

[46] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

# Indice