Calcul Numeric

Cursul 1

2022

Anca Ignat

Echipa

Andreea Arusoaie **Valentin Roşca**

Ştefan Bălăucă **Augustus Tabarcea (en)**

Sebastian Ciobanu **Anca Ignat**

Andrei Luca

ancai@info.uaic.ro , ancai_fii@yahoo.ro

numericalcalculus2019@gmail.com – teme de laborator

http://profs.info.uaic.ro/~ancai/CN/

Detalii despre modul de desfăşurare online: Discord

Consultaţii:

- **prin e-mail la adresele de mai sus sau**
- **Online, Zoom – Luni 16:30-18**

Regulament - 2022

Laborator

- 8 teme
- cei care prezintă temele până la termenul limită precizat la fiecare temă, punctajul maxim ce poate fi obţinut este punctajul afişat pentru fiecare temă
- cei care prezintă temele după termenul limită precizat punctarea se va face din 50% din punctajul temei prezentate

Examen

- teză scrisă de 1 oră, cu 3 sau 4 exerciţii din materia predată, *"cu cursurile pe masă"*
- teza scrisă este notată între 1 şi 10
- testul scris va avea loc:

  - în săptămâna 10 sau 11 - din primele 9 cursuri
  - 16 mai - 22 mai 2022 - pentru cei care nu obţin punctaj de promovare sau pentru mărirea notei - din toate cursurile

Calculul punctajului / notei final(e)

Punctaj final = punctaj laborator + 45*nota examen

Promovează disciplina acei studenţi care au:

- **nota la examen ≥ 3**

şi

- **punctaj final ≥ 410 pt**

Nota finală se calculează din punctajul final aplicând **”curba lui Gauss”**.

Desfăşurarea semestrului

Săptămânânile 1-7 şi 9-13 – şcoală conform orarului

Săptămâna a 8-a (prima săptămână de evaluare) – liberă

Săptămânile 10, 11, 12 - test scris

Bibliografie

1. Elemente de informatică şi calcul numeric, vol. 1 - C.Ignat, C.Ilioi, T.Jucan - Ed. Univ. 'Al. I. Cuza' Iaşi, 1987
2. Matrix Computations - G.H. Golub, C.F. van Loan - John Hopkins Univ. Press, 2012
3. Numerical Analysis – R.L. Burden, J.D. Faires – Brooks/Cole, Thomson Learning (10-th edition, 2015)
4. Calcul numeric în C - T. A. Beu - Ed. Albastră, Cluj, 2004
5. Numerical analysis with algorithms and programming, Santanu Saha Ray, CRC Press, 2016.
6. Numerical Recipes 3rd Edition: The Art of Scientific Computing, W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Cambridge University Press, NY, USA, 2007 (http://numerical.recipes/)
7. Linear Algebra, Ideas and Applications - R.C. Penney, 4-th ed., Wiley, 2016
8. Numerical Optimization – J. Nocedal, S.J. Wright, Springer-Verlag, New York, 1999

Capitolele cursului

- Rezolvarea sistemelor liniare ($\boldsymbol{Ax=b}$)
- Optimizare numerică ($\mathbf{min}\ \{\ \boldsymbol{F(x)\ ;\ x \in R^n}\ \}$)
- Valori şi vectori proprii ($\boldsymbol{Au = \lambda u}$)
- Interpolare numerică
- Ecuaţii neliniare ($\boldsymbol{f(x)=0}$)

“***The world cannot be understood without numbers. But the world cannot be understood with numbers alone.***”

Hans Rosling, Anna Rosling Ronnlund, Ola Rosling

Factfulness. Zece motive pentru care interpretăm greşit lumea şi de ce lucrurile stau mai bine decât crezi

“The world cannot be understood without numbers, nor through numbers alone.”

Traffic Flow

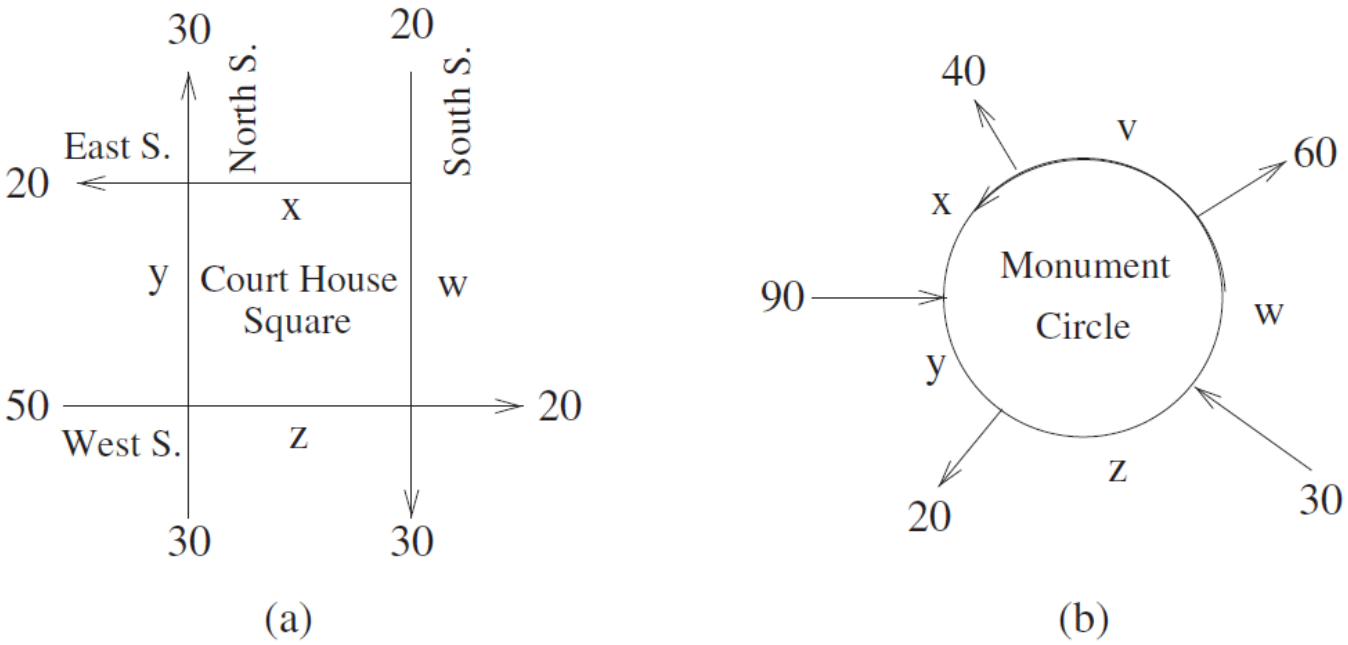
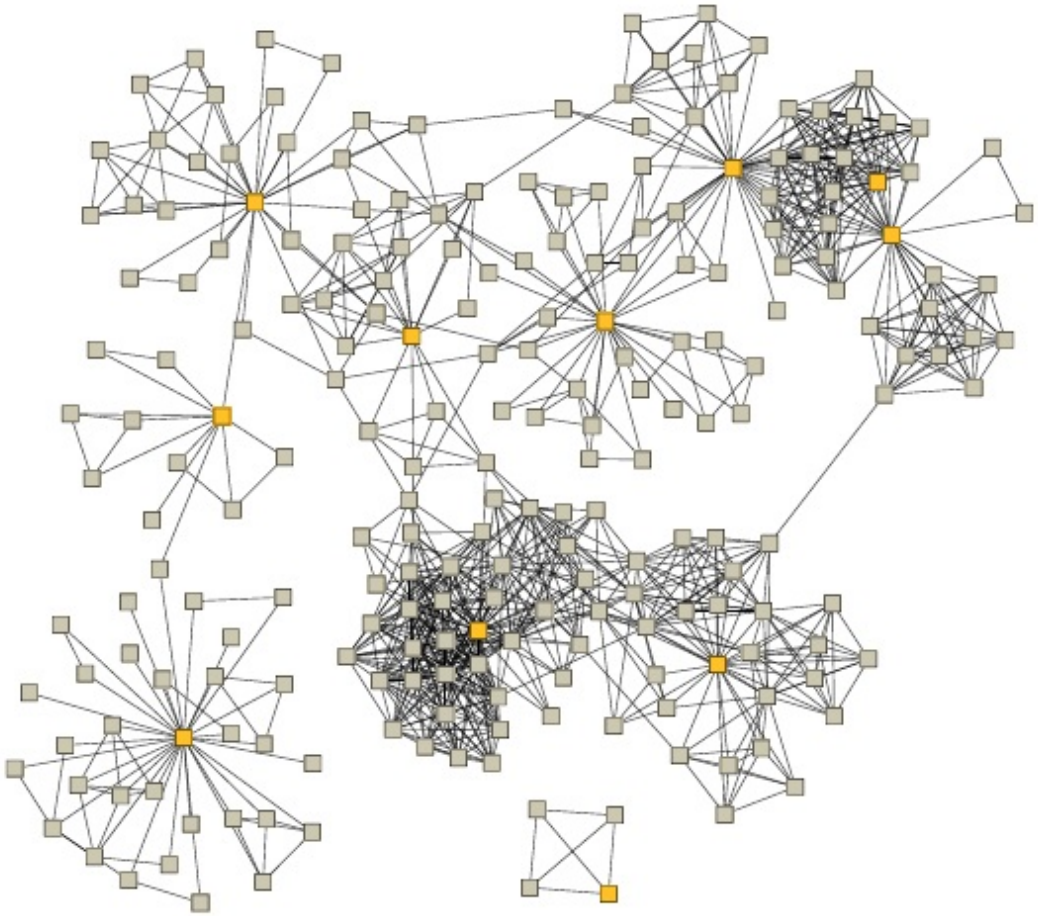


FIGURE 1.24 Two traffic patterns.

(R.C. Penney – Linear Algebra, Ideas and Applications, 4-th ed., Wiley, 2016)

- one way streets;
- the numbers represent the average number of cars per minute that enter or leave a given street at 3:30pm;
- ***x, y, z, w***, … - average number of cars per minute on a certain street
- no. of cars entering = no. of cars leaving

$$
\begin{aligned}
x+y &= 50 \\
y+z &= 80 \\
z+w &= 50 \\
x+w &= 20 \\
x+y+z+w &= 100
\end{aligned}
$$



Centralitate în reţelele sociale

- noţiune introdusă de A. Bavelas în 1948 studiind comunicarea între oameni

(V, E) – graful care modelează reţeaua, ***A*** – matricea de adiacenţă asociată, $\boldsymbol{A = (a_{ij})_{i,j=1}^{N}}$, ***N*=|*?*|**

- care sunt cele mai ‘importante’ noduri din reţea?

1. Centralitate de grad (‘*degree centrality*’) – se bazează pe noţiunea de grad/grade asociate nodurilor în grafuri

   Se numeşte **drum geodesic** între două vârfuri orice drum de lungime minimă (număr minim de muchii) dintre cele 2 vârfuri.

2. Centralitate de apropiere (‘*closeness centrality*’)

   Centralitatea de apropiere a unui nod este suma lungimilor drumurilor geodesice de la nodul respectiv la toate celelalte noduri.

3. Centralitate de interrelaţie (‘*betweennes centrality*’)

$$b(v) = \sum_{\substack{s \neq v \neq t, \\ s,t \in V}} \frac{n_{st}(v)}{n_{st}}$$

unde n_{st} este numărul total de drumuri geodesice între nodurile s şi t, iar $n_{st}(v)$ este numărul de drumuri geodesice care trec prin nodul v.

- măsoară controlul pe care îl deţine nodul v în circulaţia informaţiilor în reţea

4. Centralitate de vector propriu (‘*eigenvector centrality*’)

- se ţine cont de faptul că nu toate muchiile (conexiunile) sunt la fel de importante (ca în cazul centralităţii de grad)

- conexiunile către persoane influente vor ‘împrumuta’ importanţă mai mare decât conexiunie către persoanele mai puţin influente

x(i) = centralitatea de vector propriu a nodului $\boldsymbol{v_i}$

$$x(i) = \frac{1}{\lambda} \sum_{j \in \Gamma(v_i)} x(j) = \frac{1}{\lambda} \sum_{j=1}^{N} a_{ij} x(j) \ , \ \lambda > 0 \text{ o constanta}$$

$$x = (x(1), x(2), \ldots, x(N))^T$$

$$x = \frac{1}{\lambda} Ax \ \Leftrightarrow Ax = \lambda x$$

$\lambda_A > 0$ - valoarea proprie Perron (cea mai mare valoare proprie) , x – vectorul propriu asociat

Compresia imaginilor digitale şi descompunerea după valori singulare

o imagine digitală ↔ matrice de pixeli $\boldsymbol{A}$ cu $\boldsymbol{m}$ linii şi $\boldsymbol{n}$ coloane

$$\boldsymbol{A}=\left(\boldsymbol{a}_{ij}\right)_{\substack{i=1,\ldots,m\\ j=1,\ldots,n}},\ \boldsymbol{a}_{ij}\in\mathbb{R} \text{ sau } \mathbb{R}^3$$

$$(\boldsymbol{a}_{ij}\in\{0,1,\ldots,255\} \text{ sau } \boldsymbol{a}_{ij}\in\{0,1,\ldots,255\}^3 \text{ sau } \boldsymbol{a}_{ij}\in[0,1]^{(3)})$$

Memorarea lui $\boldsymbol{A}$: $\boldsymbol{m\cdot n\cdot}$**mem***(int/double)(·3)* bytes

Descompunerea dupa valori singulare (**SVD**) a unei matrici

$$\boldsymbol{A}=\boldsymbol{U}\,\boldsymbol{S}\boldsymbol{V}^T,\ \boldsymbol{U}\in\mathbb{R}^{m\times m},\ \boldsymbol{S}\in\mathbb{R}^{m\times n},\boldsymbol{V}\in\mathbb{R}^{n\times n}$$

$\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_m],\ \boldsymbol{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_n]$ – matrici ortogonale

$$\left(u_i, u_j\right)_{\mathbb{R}^m} = \delta_{ij} = \begin{cases} 1 \text{ daca } i = j \\ 0 \text{ daca } i \neq j \end{cases}, \quad \left(v_i, v_j\right)_{\mathbb{R}^n} = \delta_{ij} \quad \forall i, j$$

$$S = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_2 & & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & & \sigma_r & 0 \\ & & & & \ddots \\ 0 & 0 & & 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times n}$$

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, $r \leq \min\{m, n\}$ - valorile singulare ale matr. A

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$$

$$A \approx A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T$$

memorarea lui A_k necesită $k(m+n+1)\cdot$**mem(double)**

$m=1265, n=538$, $r_c = \dfrac{mn}{k(m+n+1)}$,

$k=50$ $r_c=7.54$; $k=100$ $r_c=3.77$; $k=200$ $r_c=1.88$;

$k=300$ $r_c=1.25$; $k=400$ $r_c=0.94$; $k=538$ $r_c= 0.70$

Vectori şi matrici

Fie $x_i, y_i, \lambda \in \mathbb{R}$. Se definesc vectorii $x, y \in \mathbb{R}^n$ şi operaţiile de adunare şi înmulţire cu scalari astfel:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad x + y = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{pmatrix}, \lambda x = \begin{pmatrix} \lambda x_1 \\ \lambda x_2 \\ \vdots \\ \lambda x_n \end{pmatrix} .$$

Fie vectorul $z \in \mathbb{C}^n$:

$$z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} \text{ cu } \quad z_1, z_2, ..., z_n \in \mathbb{C}.$$

Pentru $z \in \mathbb{C}$ utilizăm notaţiile:

$$z = a + ib \text{ , } Re\, z = a, \quad Im\, z = b \text{ ,}$$

$\overline{z} = a - ib$ – conjugatul numărului z

$|z| = \sqrt{a^2 + b^2}$ - modulul numărului complex z

Notăm cu $\mathbb{R}^{m\times n}$ / $\mathbb{C}^{m\times n}$ spaţiul matricilor cu elemente reale / complexe cu m linii şi n coloane

$$A=\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix},\ a_{ij}\in\mathbb{R}\ /\ \mathbb{C},\ i=1,2,...,m\ ,\ j=1,2,...,n$$

Definiţie

X se numeşte *spaţiu vectorial (spaţiu liniar)*

$$+ : X \times X \to X \text{ şi } \cdot : K \times X \to X, \qquad (K = \mathbb{R})$$

astfel încât $(X , +)$ este un grup comutativ :

$a + b = b + a \ , \ \forall a,b \in X$ – comutativitate,

$(a + b) + c = a + (b + c) \ , \ \forall a,b,c \in X$ – asociativitate ,

$\exists 0 \in X$ **a. î.** $a + 0 = 0 + a = a \ , \ \forall a \in X$ - element neutru ,

$\forall a \in X, \ \exists -a \in X$ a.î. $a + (-a) = (-a) + a = 0$ – element opus.

iar pentru operaţia de înmulţire cu scalari au loc relaţiile:

$\lambda(a+b) = \lambda a + \lambda b, \forall \lambda \in K, \quad \forall a,b \in X,$

$(\lambda + \mu) a = \lambda a + \mu a, \ \forall \lambda,\mu \in K, \quad \forall a \in X,$

$\lambda(\mu a) = (\lambda \mu) a, \ \forall \lambda,\mu \in K, \quad \forall a \in X,$

$\exists 1 \in K$ astfel încât $1 \cdot a = a, \ \forall a \in X.$

Definiţie

Fie X un spaţiu liniar. Spunem că vectorii $x_1, x_2, \ldots, x_p \in X$ sunt *liniar independenţi* dacă:

$$\alpha_1 x_1 + \alpha_2 x_2 + .. + \alpha_p x_p = 0 \Rightarrow \alpha_1 = \alpha_2 = ...\alpha_p = 0, \alpha_i \in K$$

*Spaţiu*l vectorial X este *finit dimensional* dacă există p vectori liniar independenţi în X, $x_1, x_2, \ldots, x_p \in X$, şi orice mulţime de q elemente din X cu $q > p$ este liniar dependentă. În acest caz dimensiunea spaţiului X este p (dim $X = p$).

Fie spaţiul vectorial X finit dimensional cu dim $X = p$. Orice sistem de p vectori liniar independenţi din X se numeşte *bază* a spaţiului X.

Fie $x_1 , x_2 , \ldots, x_p \in X$ o bază pentru spaţiul X . Atunci pentru $\forall x \in X$, $\exists$ unice constantele $\alpha_1, \alpha_2, \ldots, \alpha_p \in K$ astfel încât

$$x = \alpha_1 x_1 + \alpha_2 x_2 + .. + \alpha_p x_p = \sum_{i=1}^{p} \alpha_i \, x_i \, .$$

$\mathbb{R}^n$ este un spaţiu vectorial finit dimensional, dim $\mathbb{R}^n = \boldsymbol{n}$ cu baza canonică:

$$\boldsymbol{e}_1 = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \boldsymbol{e}_2 = \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ \vdots \\ \mathbf{0} \end{pmatrix}, \ldots, \boldsymbol{e}_k = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{1} \\ \vdots \\ \mathbf{0} \end{pmatrix} \text{- poziţia } \boldsymbol{k}, \ldots, \boldsymbol{e}_n = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{1} \end{pmatrix}$$

Calcul matricial

Fie matricea $A \in \mathbb{R}^{m \times n}$:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \quad A = \left(a_{ij}\right)_{i=1\ldots m, j=1\ldots n}$$

Se defineşte ***matricea transpusă***:

$$A^T = \begin{pmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{mn} \end{pmatrix}, A^T = \left(a_{ji}\right)_{i=1\ldots m, j=1\ldots n} \in \mathbb{R}^{n \times m}$$

Pentru matricea:

$$A \in \mathbb{C}^{m \times n},\ A = \left(a_{ij}\right)_{\substack{i=1\ldots m \\ j=1\ldots n}}$$

se defineşte ***matricea adjunctă*** A^H:

$$A^H = \overline{A^T} = \left(\overline{a_{ji}}\right)_{\substack{j=1\ldots n \\ i=1\ldots m}}$$

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \qquad A^H = \begin{pmatrix} \overline{a_{11}} & \cdots & \overline{a_{m1}} \\ \vdots & \ddots & \vdots \\ \overline{a_{1n}} & \cdots & \overline{a_{mn}} \end{pmatrix}$$

Pentru $A \in \mathbb{R}^{m \times n}$ matricea adjunctă coincide cu transpusa, $A^H = A^T$.

Fie vectorul $x \in \mathbb{R}^n$, acesta este considerat vector coloană, $x \in \mathbb{R}^{n\times 1}$:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \Rightarrow \quad x^T = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}$$

O primăvară frumoasă!

Calcul Numeric

Cursul 2

2022

Anca Ignat

Dacă fa**c**em înmulţirea matricială $\boldsymbol{Ae_j}$ obţinem coloana $\boldsymbol{j}$ a matricei $\boldsymbol{A}$:

$$\boldsymbol{Ae_j} = \begin{pmatrix} \boldsymbol{a_{11}} & \cdots & \boldsymbol{a_{1n}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{a_{m1}} & \cdots & \boldsymbol{a_{mn}} \end{pmatrix} \begin{pmatrix} \boldsymbol{0} \\ \vdots \\ \boldsymbol{1}_{\text{poziţia } j} \\ \vdots \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{a_{1j}} \\ \boldsymbol{a_{2j}} \\ \vdots \\ \boldsymbol{a_{mj}} \end{pmatrix}$$

$\boldsymbol{Ae_j}$ este coloana $\boldsymbol{j}$ a matricei $\boldsymbol{A}$, $\boldsymbol{j=1,...,n}$;

$\boldsymbol{e_i^T A}$ este linia $\boldsymbol{i}$ a matricei $\boldsymbol{A}$, $\boldsymbol{i=1,...,m}$.

Fie vectorii $\boldsymbol{x}, \boldsymbol{y}$, cu ajutorul lor definim produsele scalare în $\mathbb{C}^n$ şi $\mathbb{R}^n$:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{C}^n \,, \; y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{C}^n$$

$$(x, y) = \sum_{i=1}^{n} x_i \overline{y_i} = y^H x = \left(\overline{y_1} \;\; \overline{y_2} \;\; \cdots \overline{y_n} \right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n \, , y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n$$

$$(x, y) = \sum_{i=1}^{n} x_i y_i = y^T x = (y_1 \; y_2 \cdots y_n) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Proprietăţile matricei $\boldsymbol{A^H}$

1. $\boldsymbol{(A + B)^H = A^H + B^H}$
2. $\boldsymbol{(A^H)^H = A}$
3. $\boldsymbol{(AB)^H = B^H\, A^H}$
4. $\boldsymbol{(A^{-1})^H = (A^H)^{-1}}$

Proprietăţi ale matricei $\boldsymbol{A^T}$

$\boldsymbol{(A + B)^T = A^T + B^T}$

$\boldsymbol{(A^T)^T = A}$

$\boldsymbol{(AB)^T = B^T\, A^T}$

$\boldsymbol{(A^{-1})^T = (A^T)^{-1}}$

Propoziţie

Fie $A \in \mathbb{C}^{m \times n}$, $x \in \mathbb{C}^n$, $y \in \mathbb{C}^m$ atunci:

$$\left(Ax, y\right)_{\mathbb{C}^m} = \left(x , A^H y\right)_{\mathbb{C}^n} .$$

Pentru cazul real avem:

$$A \in \mathbb{R}^{m \times n}, \; x \in \mathbb{R}^n, y \in \mathbb{R}^m \Rightarrow \left(Ax, y\right)_{\mathbb{R}^m} = \left(x , A^T y\right)_{\mathbb{R}^n}$$

Demonstraţie

$$(Ax , y) = y^H (Ax) = y^H A x = y^H (A^H)^H x =$$

$$= (A^H y)^H x = (x , A^H y).$$

Tipuri de matrice

Definiţii

O matrice $\boldsymbol{A \in \mathbb{R}^{n\times n}}$ se numeşte ***simetrică*** dacă $\boldsymbol{A = A^T}$.

O matrice $\boldsymbol{A \in \mathbb{C}^{n\times n}}$ se numeşte ***autoadjunctă*** dacă $\boldsymbol{A = A^H}$.

O matrice $\boldsymbol{A \in \mathbb{C}^{n\times n}}$ se numeşte ***unitară*** dacă $\boldsymbol{A^H A = A\, A^H = I_n}$.

O matrice $\boldsymbol{A \in \mathbb{C}^{n\times n}}$ se numeşte ***ortogonală*** dacă

$$\boldsymbol{A^T A = A\, A^T = I_n}.$$

O matrice $A \in \mathbb{C}^{n\times n}$, $A=(a_{ij})$ se numește matrice ***triunghiulară inferior*** (sau ***inferior triunghiulară***) dacă

$$a_{ij} = 0 \text{ pentru } j > i$$

$$A = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 & 0 \\ \vdots & & & & & \\ a_{(n-1)1} & a_{(n-1)2} & a_{(n-1)3} & \cdots & a_{(n-1)(n-1)} & 0 \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{n(n-1)} & a_{nn} \end{pmatrix}$$

O matrice $A \in \mathbb{C}^{n\times n}$, $A=(a_{ij})$ se numește matrice ***triunghiulară superior*** (sau ***superior triunghiulară***) dacă

$$a_{ij} = 0 \text{ pentru } j < i$$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1(n-1)} & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2(n-1)} & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3(n-1)} & a_{3n} \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & a_{(n-1)(n-1)} & a_{(n-1)n} \\ 0 & 0 & 0 & \cdots & 0 & a_{nn} \end{pmatrix}$$

Notăm cu $\boldsymbol{I_n}$ matricea unitate:

$$I_n \in \mathbb{R}^{n \times n}, I_n = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Matrice diagonală $\boldsymbol{D}=\mathbf{diag}[d_1, d_2,\ldots,d_n]$

$$D \in \mathbb{R}^{n\times n}\ ,\ D = \begin{pmatrix} d_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & d_2 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & d_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & d_n \end{pmatrix}$$

Norme

Definiție

Fie X un spațiu vectorial real. Se numește ***normă*** aplicația:

$$\|\cdot\| : X \to \mathbb{R}_+$$

care îndeplinește condițiile:

(1) $\|x\| \geq 0; \quad \|x\| = 0 \Leftrightarrow x = 0;$

(2) $\|x + y\| \leq \|x\| + \|y\|, \forall\, x, y \in X;$

(3) $\|\lambda x\| = |\lambda| \|x\|, \forall\, x \in X, \forall \lambda \in \mathbb{R}.$

Vom numi ***norme vectoriale*** normele definite pe spațiile $X = \mathbb{C}^n$ sau $\mathbb{R}^n$.

Exemple

Fie spațiile vectoriale $\mathbb{C}^n$ sau $\mathbb{R}^n$. Pe aceste spații următoarele aplicații sunt norme vectoriale:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| - \text{ city-block (Manhattan)}$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2} - \text{ euclidiană}$$

$$\|x\|_\infty = \max\{|x_i|, i = 1..n\} - \text{Cebyshev (a tablei de şah)}$$

Dacă $\|\cdot\|_v$ este o normă vectorială şi $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ este o matrice nesingulară ($\det \boldsymbol{P} \neq 0$) atunci aplicația:

$$\|\cdot\|_P : \mathbb{R}^n \to \mathbb{R}, \qquad \|x\|_P = \|Px\|_v$$

este de asemenea o normă vectorială.

Definiție

Se numește ***produs scalar*** în spațiul vectorial X aplicația:

$$(\cdot,\cdot): X \times X \to K$$

care satisface condițiile :

$$(a)\ (x,x) \geq 0, \forall\, x \in X, \quad (x,x) = 0 \ \Leftrightarrow \ x = 0;$$

$$(b)\ (x,y) = \overline{(y,x)}, \forall\, x, y \in X,$$

$$(c)\ (\lambda x, y) = \lambda (x, y), \forall\, x, y \in X, \forall \lambda \in K,$$

$$(d)\ (x+y, z) = (x,z) + (y,z), \forall\, x, y, z \in X.$$

Inegalitatea lui Cauchy-Buniakovski-Schwarz:

$$|(x, y)| \leq \sqrt{(x, x)}\sqrt{(y, y)} \quad \forall x, y \in X$$

Într-un spațiu vectorial dotat cu produs scalar se poate induce o normă numită euclidiană:

$$\|x\|_2 = |x| := \sqrt{(x, x)}.$$

Reamintim definiția produselor scalare pe $\mathbb{C}^n$ şi pe $\mathbb{R}^n$ introduse anterior:

$$(x, y)_{\mathbb{C}^n} = \sum_{i=1}^{n} x_i \overline{y_i} \quad , \quad (x, y)_{\mathbb{R}^n} = \sum_{i=1}^{n} x_i y_i$$

Obținem norma euclidiană (valabilă în spaţiile $\mathbb{C}^n$ şi $\mathbb{R}^n$):

$$\|x\|_2 = |x| = \sqrt{\sum_{i=1}^{n} |x_i|^2}.$$

Norme matriceale

Definiție

Aplicația $\|\cdot\|:\mathbb{R}^{n\times n}\to\mathbb{R}$ se numește ***normă matriceală*** dacă:

(1) $\|A\|\geq 0\ \forall\ A\in\mathbb{R}^{n\times n}\ ;\ \|A\|=0\ \Leftrightarrow\ A=0.$

(2) $\|\alpha A\|=|\alpha|\|A\|\ ,\forall\alpha\in\mathbb{R},\forall\ A\in\mathbb{R}^{n\times n}.$

(3) $\|A+B\|\leq\|A\|+\|B\|,\forall\ A,B\in\mathbb{R}^{n\times n}.$

(4) $\|A*B\|\leq\|A\|\cdot\|B\|,\forall\ A,B\in\mathbb{R}^{n\times n}.$

Exemple

Norma Frobenius definită de relația $\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}|^2}$ este o normă matriceală.

Aplicația $\|A\|_{\max} = \max\{|a_{ij}|; i = 1,\ldots,n,\ j = 1,\ldots,n\}$ $\underline{NU}$ este o normă matriceală.

Pentru $n = 2$ fie:

$$A = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}, B = A^T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$A * B = I_2 , \|A\|_{\max} = \|B\|_{\max} = \frac{1}{\sqrt{2}}$$

$$\|A * B\|_{\max} = 1 > \|A\|_{\max} \bullet \|B\|_{\max} = \frac{1}{2}.$$

Norme matriceale naturale

- $\|\cdot\|_v : \mathbb{R}^n \to \mathbb{R}_+$ o normă vectorială $\to \|\cdot\|_i : \mathbb{R}^{n\times n} \to \mathbb{R}_+$ ***normă matriceală naturală*** sau ***indusă***.

$$\|A\|_i = \max\{ \frac{\|Ax\|_v}{\|x\|_v} ; x \in \mathbb{R}^n , x \neq 0\}$$

Definiţii echivalente :

$$\|A\|_i = \max\{ \|Ax\|_v ; x \in \mathbb{R}^n , \|x\|_v \leq 1\}$$

$$= \max\{ \|Ax\|_v ; x \in \mathbb{R}^n , \|x\|_v = 1\}$$

$\|A\|_i$ se numeşte ***normă matriceală naturală*** sau ***normă indusă*** de norma vectorială $\|\cdot\|_v$

Avem următoarea relaţie:

$$\|Ax\|_v \le \|A\|_i \|x\|_v \,, \forall\, A \in \mathbb{R}^{n\times n} \,, \forall x \in \mathbb{R}^n .$$

Norma Frobenius $\|\cdot\|_F$ nu este o normă naturală.

$$\|I_n\|_i = \max\{ \frac{\|I_n x\|_v}{\|x\|_v} ;\, x \neq 0 \} = 1, \quad \forall\ \|\cdot\|_i ,$$

$$\|I_n\|_F = (1+1+\cdots+1)^{1/2} = \sqrt{n} \neq 1 \ \text{pentru} \quad n \ge 2.$$

Pentru $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ norma matriceală indusă este:

$$\|A\|_1 = \max\{\sum_{i=1}^{n} |a_{ij}| ; j = 1, 2, \ldots, n\}$$

Pentru $\|x\|_\infty = \max\{|x_i| ; i = 1, \ldots, n\}$ norma matricială indusă este:

$$\|A\|_\infty = \max\{\sum_{j=1}^{n} |a_{ij}| ; i = 1, 2, \ldots, n\}.$$

- $\|\cdot\|_v$ şi $\|\cdot\|_{v,P}$ - norme vectoriale $\rightarrow$ $\|\cdot\|_i$ şi respectiv $\|\cdot\|_{i,P}$

normele matriciale induse

$$\begin{array}{ccc} \|\cdot\|_v & \rightarrow & \|\cdot\|_i \\ \downarrow & & ? \\ \|\cdot\|_{v,P} & \rightarrow & \|\cdot\|_{i,P} \end{array}$$

$$\|x\|_{v,P} = \|Px\|_v \quad \rightarrow \quad \|A\|_{i,P} = \left\|PAP^{-1}\right\|_i$$

Valori şi vectori proprii

Definiţii

Fie $A \in \mathbb{R}^{n \times n}$. Se numeşte ***valoare proprie*** (***autovaloare***) a matricei A un număr complex $\lambda \in \mathbb{C}$ pentru care există un vector nenul $x \in \mathbb{C}^n$, $x \neq 0$ a.î.:

$$Ax = \lambda x.$$

Vectorul x se numeşte ***vector propriu*** (***autovector***) asociat val. proprii λ.

$$Ax = \lambda x \;\Leftrightarrow\; (\lambda I_n - A)x = 0, x \neq 0 \;\Leftrightarrow\; \det(\lambda I_n - A) = 0$$

→ Matricea $\lambda I_n - A$ este singulară.

Polinomul:

$$p_A(\lambda) = \det(\lambda I_n - A) = \lambda^n - a_1\lambda^{n-1} - a_2\lambda^{n-2} - \ldots - a_{n-1}\lambda - a_n$$

se numeşte ***polinom caracteristic*** asociat matricei A.

→ **grad** $p_A = n \rightarrow$ are n rădăcini care sunt valorile proprii ale matricei A.

Se numeşte ***rază spectrală*** a matricei A:

$$\rho(A) = \max\{|\lambda_i|, i = 1,\ldots,n, \lambda_i - \text{valorile proprii ale matricei A}\}$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$$ norma indusă este

$\|A\|_2 = |A| = \sqrt{\rho(A^T A)}$ se numeşte ***norma spectrală***.

Propoziţie

Fie $\|\cdot\|$ o normă matriceală naturală. Atunci:

$$\rho(A) \leq \|A\|, \forall\, A \in \mathbb{R}^{n \times n}.$$

Numere în format binar

În 1985 IEEE a publicat un raport numit Binary Floating Point Arithmetic Standard 754-1985 și o actualizare în 2008 IEEE 754-2008 care furnizează standarde pentru numere în virgulă mobilă binare și decimale, formate de interschimbare a tipului de date, algoritmi de rotunjire aritmetică, tratarea excepțiilor. Aceste standarde sunt respectate de toți fabricanții de calculatoare care folosesc arhitectura în virgulă mobilă.

O reprezentare binară pe 64 de biți a unui număr real se face în felul următor: primul bit este bitul de semn, următorii 11 biți reprezintă exponentul c iar următorii 52 de biți conțin informații despre partea fracționară, f, numită și mantisă:

$$(-1)^s \, 2^{c-1023} (1 + f) \; .$$

0 10000000011 1011100100010000000000000000000000000000000000000000 =
27.56640625

[27.566406249999982236431605997495353221893310546875,
27.566406250000001776356839400250464677810668945312 5).

Cel mai mic număr pozitiv care poate fi reprezentat este cu $s = 0, c = 1, f = 0$ adică

$$z = 2^{-1022}(1+0) \approx 0.22251 \times 10^{-307}$$

iar cel mai mare este pentru $s = 0,\ c = 2046,\ f = 1 - 2^{-52}$

$$Z = 2^{1023}(2 - 2^{-52}) \approx 0.17977 \times 10^{309}.$$

Numerele care apar în calcule și sunt mai mici decât z sunt setate în general la 0 (*underflow*) iar cele mai mari decât Z duc, de obicei, la oprirea calculelor (*overflow*).

Se observă că numărul 0 are două reprezentări: $s=0, c=1, f=0$ și $s=1, c=1, f=0$.

Reprezentarea zecimală

$\pm 0.d_1 d_2 \ldots d_k \times 10^n \quad 1 \le d_1 \le 9, \; 0 \le d_i \le 9, \; i = 2, \ldots, k$ -

reprezentarea zecimală folosind k cifre. Orice număr real y:

$$y = 0.d_1 d_2 \ldots d_k d_{k+1} d_{k+2} \ldots \times 10^n$$

poate fi reprezentat folosind k cifre printr-o simplă trunchiere

$$fl(y) = 0.d_1 d_2 \ldots d_k \times 10^n \quad .$$

O altă metodă de a obține o reprezentare cu k cifre este prin rotunjire:

$$fl(y) = 0.\delta_1\delta_2 \ldots \delta_k \times 10^n$$

Dacă $d_{k+1} \geq 5$ se adaugă 1 la d_k pentru a obține $fl(y)$ (*round up*), altfel se face trunchierea la k cifre (*round down*).

Un număr r^* aproximează numărul r cu t cifre exacte dacă t este cel mai mare întreg nenegativ pentru care:

$$\frac{|r - r^*|}{|r|} \leq 5 \times 10^{-t} .$$

În cazul trunchierii avem

$$\left|\frac{y - fl(y)}{y}\right| \leq 10^{-k+1}$$

iar când se face rotunjirea:

$$\left|\frac{y - fl(y)}{y}\right| \leq 0.5 \times 10^{-k+1}.$$

Operațiile elementare

$$x +_c y = fl(fl(x) + fl(y))$$
$$x -_c y = fl(fl(x) - fl(y))$$
$$x \times_c y = fl(fl(x) \times fl(y))$$
$$x \div_c y = fl(fl(x) \div fl(y))$$

Surse de erori în calculele numerice

1. Erori în datele de intrare:
   - măsurători afectate de erori sistematice sau perturbații temporare,
   - erori de rotunjire: 1/3 , π , 1/7,…

2. Erori de modelare

- erori de discretizare:

  limita unui şir , suma unei serii , funcţii neliniare aproximate de funcţii liniare, aproximarea derivatei unei funcţii …

- simplificări în modelul matematic

  idealizări , ignorarea unor parametri…

3. Erori în timpul calculelor:
   - de rotunjire datorate capacității limitate de memorare a datelor, operațiile nu sunt efectuate exact.
   - erori ale bibliotecilor folosite (*‚bug'*).
4. Erori umane (date, algoritmi, înțelegerea problemei)

Eroare absolută , eroare relativă

a – valoarea exactă,

$\tilde{a}$ – valoarea aproximativă.

Eroare absolută : $a-\tilde{a}$ **sau** $|a-\tilde{a}|$ **sau** $\|a-\tilde{a}\|$

$$a = \tilde{a} \pm \Delta_a \text{ , } |a-\tilde{a}| \le \Delta_a$$

Eroare relativă: $a \neq 0$ $\quad \dfrac{a-\tilde{a}}{a}$ **sau** $\dfrac{|a-\tilde{a}|}{|a|}$ **sau** $\dfrac{\|a-\tilde{a}\|}{\|a\|}$

$$\frac{|a-\tilde{a}|}{|a|} \le \delta_a$$ (δ_a se exprimă, de regulă, în %).

În aproximările 1kg ±5g, 50g±5g erorile absolute sunt egale dar pentru prima cantitate eroarea relativă este 0,5% iar pentru a doua eroarea relativă este 10%.

$$a_1 = \tilde{a}_1 \pm \Delta_{a_1}\ , a_2 = \tilde{a}_2 \pm \Delta_{a_2}\ ,$$

$$a_1 \pm a_2 = (\tilde{a}_1 \pm \tilde{a}_2) \pm \left(\Delta_{a_1} \pm \Delta_{a_2}\right)$$

$$\Delta_{a_1 + a_2} \leq \Delta_{a_1} + \Delta_{a_2}\ .$$

a_1 cu eroare relativă δ_{a_1} şi a_2 cu eroare relativă δ_{a_2} :

$a = a_1 * a_2$ sau $\frac{a_1}{a_2}$ rezultă $\delta_a = \delta_{a_1} + \delta_{a_2}$.

Calcul Numeric

Cursul 3

2022

Anca Ignat

Condiționare ←→ stabilitate

Condiționarea unei probleme caracterizează sensibilitatea soluției în raport cu perturbarea datelor de intrare, în ipoteza unor calcule exacte (independent de algoritmul folosit pentru rezolvarea problemei).

Fie x datele exacte de intrare, $\tilde{x}$ o aproximație cunoscută a acestora, $P(x)$ soluţia exactă a problemei şi $P(\tilde{x})$ soluţia problemei cu $\tilde{x}$ ca date de intrare. Se presupune că s-au făcut calcule exacte la obţinerea soluţiilor $P(x)$ şi $P(\tilde{x})$.

O problemă se consideră a fi ***prost condiţionată*** dacă $\boldsymbol{P(x)}$ şi $\boldsymbol{P(\tilde{x})}$ diferă mult chiar dacă eroarea relativă $\dfrac{\| x - \tilde{x} \|}{\| x \|}$ este mică.

Condiţionarea numerică a unei probleme este exprimată prin amplificarea erorii relative:

$$k(x) = \frac{\dfrac{\| P(x) - P(\tilde{x}) \|}{\| P(x) \|}}{\dfrac{\| x - \tilde{x} \|}{\| x \|}} \quad \textbf{pentru } x \neq 0 \ \textbf{ si } P(x) \neq 0$$

O valoare mică pentru ***k(x)*** caracterizează o problemă bine-condiţionată.

Condiţionarea este o proprietate locală (se evaluează pentru diverse date de intrare ***x***). O problemă este bine-condiţionată dacă este bine-condiționată în orice punct.

Eroarea relativă în datele de ieşire ≈

Număr de condiţionare × Eroarea relativă în datele de intrare

Se consideră polinomul Wilkinson:

$$w(x) = (x-1)(x-2)\cdots(x-20) = x^{20} - 210x^{19} + P_{18}(x)$$

Dacă se schimbă coeficientul ***210*** al lui x^{19} cu

$$-210 - 2^{-23} = -210.0000001192$$

soluțiile (cu 5 zecimale exacte) noului polinom sunt:

1.00000 2.00000, 3.00000, 4.00000, 5.00000, 6.00001, 6.99970, 8.00727, 8.91725, 20.84691,10.09527 ± *i*0.64350 11.79363 ± *i*1.65233, 13.99236 ± *i*2.51883, 16.73074 ± *i*2.81262, 19.50244 ± *i*1.94033

Pentru rezolvarea unei probleme $\boldsymbol{P}$, calculatorul execută un algoritm $\tilde{\boldsymbol{P}}$. Deoarece se folosesc numere în virgulă mobilă, calculele sunt afectate de erori:

$$\boldsymbol{P(x) \neq \tilde{P}(x)}$$

Stabilitatea numerică exprimă mărimea erorilor numerice introduse de algoritm, în ipoteza unor date de intrare exacte, $\boldsymbol{\| P(x) - \tilde{P}(x) \|}$ sau $\boldsymbol{\dfrac{\| P(x) - \tilde{P}(x) \|}{\| P(x) \|}}$.

O eroare relativă de ordinul erorii de rotunjire caracterizează un ***algoritm numeric stabil***.

Un **algoritm numeric stabil** aplicat unei **probleme bine condiționate** conduce la **rezultate cu precizie foarte bună**.

Un algoritm $\tilde{\boldsymbol{P}}$ destinat rezolvării problemei $\boldsymbol{P}$ este numeric stabil dacă este îndeplinită una din condițiile:

1. $\tilde{P}(x) \approx P(x)$ pentru orice intrare x;

2. există $\tilde{x}$ apropiat de x, astfel ca $\tilde{P}(x) \approx P(\tilde{x})$

x = datele exacte,

$P(x)$ = soluţia exactă folosind date exacte,

$\tilde{P}(x)$ = soluţia „*calculată*” folosind algoritmul $\tilde{P}$ cu date exacte de intrare

Rezolvarea sistemelor liniare

Istoric

- 1900 î.Hr., Babilon - apar primele probleme legate de ecuaţii liniare simultane
- 300 î.Hr. Babilon - tăbliţă cu următoarea problemă:

  ”*Avem două câmpuri de arie totală 1800 ha. Producţia la hectar pe primul câmp este de 2/3 buşel (=36,3l) iar pe al doilea este de 1/2 buşel. Dacă producţia totală este de 1100 buşeli, să se determine aria fiecărui teren în parte.*”

- 200-100 î.Hr. China – *9 capitole despre arta matematică* – metodă de rezolvare foarte asemănătoare eliminării Gauss („*Avem 3 tipuri de grâu. Ştim că 3 baloturi din primul tip, 2 baloturi din al doilea tip şi 1 balot din al treilea tip cântăresc 39 măsuri. De asemenea, 2 baloturi din primul tip, 3 baloturi din al doilea tip şi 1 balot din al treilea tip cântăresc 34 măsuri şi 1 balot din primul tip, 2 baloturi din al doilea tip şi 3 baloturi din al treilea tip cântăresc 26 măsuri. Câte măsuri cântăreşte un balot din fiecare tip de grâu”)*

- 1545, Cardan – în *Ars Magna*, propune o regulă (*regula de modo)* pentru rezolvarea unui sistem de 2 ecuaţii cu 2 necunoscute (seamănă cu regula lui Cramer)
- 1683, Seki Kowa, Japonia - ideea de „*determinant*”- „*Method of solving the dissimulated problems”*. Calculează ceea ce astăzi cunoaştem sub numele de determinant, determinanţii matricelor ***2x2***, ***3x3***, ***4x4***, ***5x5*** în legătură cu rezolvarea unor ecuaţii dar nu a sistemelor de ecuaţii.

- 1683, Leibniz într-o scrisoare către l'Hôpital explică faptul că sistemul de ecuaţii:

$$10+11x+12y=0$$
$$20+21x+22y=0$$
$$30+31x+32y=0$$

are soluţie deoarece :

10*21*32+11*22*30+12*20*31=10*22*31+11*20*32+12*21*30

(condiţia ca determinantul matricei coeficienţilor este 0).

Leibniz era convins că o notaţie matematică bună este cheia progresului şi experimentează mai mult de 50 de moduri diferite de a scrie coeficienţii unui sistem de ecuaţii. Leibniz foloseşte termenul de „*rezultant*” în loc de determinant şi a demonstrat regula lui Cramer pentru „rezultanţi”. Ştia că orice determinant poate fi dezvoltat în raport cu o coloană – operaţia se numeşte azi dezvoltarea Laplace.

- 1750, Cramer prezintă o formulă bazată pe determinanţi pentru rezolvarea unui sistem de ecuaţii liniare – *regula lui Cramer* – *„Introduction in the analysis of algebraic curves”* (dă o regulă generală pentru sisteme *n x n*:

  ‚One finds the value of each unknown by forming ***n*** *fractions of which the common denominator has as many terms as there are permutations of* ***n*** *things’*

- 1764 Bezout, 1771 Vandermonde, 1772 Laplace – reguli de calcul al determinanţilor

- 1773 Lagrange – prima utilizare implicită a matricelor în legătură cu formele biliniare ce apar la optimizarea unei funcţii reale de 2 sau mai multe variabile (dorea să caracterizeze punctele de maxim şi minim a funcţiilor de mai multe variabile)

- 1800-1801, Gauss introduce noţiunea de „*determinant”* (determină proprietăţile formei pătratice) – *Disquisitiones arithmeticae(1801)*; descrie operaţiile de înmulţire matricială şi inversă a unei matrice în contextul tabloului coeficienţilor unei forme pătratice. Gauss dezvoltă *eliminarea Gaussiană* pe când studia orbita asteroidului Pallas de unde obţine un sistem liniar cu 6 ecuaţii cu 6 necunoscute.
- 1812, Cauchy foloseşte termenul de „*determinant*” în sensul cunoscut azi.

- 1826, Cauchy găsește valorile proprii şi deduce rezultate legate de diagonalizarea unei matrice. Introduce noțiunea de matrice asemenea şi demonstrează ca acestea au aceeași ecuație caracteristică. Demonstrează că orice matrice reală simetrică este diagonalizabilă.
- 1850, Sylvester introduce pentru prima data termenul de ***matrice*** (din latină, „uter” – un loc unde ceva se formează sau este produs, „*an oblong arrangement of terms*”)

- 1855, Cayley – algebră matricială, prima definiţie abstractă a unei matrice. Studiază transformările liniare şi compunerea lor ceea ce îl conduce la operaţiile cu matrice (adunare, înmulţire, înmulţirea cu un scalar, inversa)

- 1858, Cayley în *Memoriu asupra teoriei matricelor : „Sunt multe lucruri de spus despre această teorie a matricelor şi, după părerea mea, această teorie ar trebui să preceadă teoria determinanţilor”*

- Jordan (1870 – Treatise on substitutions and algebraic equations – forma canonică Jordan), Frobenius (1878 – On linear substituions and bilinear forms, rangul unei matrici)
- 1890, Weierstrass – On determinant theory, definiţia axiomatică a determinantului
- 1925, Heisenberg reinventează algebra matricială pentru mecanica cuantică

- 1947, vonNeuman & Goldstine introduc numerele de condiţionare atunci când analizează erorile de rotunjire
- 1948, Turing introduce descompunerea LU a unei matrice
- 1958, Wilkinson dezvoltă factorizarea QR

....

Propoziție

Fie $A \in \mathbb{R}^{n \times n}$ pentru care există o normă matricială naturală astfel ca $\|A\| < 1$. Atunci există matricele $(I_n \pm A)^{-1}$ şi avem evaluările:

$$\frac{1}{1+\|A\|} \leq \left\|(I \pm A)^{-1}\right\| \leq \frac{1}{1-\|A\|}.$$

Evaluarea erorii în rezolvarea sistemelor liniare

(condiţionarea sistemelor liniare)

Fie $\boldsymbol{A \in \mathbb{R}^{n\times n}}$, $\boldsymbol{b \in \mathbb{R}^{n}}$, $\boldsymbol{x \in \mathbb{R}^{n}}$ şi sist. de ec. liniare:

$$\boldsymbol{Ax = b}$$

$\boldsymbol{A}$ **nesingulară** $\Leftrightarrow$ $\det \boldsymbol{A \neq 0}$ $\Rightarrow$ $\exists$ **sol. sist.** $\boldsymbol{x = A^{-1}b}$

Pentru erorile în datele de intrare facem notațiile:

- $\boldsymbol{\Delta A \in \mathbb{R}^{n\times n}}$ **eroarea absolută pentru** $\boldsymbol{A}$;
- $\boldsymbol{\Delta b \in \mathbb{R}^{n}}$ **eroarea absolută pentru** $\boldsymbol{b}$;

În realitate se rezolvă sistemul:

$$(A + \Delta A)\,\tilde{x} = b + \Delta b$$

soluţia fiind $\tilde{x}$:

$$\tilde{x} = x + \Delta x$$

În mod natural se ridică următoarele probleme :

1. Dacă A este matrice nesingulară, $\Delta A = ?$ a.î. $A + \Delta A$ să fie nesingulară ?
2. Pp. A şi $A + \Delta A$ nesingulare care sunt relaţiile între $\frac{\|\Delta A\|}{\|A\|}, \frac{\|\Delta b\|}{\|b\|}$ şi $\frac{\|\Delta x\|}{\|x\|}$?

1. Pp. $\boldsymbol{A}$ nesingulară.

$$\boldsymbol{A+\Delta A=A\left(I_n+A^{-1}\Delta A\right)\rightarrow}$$

$$\boldsymbol{A+\Delta A}\ \textbf{nesingulară}\ \Leftrightarrow \boldsymbol{\left(I_n+A^{-1}\Delta A\right)}\ \textbf{nesingulară}$$

Propoziţie

Fie $\boldsymbol{A}$ nesingulară şi $\|\boldsymbol{\Delta A}\|<\dfrac{\mathbf{1}}{\left\|\boldsymbol{A^{-1}}\right\|}$. Atunci $\boldsymbol{I+A^{-1}\Delta A}$ este

nesingulară şi avem:

$$\left\| \left(I_n + A^{-1}\Delta A \right)^{-1} \right\| \leq \frac{1}{1 - \left\| A^{-1} \right\| \cdot \left\| \Delta A \right\|}$$

Demonstraţie. Avem:

$$\left\| \Delta A \right\| < \frac{1}{\left\| A^{-1} \right\|} \Rightarrow \left\| A^{-1}\Delta A \right\| \leq \left\| A^{-1} \right\| \cdot \left\| \Delta A \right\| < 1 \overset{\text{Prop}}{\Rightarrow} \exists \left(I + A^{-1}\Delta A \right)^{-1}$$

$$\left\| \left(I + A^{-1}\Delta A \right)^{-1} \right\| \leq \frac{1}{1 - \left\| A^{-1}\Delta A \right\|} \leq \frac{1}{1 - \left\| A^{-1} \right\| \cdot \left\| \Delta A \right\|} \ .$$

Pp. că A este nesingulară şi $\left\| \Delta A \right\| < \frac{1}{\left\| A^{-1} \right\|}$.

$$(A+\Delta A)(x+\Delta x)=b+\Delta b \Rightarrow (A+\Delta A)\Delta x + Ax + (\Delta A)x = b+\Delta b \Rightarrow$$

$$A(I + A^{-1}\Delta A)\Delta x = \Delta b - (\Delta A)x \Rightarrow \Delta x = (I + A^{-1}\Delta A)^{-1} A^{-1}[\Delta b - (\Delta A)x] \Rightarrow$$

$$\|\Delta x\| \le \left\|(I + A^{-1}\Delta A)^{-1}\right\| \left\|A^{-1}\right\| (\|\Delta b\| + \|\Delta A\| \|x\|) \Rightarrow$$

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\left\|A^{-1}\right\|}{1 - \left\|A^{-1}\right\| \|\Delta A\|} \left(\frac{\|\Delta b\|}{\|x\|} + \|\Delta A\| \right) \qquad \textbf{(1)}$$

Din $Ax = b$ obţinem $\|b\| \le \|A\| \|x\| \Rightarrow \frac{1}{\|x\|} \le \frac{\|A\|}{\|b\|}$ şi ţinând seamă

de acest rezultat, din (1) deducem:

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\left\|A^{-1}\right\| \|A\|}{1 - \left\|A^{-1}\right\| \|\Delta A\|} \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right).$$

$k(A) = \|A^{-1}\| \, \|A\|$ ***numărul de condiţionare*** al matricei A.

Propoziţie

Dacă matricea A este nesingulară şi $\|\Delta A\| < \dfrac{1}{\|A^{-1}\|}$ atunci:

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{k(A)}{1 - k(A) \cdot \dfrac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|} \right).$$

Din $I_n = A \, A^{-1}$ rezultă $1 = \|I_n\| \le \|A\| \|A^{-1}\| = k(A)$.

$k(A) \ge 1, \ \forall A$ dar dep. de norma matricială naturală utilizată.

O matrice ***A*** pentru care numărul de condiţionare este mare se numeşte matrice ***prost condiţionată (k(A) ‚mare’).***

Ax=b **cu** ***k(A) mare*** $\rightarrow$ $\frac{\|\Delta x\|}{\|x\|}$ poate fi mare chiar dacă erorile relative $\frac{\|\Delta b\|}{\|b\|}$ **şi** $\frac{\|\Delta A\|}{\|A\|}$ sunt mici.

Fie $\boldsymbol{A}$ o matrice simetrică $\boldsymbol{A = A^T}$, nesingulară. Utilizând norma matricială subordonată normei vectoriale euclidiene:

$$\|\boldsymbol{A}\|_2 = \sqrt{\rho\left(\boldsymbol{A^T A}\right)} = \sqrt{\rho\left(\boldsymbol{A^2}\right)}$$

$$\boldsymbol{k(A)} = \|\boldsymbol{A}\|_2 \cdot \left\|\boldsymbol{A^{-1}}\right\|_2$$

Matricea simetrică $\boldsymbol{A}$ are valorile proprii reale $\boldsymbol{\lambda_1, \lambda_2, ..., \lambda_n}$,

$\boldsymbol{A^2}$ are valorile proprii $\boldsymbol{\lambda_1^2, \lambda_2^2, ..., \lambda_n^2}$

$\boldsymbol{A^{-1}}$ are valorile proprii $\boldsymbol{\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, ..., \frac{1}{\lambda_n}}$.

$$|\lambda_1| \le |\lambda_2| \le .... \le |\lambda_n| \Rightarrow \rho(A) = |\lambda_n| \quad \text{şi} \quad \rho\left(A^{-1}\right) = \frac{1}{|\lambda_1|}$$

$$A = A^T \rightarrow \|A\|_2 = \rho(A) = |\lambda_n|, \left\|A^{-1}\right\|_2 = \rho\left(A^{-1}\right) = \frac{1}{|\lambda_1|},$$

$k_2(A) = \| A \|_2 \cdot \| A^{-1} \|_2 = \dfrac{|\lambda_n|}{|\lambda_1|}$ ***număr de condiţionare spectral.***

A matrice ortogonală $\rightarrow$ $k_2(A)=1$

$$A^T A = A \cdot A^T = I_n \Rightarrow A^{-1} = A^T$$

$$\|A\|_2 = \sqrt{\rho\left(A^T A\right)} = \sqrt{\rho(I)} = 1 = \left\|A^T\right\|_2$$

$$k_2(A) = \|A\|_2 \cdot \left\|A^{-1}\right\|_2 = \|A\|_2 \cdot \left\|A^T\right\|_2 = 1,$$

Matrice aproape singulară dar cu număr de condiţionare mic:

$$A = \mathbf{diag}\,[1, 0.1, 0.1, ..., 0.1] \in \mathbb{R}^{100\times 100} \Rightarrow \det A = 1 \cdot (0.1)^{99} = 10^{-99}$$

$$\| A \|_2 = 1 \ , \ \| A^{-1} \|_2 = 10 \ \Rightarrow \ k_2(A) = \| A \|_2 \ \| A^{-1} \|_2 = 10$$

Matrice foarte prost condiționată cu det. nenul (**det A=1**):

$$A=\begin{pmatrix} 1 & 2 & 0 & \cdots & 0 \\ 0 & 1 & 2 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 2 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix},$$

$$A^{-1}=\begin{pmatrix} 1 & -2 & 4 & \cdots & (-2)^{i-1} & \cdots & (-2)^{n-1} \\ 0 & 1 & -2 & \cdots & (-2)^{i-2} & \cdots & (-2)^{n-2} \\ \vdots & & & & & & \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

$$\| A \|_\infty = \| A \|_1 = 3 \ ,$$

$$\| A^{-1} \|_\infty = \| A^{-1} \|_1 = 1 + 2 + \cdots + 2^{n-1} = 2^n - 1$$

$$n = 100 \ \Rightarrow \ k(A) = \| A \|_\infty \| A^{-1} \|_\infty = \| A \|_1 \| A^{-1} \|_1 = 3 \cdot (2^{100} - 1)$$

$$\det A = 1$$

$$\begin{cases} x + y = 2 \\ x + 1.001y = 2 \end{cases}, \quad \begin{cases} x + y = 2 \\ x + 1.001y = 2.001 \end{cases} \qquad k(A) = 4002$$

$$x = 2,\ y = 0 \qquad\qquad x = 1,\ y = 1$$

$$\begin{cases} 400x - 201y = 200 \\ -800x + 401y = -200 \end{cases}, \quad \begin{cases} 401x - 201y = 200 \\ -800x + 401y = -200 \end{cases}$$

$$x = -100,\ y = -200 \qquad\qquad x = 40000,\ y = 79800$$

$$k_2(A) = 2503 \qquad\qquad k_2(A) = 1002000$$

$$\begin{cases} 1.2969x + 0.8648\ y = 0.8642 \\ 0.2161x + 0.1441\ y = 0.1440 \end{cases} \qquad x = 2\ ,\ \ y = -2 \qquad k_2(A) = 249730000$$

$$\overline{x} = 0.9911,\ \ \overline{y} = -0.4870\ \ ,$$

$$r = b - Az = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix} - \begin{pmatrix} 1.2969 & 0.8648 \\ 0.1441 & 0.1441 \end{pmatrix} \begin{pmatrix} 0.9911 \\ -0.4870 \end{pmatrix} = \begin{pmatrix} 10^{-8} \\ -10^{-8} \end{pmatrix}$$

Matricea Hilbert

$$H = (h_{ij})_{i,j=1}^{n} \quad , \quad h_{ij} = \frac{1}{i+j-1} = \int_0^1 x^{i+j-2}\, dx$$

$$k_2(H_n) \approx \frac{(\sqrt{2}+1)^{4(n+1)}}{2^{\frac{15}{4}}\sqrt{\pi n}} \sim e^{3.5n}$$

n	$k_2(H_n)$	n	$k_2(H_n)$
1	1	7	$4.753 \cdot 10^8$
2	19.281	8	$1.526 \cdot 10^{10}$
3	$5.241 \cdot 10^2$	9	$4.932 \cdot 10^{11}$
4	$1.551 \cdot 10^4$	10	$1.602 \cdot 10^{13}$
5	$4.766 \cdot 10^5$	11	$5.220 \cdot 10^{14}$
6	$1.495 \cdot 10^7$	12	$1.678 \cdot 10^{16}$

$$H^{-1} = (g_{ij}) \quad g_{ij} = \frac{(-1)^{i+j}}{(i+j-1)} \frac{(n+i-1)!\,(n+j-1)!}{\left[(i-1)!(j-1)!\right]^2 (n-i)!(n-j)!}$$

Metode numerice de rezolvarea sistemelor liniare

Fie matricea nesingulară $A \in \mathbb{R}^{n \times n}$ şi $b \in \mathbb{R}^n$. Rezolvarea sistemului de ecuații liniare $Ax=b$ se poate face folosind ***regula lui Cramer:***

$$x_i = \frac{\det A_i(b)}{\det A}, i = 1, \ldots, n,$$

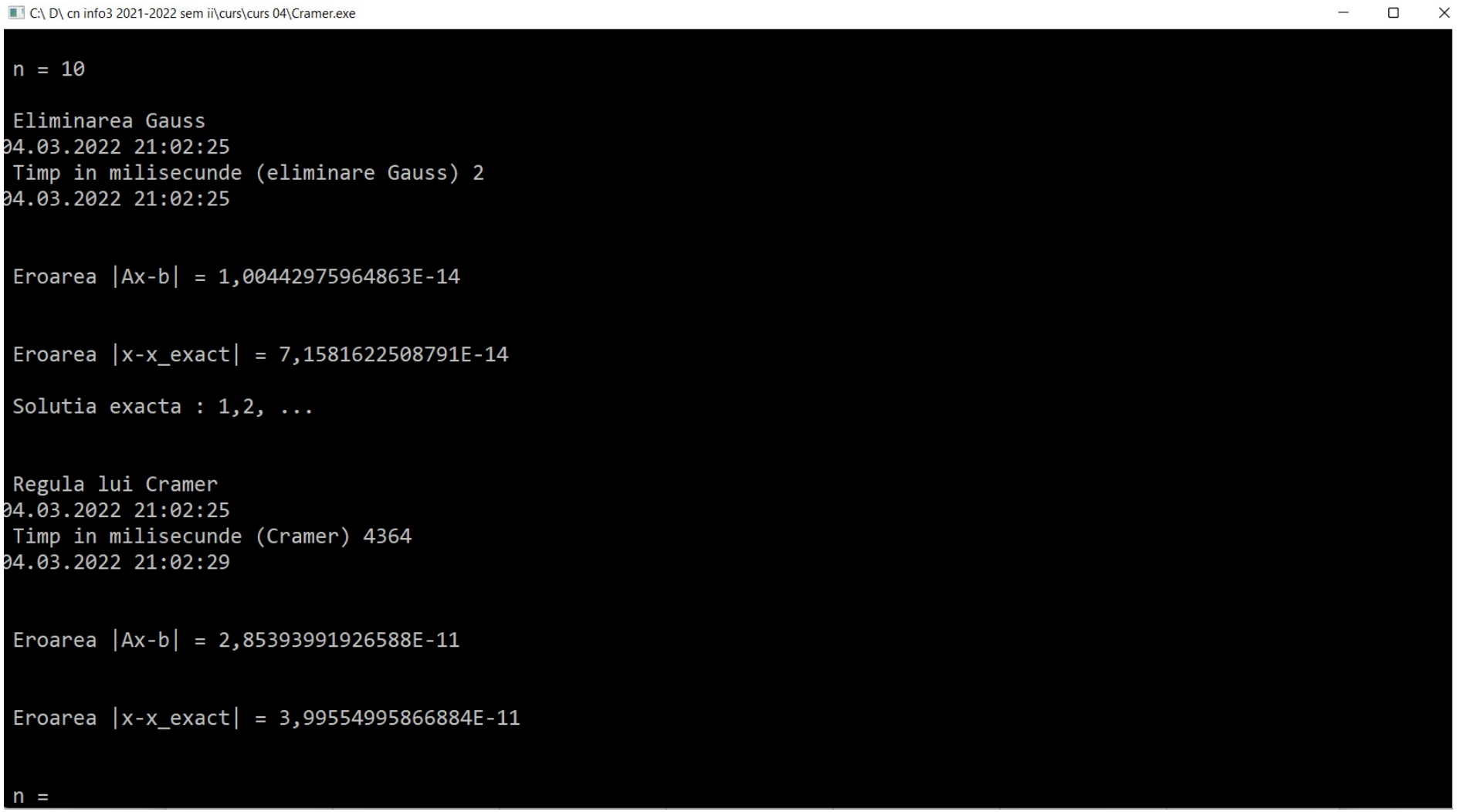
în care $A_i(b)$ se obține din matricea A prin înlocuirea coloanei i cu vectorul b.

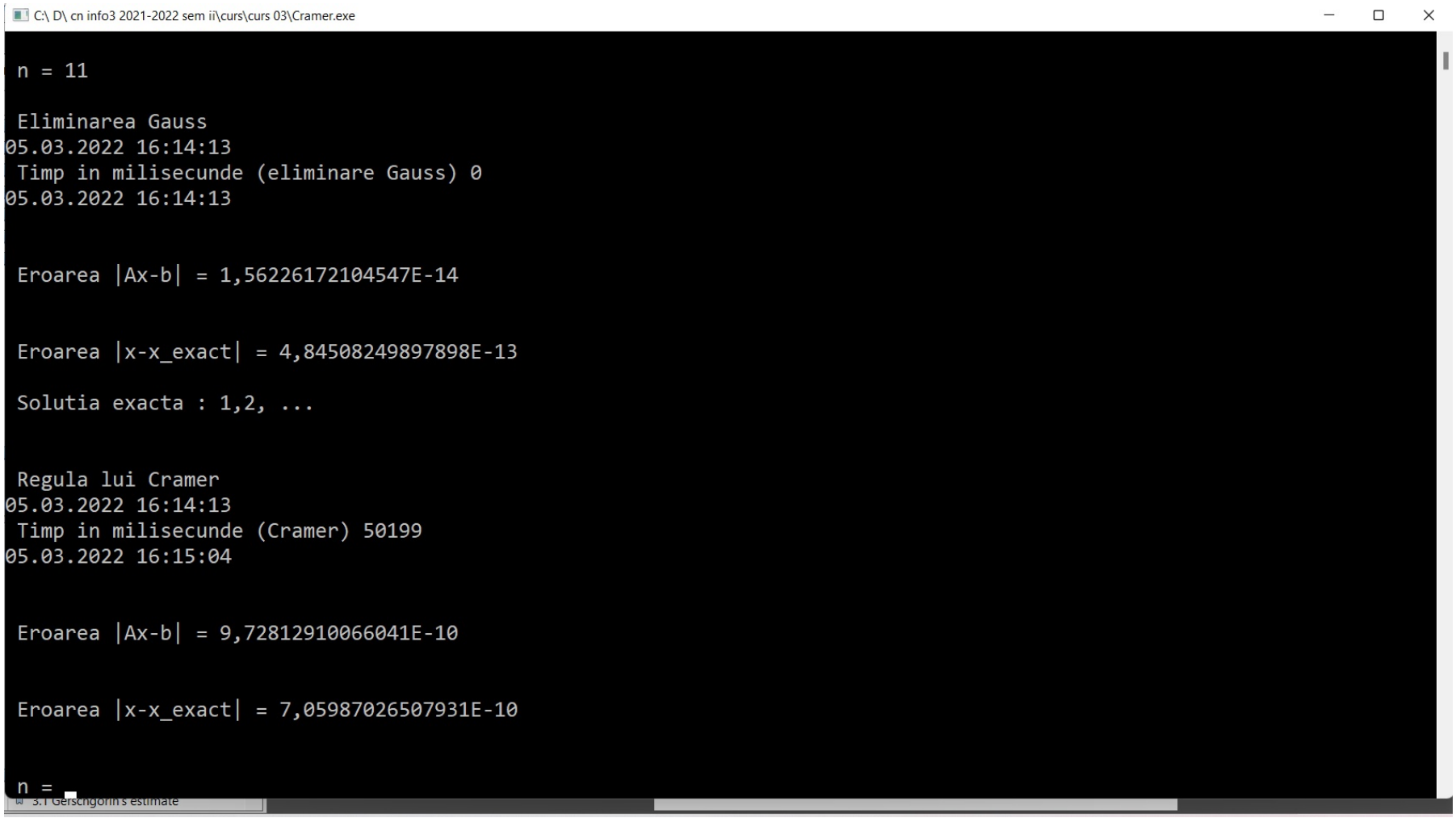
Algoritmul dat de regula lui Cramer este foarte costisitor din punct de vedere al resurselor și instabil numeric.

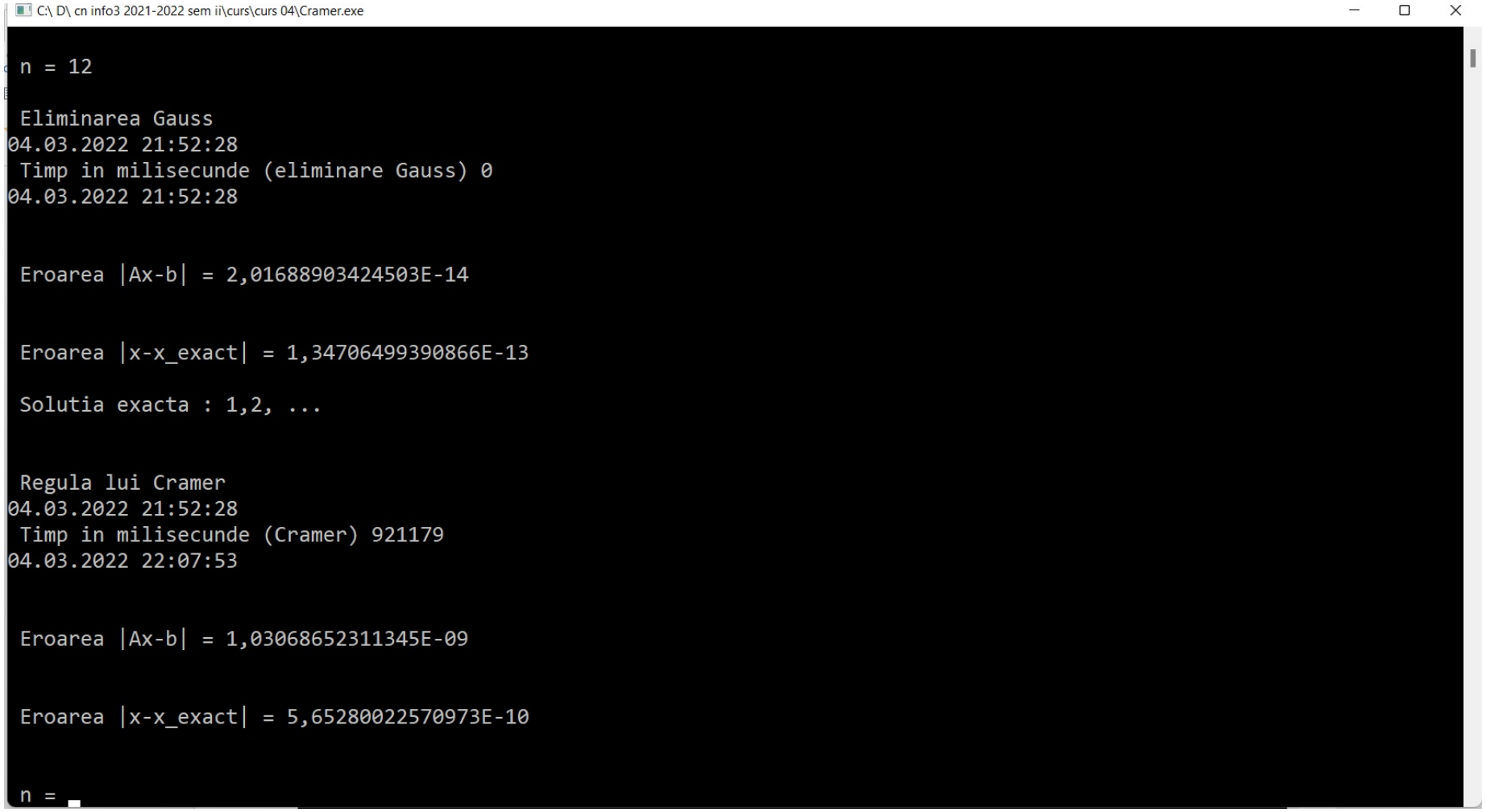
Din aceste motive s-au căutat alte metode de aproximare a soluției $\boldsymbol{x}$**.** Unul din cele mai folosiți algoritmi este **algoritmul de eliminare Gauss :**

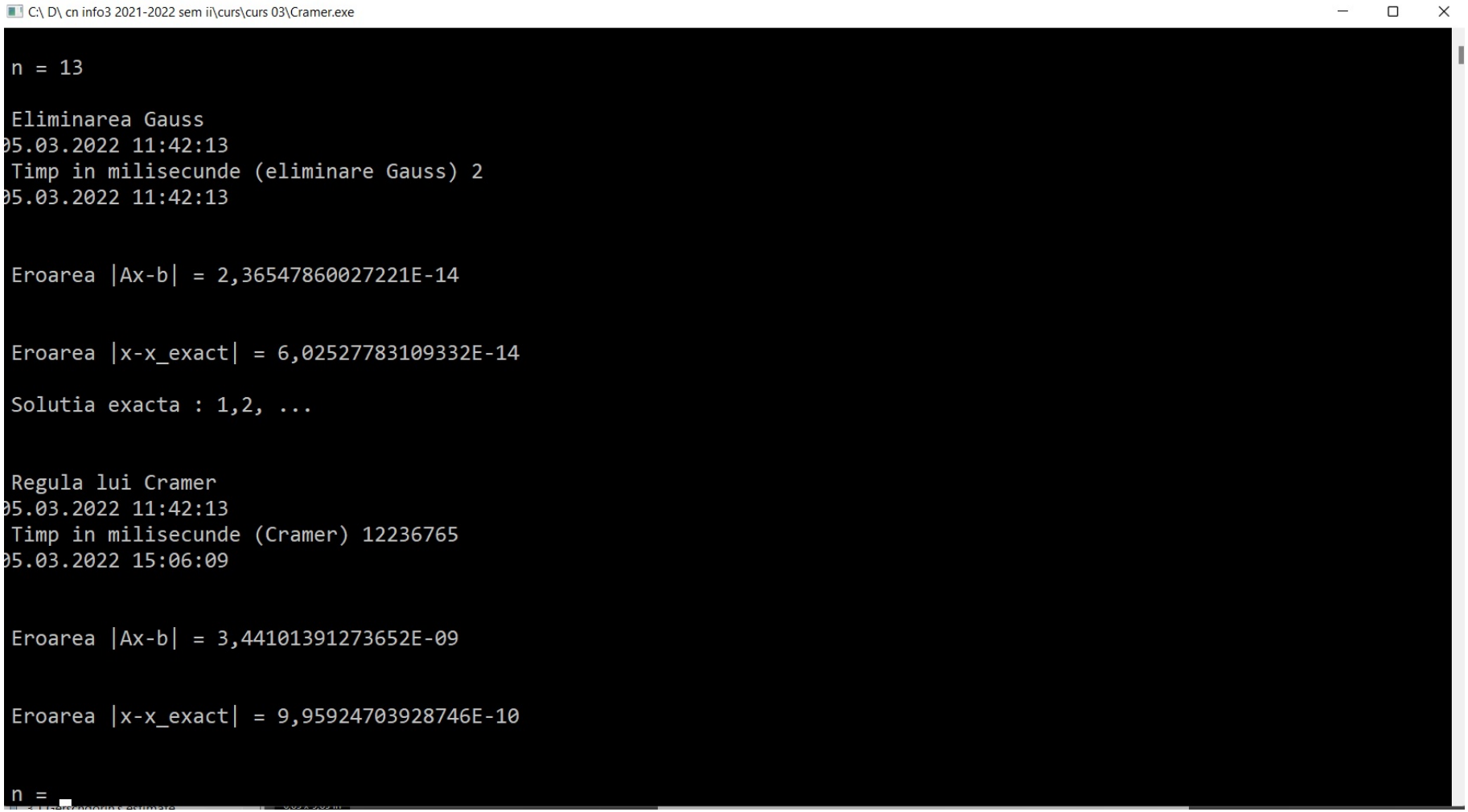
$$\boldsymbol{Ax=b} \quad \leftrightarrow \boldsymbol{\tilde{A}x=\tilde{b}}\ \textbf{cu}\ \boldsymbol{\tilde{A}}\ \text{matrice superior triunghiulară}$$

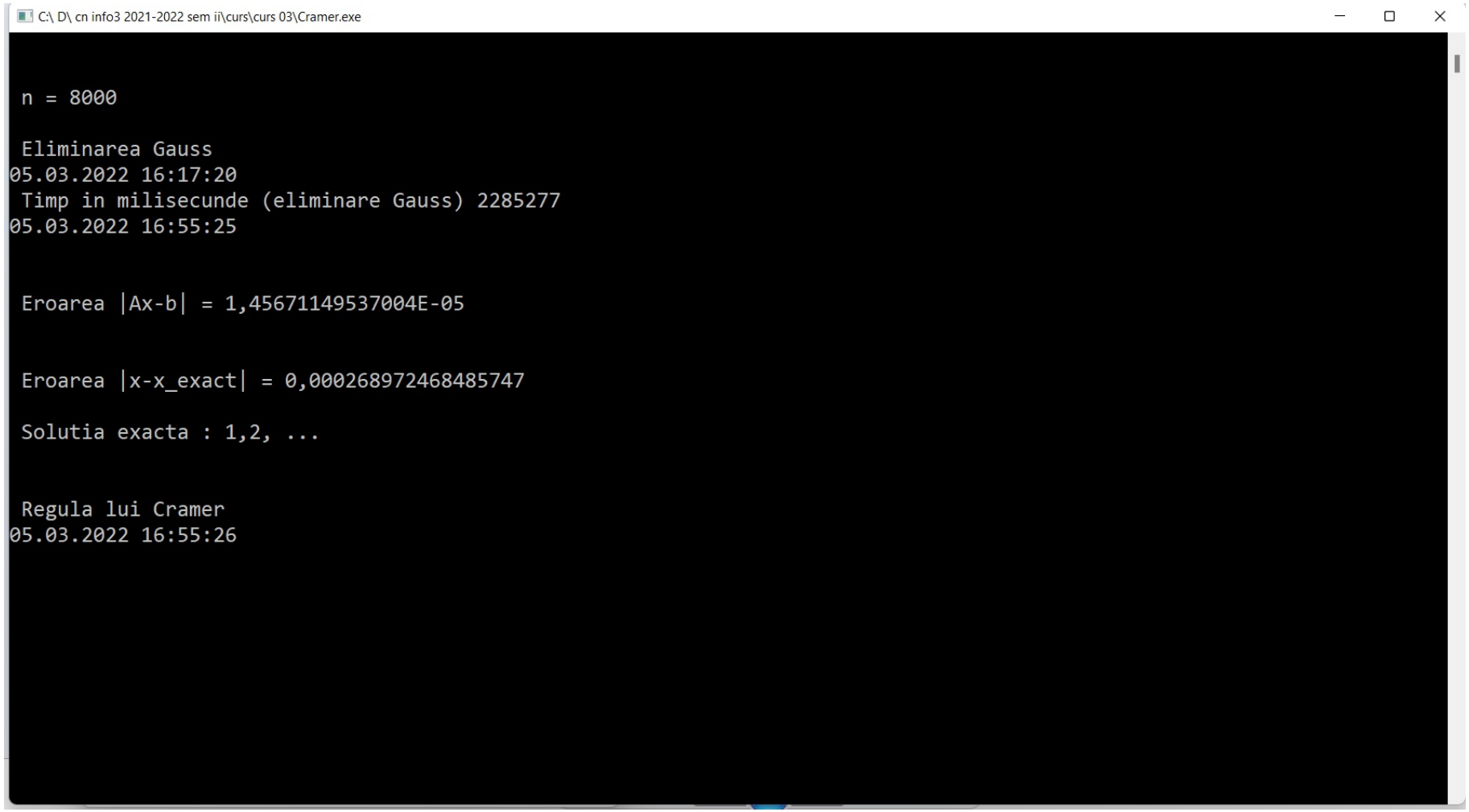
$$\boldsymbol{x=A^{-1}b=\tilde{A}^{-1}\tilde{b}} \quad \textbf{(notăm } \boldsymbol{Ax=b\sim\widetilde{A}x=\tilde{b}}\textbf{)}$$


C:\ D\ cn info3 2021-2022 sem ii\curs\curs 04\Cramer.exe
n = 10
Eliminarea Gauss
04.03.2022 21:02:25
Timp in milisecunde (eliminare Gauss) 2
04.03.2022 21:02:25
Eroarea |Ax-b| = 1,00442975964863E-14
Eroarea |x-x_exact| = 7,1581622508791E-14
Solutia exacta : 1,2, ...
Regula lui Cramer
04.03.2022 21:02:25
Timp in milisecunde (Cramer) 4364
04.03.2022 21:02:29
Eroarea |Ax-b| = 2,85393991926588E-11
Eroarea |x-x_exact| = 3,99554995866884E-11
n =


C:\ D\ cn info3 2021-2022 sem ii\curs\curs 03\Cramer.exe
n = 11
Eliminarea Gauss
05.03.2022 16:14:13
Timp in milisecunde (eliminare Gauss) 0
05.03.2022 16:14:13
Eroarea |Ax-b| = 1,56226172104547E-14
Eroarea |x-x_exact| = 4,84508249897898E-13
Solutia exacta : 1,2, ...
Regula lui Cramer
05.03.2022 16:14:13
Timp in milisecunde (Cramer) 50199
05.03.2022 16:15:04
Eroarea |Ax-b| = 9,72812910066041E-10
Eroarea |x-x_exact| = 7,05987026507931E-10
n =


C:\ D\ cn info3 2021-2022 sem ii\curs\curs 04\Cramer.exe
n = 12
Eliminarea Gauss
04.03.2022 21:52:28
Timp in milisecunde (eliminare Gauss) 0
04.03.2022 21:52:28
Eroarea |Ax-b| = 2,01688903424503E-14
Eroarea |x-x_exact| = 1,34706499390866E-13
Solutia exacta : 1,2, ...
Regula lui Cramer
04.03.2022 21:52:28
Timp in milisecunde (Cramer) 921179
04.03.2022 22:07:53
Eroarea |Ax-b| = 1,03068652311345E-09
Eroarea |x-x_exact| = 5,65280022570973E-10
n =


C:\ D\ cn info3 2021-2022 sem ii\curs\curs 03\Cramer.exe
n = 13
Eliminarea Gauss
05.03.2022 11:42:13
Timp in milisecunde (eliminare Gauss) 2
05.03.2022 11:42:13
Eroarea |Ax-b| = 2,36547860027221E-14
Eroarea |x-x_exact| = 6,02527783109332E-14
Solutia exacta : 1,2, ...
Regula lui Cramer
05.03.2022 11:42:13
Timp in milisecunde (Cramer) 12236765
05.03.2022 15:06:09
Eroarea |Ax-b| = 3,44101391273652E-09
Eroarea |x-x_exact| = 9,95924703928746E-10
n =


C:\ D\ cn info3 2021-2022 sem ii\curs\curs 03\Cramer.exe
n = 8000
Eliminarea Gauss
05.03.2022 16:17:20
Timp in milisecunde (eliminare Gauss) 2285277
05.03.2022 16:55:25
Eroarea |Ax-b| = 1,45671149537004E-05
Eroarea |x-x_exact| = 0,000268972468485747
Solutia exacta : 1,2, ...
Regula lui Cramer
05.03.2022 16:55:26

Metoda substituției

Fie sistemul liniar $\boldsymbol{Ax = b}$ unde matricea sistemului $\boldsymbol{A}$ este triunghiulară. Pentru a găsi soluția unică a sistemului, trebuie ca matricea să fie nesingulară. Determinantul matricelor triunghiulare este dat de formula:

$$\det A = a_{11}a_{22}\cdots a_{nn}$$

$$\det A \neq 0 \ , a_{ii} \neq 0 \quad \forall i = 1, 2, \ldots, n$$

Vom considera întâi cazul când matricea A este inferior triunghiulară. Sistemul are forma:

$$
\begin{aligned}
& a_{11}x_1 = b_1 \\
& a_{21}x_1 + a_{22}x_2 = b_2 \\
& \vdots \\
& a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ii}x_i = b_i \\
& \vdots \\
& a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{ni}x_i + \cdots + a_{nn}x_n = b_i
\end{aligned}
\qquad (1)
$$

Necunoscutele $x_1, x_2, \ldots, x_n$, se deduc folosind ecuațiile sistemului de la prima către ultima.

Din prima ecuație se deduce x_1:

$$x_1 = \frac{b_1}{a_{11}} \qquad (2)$$

Din a doua ecuație , utilizând valoarea x_1 din (2), obținem x_2:

$$x_2 = \frac{b_2 - a_{21}x_1}{a_{22}}$$

Când ajungem la ecuația $\boldsymbol{i}$:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ii-1}x_{i-1} + a_{ii}x_i = b_i$$

folosind variabilele $x_1, x_2, \ldots, x_{i-1}$ calculate anterior, avem:

$$x_i = \frac{b_i - a_{i1}x_1 - \cdots - a_{ii-1}x_{i-1}}{a_{ii}}$$

Din ultima ecuație se deduce x_n astfel:

$$x_n = \frac{b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{nn-1}x_{n-1}}{a_{nn}}$$

Algoritmul de calcul al soluției sistemelor (1) cu matrice inferior triunghiulară este următorul:

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j}{a_{ii}}, i = 1, 2, \ldots, n-1, n$$

Acest algoritm se numește ***metoda substituției directe***.

Vom considera, în continuare sistemul (1) cu matrice superior triunghiulară :

$$a_{11}x_1 + \cdots + a_{1i}x_i + \cdots + a_{1n-1}x_{n-1} + a_{1n}x_n = b_1$$
$$\ddots$$
$$a_{ii}x_i + \cdots + a_{in-1}x_{n-1} + a_{in}x_n = b_i$$
$$\ddots$$
$$a_{n-1n-1}x_{n-1} + a_{n-1n}x_n = b_{n-1}$$
$$a_{nn}x_n = b_n$$

Necunoscutele $x_1, x_2, \ldots, x_n$ se deduc pe rând, folosind ecuațiile sistemului, de la ultima către prima.

Din ultima ecuație găsim x_n:

$$x_n = \frac{b_n}{a_{nn}}$$

Folosind valoarea lui x_n dedusă mai sus, din penultima ecuație obținem:

$$x_{n-1} = \frac{b_{n-1} - a_{n-1n} x_n}{a_{n-1n-1}}$$

Când ajungem la ecuația i:

$$a_{ii}x_i + a_{ii+1}x_{i+1} + \cdots + a_{in}x_n = b_i$$

se cunosc deja $x_{i+1}, x_{i+2}, \ldots, x_n$ de unde ducem:

$$x_i = \frac{b_i - a_{ii+1}x_{i+1} - \cdots - a_{in}x_n}{a_{ii}}$$

Din prima ecuație găsim valoarea lui x_1:

$$x_1 = \frac{b_1 - a_{12}x_2 - \cdots - a_{1n}x_n}{a_{11}}$$

Procedeul descris mai sus se numește de ***metoda substituției inverse*** pentru rezolvarea sistemelor liniare cu matrice superior triunghiulară:

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}}, i = n, n-1, \cdots, 2, 1.$$

M - numărul de operații ***, /** (înmulțiri/împărțiri) efectuate

A - numărul operațiilor ± (adunări/scăderi) efectuate.

Atunci pentru calculul componentei x_i se efectuează

$M=n-i+1$, $A=n-i$ şi în total:

$$M=\sum_{i=n}^{1}(n-i+1)=\sum_{k=1}^{n}k=\frac{n(n+1)}{2},$$

$$A=\sum_{i=n}^{1}(n-i)=\sum_{k=1}^{n-1}k=\frac{n(n-1)}{2}$$

Efortul de calcul pentru metoda substituției directe este

$$M=\frac{n(n+1)}{2}\qquad A=\frac{n(n-1)}{2}.$$

Calcul Numeric

Cursul 4

2022

Anca Ignat

Algoritmul de eliminare Gauss

Algoritmul se realizează în ***n-1*** pași prin transformarea sistemului dat într-un sistem echivalent cu matrice triunghiulară superior.

Pas 1

la acest pas se obține sistemul:

$\boldsymbol{A^{(1)}x = b^{(1)} \sim Ax = b,}$ unde $\boldsymbol{A^{(1)}}$ are prima coloană în formă superior triunghiulară.

Pas 2

se construiește sistemul

$\boldsymbol{A^{(2)}x = b^{(2)} \sim Ax = b,}$ unde $\boldsymbol{A^{(2)}}$ are primele două coloane în formă superior triunghiulară.

⋮

Pasul r

se obține sistemul $\boldsymbol{A^{(r)}x = b^{(r)} \sim Ax = b,}$ unde $\boldsymbol{A^{(r)}}$ are primele $\boldsymbol{r}$ coloane în formă superior triunghiulară.

⋮

Pasul n-1 :

se obține sistemul

$\boldsymbol{A^{(n-1)}x = b^{(n-1)} \sim Ax = b,}$ unde $\boldsymbol{A^{(n-1)}}$ are primele $\boldsymbol{n\text{-}1}$ coloane în formă superior triunghiulară.

Dacă la un anumit pas matricea $\boldsymbol{A^{(r)}}$ **nu poate fi construită** aceasta ne va arăta că matricea $\boldsymbol{A}$ **este singulară**.

În realizarea acestor pași se utilizează următoarele operații elementare:

- **înmulțirea unei ecuații cu un factor și adunarea la altă ecuație**;

- **interschimbarea a două ecuaţii și/sau două coloane în matricea** $\boldsymbol{A}$.

Pasul 1

Intrare : sistemul $Ax=b$

Ieșire : sistemul $A^{(1)}x = b^{(1)} \sim Ax = b$, matricea $A^{(1)}$ are prima coloană în formă superior triunghiulară.

Fie ecuația i, cu $i=1,\ldots,n$

$$E_i : \quad a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i .$$

Presupunem $a_{11} \neq 0$. Operațiile efectuate au ca obiectiv anularea coeficienților lui x_1 din ecuațiile de la 2 la n şi sunt descrise în continuare:

$$E_1 * \left(\frac{-a_{21}}{a_{11}} \right) + E_2 = E_2^{(1)} \quad \Rightarrow \quad a_{21}^{(1)} = 0$$

$$\vdots$$

$$E_1 * \left(\frac{-a_{i1}}{a_{11}} \right) + E_i = E_i^{(1)} \quad \Rightarrow \quad a_{i1}^{(1)} = 0$$

$$\vdots$$

$$E_1 * \left(\frac{-a_{n1}}{a_{11}} \right) + E_n = E_n^{(1)} \quad \Rightarrow \quad a_{n1}^{(1)} = 0$$

Sistemul obținut prin aceste operații are forma:

$$\begin{cases} a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + \cdots\cdots + a_{1n}^{(1)} x_n = b_1^{(1)} \\ \qquad a_{22}^{(1)} x_2 + \cdots\cdots + a_{2n}^{(1)} x_n = b_2^{(1)} \\ \qquad \vdots \\ \qquad a_{i2}^{(1)} x_2 + \cdots\cdots + a_{in}^{(1)} x_n = b_i^{(1)} \\ \qquad \vdots \\ \qquad a_{n2}^{(1)} x_2 + \cdots\cdots + a_{nn}^{(1)} x_n = b_n^{(1)} \end{cases}$$

Pas 2

Intrare : $\boldsymbol{A^{(1)}x = b^{(1)}}$

Ieșire : $\boldsymbol{A^{(2)}x = b^{(2)} \sim Ax = b,}$ $\boldsymbol{A^{(2)}}$ are primele două coloane în formă superior triunghiulară.

Se presupune $\boldsymbol{a_{22}^{(1)} \neq 0}$ și se urmărește anularea elementelor $\boldsymbol{a_{32}^{(2)}, a_{42}^{(2)}, \ldots, a_{n2}^{(2)}}$ (transformarea coloanei 2 în formă superior triunghiulară). Operațiile efectuate asupra ecuațiilor $\boldsymbol{E_i^{(1)}, i = 3, \cdots, n}$ sunt următoarele :

$$\begin{cases} E_2^{(1)} * \left(-\dfrac{a_{32}^{(1)}}{a_{22}^{(1)}} \right) + E_3^{(1)} = E_3^{(2)} \;\Rightarrow\; a_{32}^{(2)} = 0; \\ \vdots \\ E_2^{(1)} * \left(-\dfrac{a_{i2}^{(1)}}{a_{22}^{(1)}} \right) + E_i^{(1)} = E_i^{(2)} \;\Rightarrow\; a_{i2}^{(2)} = 0; \\ \vdots \\ E_2^{(1)} * \left(-\dfrac{a_{n2}^{(1)}}{a_{22}^{(1)}} \right) + E_n^{(1)} = E_n^{(2)} \;\Rightarrow\; a_{n2}^{(2)} = 0; \end{cases}$$

Se observă că nu se schimbă forma superior triunghiulară a primei coloane.

Pas r

Intrare : $\boldsymbol{A^{(r-1)}x = b^{(r-1)}}$

Ieșire : $\boldsymbol{A^{(r)}x = b^{(r)} \sim Ax = b,}$ $\boldsymbol{A^{(r)}}$ are primele $\boldsymbol{r}$ coloane în formă superior triunghiulară.

Sistemul are următoarea formă:

$$
\begin{cases}
a_{11}^{(r-1)}x_1+\cdots+a_{1r}^{(r-1)}x_r+\cdots+a_{1n}^{(r-1)}x_n=b_1^{(r-1)} \\
\quad\ddots \\
\qquad\ddots \\
\qquad\qquad a_{rr}^{(r-1)}x_r+\cdots+a_{rn}^{(r-1)}x_n=b_r^{(r-1)} \\
\qquad\qquad a_{r+1r}^{(r-1)}x_r+\cdots+a_{r+1n}^{(r-1)}x_n=b_{r+1}^{(r-1)} \\
\qquad\qquad\quad\vdots \\
\qquad\qquad a_{ir}^{(r-1)}x_r+\cdots+a_{in}^{(r-1)}x_n=b_i^{(r-1)} \\
\qquad\qquad\quad\vdots \\
\qquad\qquad a_{nr}^{(r-1)}x_r+\cdots+a_{nn}^{(r-1)}x_n=b_n^{(r-1)}
\end{cases}
$$

Presupunem $a_{rr}^{(r-1)} \neq 0$.

Vom urmări anularea elementelor $a_{r+1r}^{(r)}, a_{r+2r}^{(r)}, \cdots, a_{nr}^{(r)}$.

$$\begin{cases} E_r^{(r-1)} * \left(-\dfrac{a_{r+1r}^{(r-1)}}{a_{rr}^{(r-1)}} \right) + E_{r+1}^{(r-1)} = E_{r+1}^{(r)} \quad \Rightarrow \quad a_{r+1r}^{(r)} = 0; \\ \vdots \\ E_r^{(r-1)} * \left(-\dfrac{a_{ir}^{(r-1)}}{a_{rr}^{(r-1)}} \right) + E_i^{(r-1)} = E_i^{(r)} \quad \Rightarrow \quad a_{ir}^{(r)} = 0; \\ \vdots \\ E_r^{(r-1)} * \left(-\dfrac{a_{nr}^{(r-1)}}{a_{rr}^{(r-1)}} \right) + E_n^{(r-1)} = E_n^{(r)} \quad \Rightarrow \quad a_{nr}^{(r)} = 0; \end{cases}$$

Se observă că nu se schimbă forma superior triunghiulară a primelor ***r-1*** coloane.

La fiecare pas s-a făcut ipoteza $\boldsymbol{a_{rr}^{(r-1)} \neq 0}$. Elementul $\boldsymbol{a_{rr}^{(r-1)}}$ poartă numele de ***pivot***. În cazul în care elementul pivot este nul se pot aplica următoarele strategii, numite de ***pivotare***:

Pivotare (stabilitate + $a_{rr}^{(r-1)} \neq 0$?)

1^0 ***Fără pivotare***

Se caută primul indice $i_0 \in \{r, r+1, \cdots, n\}$ astfel încât $a_{i_0 r}^{(r-1)} \neq 0$. Se interschimbă ecuaţiile i_0 și r.

Să observăm că în procesul de calcul la pasul r intervine factorul $\frac{1}{a_{rr}^{(r-1)}}$ astfel că valori mici ale lui $\left|a_{rr}^{(r-1)}\right|$ conduc la amplificarea erorilor de calcul. Pentru a asigura stabilitatea numerică a procesului de calcul este de dorit ca $\left|a_{rr}^{(r-1)}\right|$ să fie ***'mare'***.

2^0 *Pivotare parțială*

Se determină indicele i_0:

$$\left|a_{i_0 r}^{(r-1)}\right| = \max\left\{\left|a_{ir}^{(r-1)}\right|; i = r,\ldots,n\right\}$$

și se inter-schimbă ecuațiile i_0, r dacă $i_0 \neq r$.

3^0 *Pivotare totală*

Se determină indicii i_0 şi j_0:

$$\left|a_{i_0 j_0}^{(r-1)}\right| = \max\left\{\left|a_{ij}^{(r-1)}\right|; i = r,\ldots,n, j = r,\ldots,n\right\}$$

și se inter-schimbă ecuațiile i_0, r dacă $i_0 \neq r$ şi coloanele j_0, r dacă $j_0 \neq r$

Schimbarea coloanelor implică schimbarea ordinii variabilelor astfel încât în final va trebui refăcută ordinea inițială a variabilelor.

Dacă după pivotare elementul pivot rămâne nul, $\boldsymbol{a_{rr}^{(r-1)} = 0}$, atunci putem deduce că $\boldsymbol{A^{(r-1)}}$ este singulară.

În adevăr, dacă în procesul de pivotare parțială $\boldsymbol{a_{rr}^{(r-1)} = 0}$, atunci

$$A^{(r-1)} = \begin{bmatrix} a_{11}\ a_{12} & \cdots\cdots\cdots & a_{1n} \\ & \ddots & \\ & a_{rr} = 0 \cdots & a_{rn} \\ & 0 & \\ & 0 & \\ & 0 \cdots\cdots\cdots & a_{nn} \end{bmatrix} \Rightarrow$$

$$\det A^{(r-1)} = a_{11}^{(r-1)} a_{22}^{(r-1)} \cdots a_{r-1r-1}^{(r-1)} \det \begin{bmatrix} 0 \cdots\cdots & a_{rn} \\ 0 & \\ \vdots & \\ 0 \cdots\cdots & a_{nn} \end{bmatrix} = 0$$

Deoarece operațiile efectuate (cele de interschimbare de ecuaţii și/sau coloane) nu au schimbat decât semnul determinantului avem:

$$\det A = \pm \det A^{(r-1)} = 0 \;\Rightarrow\; \det A = 0$$

prin urmare matricea A inițială este singulară.

Și în cazul procesului de pivotare totală dacă $a_{rr}^{(r-1)} = 0$, atunci:

$$A^{(r-1)} = \begin{bmatrix} a_{11}\ a_{12} \cdots\cdots\cdots a_{1n} \\ \ddots \\ a_{rr} = 0 \cdots \mathbf{0} \\ 0 \\ 0 \\ 0 \cdots\cdots\cdots \mathbf{0} \end{bmatrix} \Rightarrow$$

$$\det A^{(r-1)} = a_{11}^{(r-1)} a_{22}^{(r-1)} \cdots a_{r-1r-1}^{(r-1)} \det \begin{bmatrix} \mathbf{0} \cdots\cdots \mathbf{0} \\ \mathbf{0} \cdots\cdots \mathbf{0} \\ \vdots \qquad\quad \vdots \\ \mathbf{0} \cdots\cdots \mathbf{0} \end{bmatrix} = \mathbf{0}$$

$\det A = \pm \det A^{(r-1)} = \mathbf{0} \Rightarrow A$ este matrice singulară.

$r = 1$;

pivotare(r**);**

while ($r \leq n-1$ **şi** $|a_{rr}| > \varepsilon$**)**

// Pas r

- **for** $i = r+1, \ldots, n$
  - $f = -\dfrac{a_{ir}}{a_{rr}}$;
  - **for** $j = r+1, \ldots, n$

    $a_{ij} = a_{ij} + f * a_{rj}$;
  - $a_{ir} = 0$;
  - $b_i = b_i + f * b_r$;
- $r = r+1$;
- **pivotare(**r**);**

if $(|a_{rr}| \leq \varepsilon)$ **'MATRICE SINGULARA'**

else $\{ A \leftarrow A^{(n-1)}, \; b \leftarrow b^{(n-1)}$

se rezolvă sistemul triunghiular superior $Ax = b\}$

Numărul de operații efectuate la pasul r și în total este:

$$(n-r)\left[1M+(n-r)A+(n-r)M+1A+1M\right] \Rightarrow$$

$$\mathbf{M:}\quad \sum_{r=1}^{n-1}(n-r)^2+2\sum_{r=1}^{n-1}(n-r)=\frac{(n-1)n(2n+5)}{6},$$

$$\mathbf{A:}\quad \sum_{r=1}^{n-1}(n-r)^2+\sum_{r=1}^{n-1}(n-r)=\frac{(n-1)n(n+1)}{3},$$

$$\mathbf{M:}\ \frac{n^3}{3}+\mathcal{O}(n^2) \qquad ; \qquad \mathbf{A:}\ \frac{n^3}{3}+\mathcal{O}(n^2)$$

Eliminarea *„chinezească”*

200-100 î.Cr. China – *9 capitole despre arta matematică* – metodă de rezolvare foarte asemănătoare eliminării Gauss

„Avem 3 tipuri de grâu. Știm că 3 baloturi din primul tip, 2 baloturi din al doilea tip și 1 balot din al treilea tip cântăresc 39 măsuri. De asemenea, 2 baloturi din primul tip, 3 baloturi din al doilea tip și 1 balot din al treilea tip cântăresc 34 măsuri și 1 balot din primul tip, 2 baloturi din al doilea tip și 3 baloturi din al treilea tip cântăresc 26 măsuri. Câte măsuri cântărește un balot din fiecare tip de grâu”

Notația actuală:

$$3b_1 + 2b_2 + b_3 = 39$$
$$2b_1 + 3b_2 + b_3 = 34$$
$$b_1 + 2b_2 + 3b_3 = 26$$

Notația *chinezească*

1	**2**	**3**
2	**3**	**2**
3	**1**	**1**
26	**34**	**39**

Pasul 1

Se înmulțește coloana a doua cu 3 și se scade din ea coloana a treia atât timp cât este posibil.

Se înmulțește prima coloană cu 3 și se scade din ea coloana a treia atât timp cât este posibil.

Se ajunge la forma:

0	**0**	**3**
4	**5**	**2**
8	**1**	**1**
39	**24**	**39**

Pasul 2

Se înmulțește prima coloană cu 5 și se scade din ea coloana a doua atât timp cât este posibil.

Se ajunge la forma:

0	**0**	**3**
0	**5**	**2**
36	**1**	**1**
99	**24**	**39**

Pentru rezolvare se folosește metoda substituției inverse pe sistemul obținut mai sus.

Descompuneri LU

$$A \in \mathbb{R}^{n \times n}, \quad A = L \cdot U,$$

L inferior triunghiulară şi U superior triunghiulară

$$L, U \in \mathbb{R}^{n \times n}$$

$$Ax = b \leftrightarrow LUx = b \leftrightarrow \begin{cases} Ly = b \Rightarrow \text{soluţia } y^* \\ Ux = y^* \Rightarrow \text{soluţia } x^* \end{cases}, \quad x^* = A^{-1}b$$

Fie minorul principal principal al matricei $\boldsymbol{A}$:

$$\boldsymbol{A_p = \det \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & & & \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{bmatrix} \in \mathbb{R}^{p \times p} \quad , \quad p = 1, \ldots, n}$$

Teoremă (descompunere *LU*)

Fie $\boldsymbol{A \in \mathbb{R}^{n \times n}}$ o matrice reală pătratică de dimensiune $\boldsymbol{n}$ astfel încât $\boldsymbol{\det A_p \neq 0}$, $\boldsymbol{\forall p = 1,\ldots,n}$. Atunci există o unică matrice inferior triunghiulară $\boldsymbol{L = (l_{ij})_{i,j=1,\ldots,n}}$ şi o unică matrice superior triunghiulară $\boldsymbol{U = (u_{ij})_{i,j=1,\ldots,n}}$ cu $\boldsymbol{u_{ii} = 1, i = 1,\ldots,n}$ astfel încât

$$\boldsymbol{A = L \cdot U} \qquad (1)$$

Demonstraţie. ***Existenţa***: demonstraţia se face prin inducţie după $\boldsymbol{n}$ dimensiunea matricii $\boldsymbol{A}$.

Algoritmul Crout *de calcul al descompunerii* ***LU***

Fie $\boldsymbol{A \in \mathbb{R}^{n \times n}}$ o matrice reală pătratică de dimensiune $\boldsymbol{n}$ care satisface ipotezele teoremei de mai sus. Algoritmul de calcul al matricelor $\boldsymbol{L}$ şi $\boldsymbol{U}$ are $\boldsymbol{n}$ etape. La fiecare pas se determină simultan:

- câte o coloană din matricea $\boldsymbol{L}$ şi
- câte o linie din matricea $\boldsymbol{U}$.

Descriem în continuare, un pas oarecare.

Pasul p $(p = 1, 2, \ldots, n)$

Se determină elementele coloanei p ale matricii L:

$$l_{ip},\ i = p, \ldots, n$$

şi elementele liniei p ale matricii U ,

$$u_{pp} = 1\ ,\ \ u_{pi}, i = p + 1, \ldots, n,$$

$$(u_{pi} = l_{ip} = 0,\ i = 1, \ldots, p-1).$$

$$\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \boldsymbol{l}_{pp} \\ \boldsymbol{l}_{p+1\,p} \\ \vdots \\ \boldsymbol{l}_{np} \end{pmatrix}$$ col. $\boldsymbol{p}$ a matr. $\boldsymbol{L}$

$$\begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} & \boldsymbol{u}_{pp+1} & \cdots & \boldsymbol{u}_{pn} \end{pmatrix}$$ lin. $\boldsymbol{p}$ a matr. $\boldsymbol{U}$

Se cunosc de la paşii anteriori:

- elementele primelor ***p-1*** coloane din ***L*** (elemente l_{ik} cu $k = 1,\ldots, p-1, \forall j$).

- elementele primelor ***p-1*** linii din ***U*** (elemente u_{kj} cu $k = 1,\ldots, p-1, \forall j$)

Calculul elementelor ***coloanei p*** din matricea *L*, l_{ip} $i = p,\ldots,n$ se face folosind elementul a_{ip} şi $(LU)_{ip}$. Avem:

$$a_{ip} = (LU)_{ip} = \sum_{k=1}^{n} l_{ik} u_{kp} \; (u_{kp} = 0, k = p+1, \ldots, n) = \sum_{k=1}^{p} l_{ik} u_{kp} =$$

$$= \sum_{k=1}^{p-1} l_{ik} u_{kp} + l_{ip} u_{pp} = \sum_{k=1}^{p-1} l_{ik} u_{kp} + l_{ip} \quad (u_{pp} = 1)$$

Pentru $i = p, \ldots, n$ avem:

$$l_{ip} = a_{ip} - \sum_{k=1}^{p-1} l_{ik} u_{kp}, \quad i = p, \ldots, n \qquad (2)$$

($u_{pp} = 1$, l_{ik}, u_{kp} $k = 1, \ldots, p-1$ sunt elemente de pe coloane din L şi linii din U calculate la paşii anteriori)

Calculul elementelor ***liniei p*** din matricea ***U:***

$$u_{pi}\ , i = p+1,\ldots,n\ (u_{pi} = 0\ , i = 1,\ldots, p-1\ , u_{pp} = 1)$$

se face analog:

$$a_{pi} = (LU)_{pi} = \sum_{k=1}^{n} l_{pk} u_{ki} (l_{pk} = 0, k = p+1,\ldots,n) = \sum_{k=1}^{p} l_{pk} u_{ki} =$$

$$= \sum_{k=1}^{p-1} l_{pk} u_{ki} + l_{pp} u_{pi}$$

Dacă $l_{pp} \neq 0$ putem calcula elementele nenule ale coloanei ***p*** din matricea ***U*** astfel:

$$u_{pi} = \frac{(a_{pi} - \sum_{k=1}^{p-1} l_{pk} u_{ki})}{l_{pp}} \quad , \quad i = p+1, \cdots, n \qquad (3)$$

(elementele l_{pk}, u_{ki} $k = 1, \ldots, p-1$ sunt calculate anterior pasului p)

Dac $l_{pp} = 0$, calculele se opresc, descompunerea LU nu poate fi calculată - matricea A are un minor A_p cu determinantul 0.

Unicitatea: Demonstraţie prin reducere la absurd.
Facem observaţia că inversa unei matrici nesingulare triunghiulară inferior (superior) este o matrice de acelaşi tip.

Presupunem că

$$A = L \cdot U = L_1 \cdot U_1 \qquad (4)$$

Din ipoteza A nesingulară rezultă existenţa inverselor matricelor L, L_1, U, U_1. Înmulţind egalitatea (4) la stânga cu L^{-1}şi cu U_1^{-1} la dreapta obţinem

$$U U_1^{-1} = L^{-1} L_1.$$

Matricea $U U_1^{-1}$ este superior triunghiulară cu elementele diagonale egale cu 1 iar matricea $L^{-1} L_1$ este inferior triunghiulară. Rezultă că:

$$U U_1^{-1} = L^{-1} L_1 = I_n, \qquad \text{deci} \qquad L=L_1, \quad U=U_1.$$

Numărul de operații efectuate:

A (adunări, scăderi):

$$\sum_{p=1}^{n}\left[(n-p+1)(p-1)+(n-p)(p-1)\right]=\frac{n(n-1)(2n-1)}{6}=\frac{1}{3}n^3+O(n^2)$$

M (înmulțiri, împărțiri):

$$\sum_{p=1}^{n}\left[(n-p+1)(p-1)+(n-p)p\right]=\frac{(n-1)n(n+1)}{3}=\frac{1}{3}n^3+O(n^2)$$

Descompunerea Cholesky

O matrice $\boldsymbol{A \in \mathbb{R}^{n \times n}}$ se numeşte ***pozitiv definită*** dacă:

$$\left(\boldsymbol{Ax, x}\right)_{\mathbb{R}^n} > \boldsymbol{0} \quad \forall\, \boldsymbol{x \in \mathbb{R}^n}\ ,\ \boldsymbol{x \neq 0}$$

Notaţie: $\boldsymbol{A > 0}$

Fie $\boldsymbol{A \in \mathbb{R}^{n \times n}}$ o matrice simetrică ($\boldsymbol{A = A^T}$) şi pozitiv definită.

Descompunerea Cholesky pentru matricea $\boldsymbol{A}$ este de forma:

$\boldsymbol{A = LL^T}$, $\boldsymbol{L}$ matrice inferior triunghiulară

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} = LL^T = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & & & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ 0 & l_{22} & \cdots & l_{n2} \\ \vdots & & & \\ 0 & 0 & \cdots & l_{nn} \end{pmatrix}$$

Matricea L se calculează în n paşi, coloană după coloană.

Pas *r* *(r=1,…,n)*

Se calculează elementele coloanei ***r*** a matricii ***L***:
întâi elementul diagonal l_{rr} apoi
celelalte elemente l_{ir} $(i=r+1,\ldots n)$

Coloana ***r*** a matricii ***L:***

$$\begin{pmatrix} 0 & \cdots & 0 & l_{rr} & l_{r+1r} & \cdots & l_{ir} & \cdots & l_{nr} \end{pmatrix}^T$$

- se cunosc elementele primelor ***(r-1)*** coloane ale matricii ***L***

Calcul l_{rr}:

$$a_{rr} = \left(LL^T \right)_{rr} = \begin{pmatrix} l_{r1} & \cdots & l_{rr-1} & l_{rr-1} & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} l_{r1} \\ \vdots \\ l_{rr-1} \\ l_{rr} \\ 0 \\ \vdots \\ 0 \end{pmatrix} =$$

$$= l_{r1}^2 + l_{r2}^2 + \cdots + l_{rr-1}^2 + l_{rr}^2 \quad \Rightarrow \quad l_{rr} = \pm \sqrt{a_{rr} - \sum_{k=1}^{r-1} l_{rk}^2}$$

Calcul l_{ir} $(i=r+1,\ldots,n)$:

$$a_{ir} = \left(LL^T \right)_{ir} = \begin{pmatrix} l_{i1} & \cdots & l_{ir-1} & l_{ir} & \cdots & l_{ii} & \cdots & 0 \end{pmatrix} \begin{pmatrix} l_{r1} \\ \vdots \\ l_{rr-1} \\ l_{rr} \\ 0 \\ \vdots \\ 0 \end{pmatrix} =$$

$$= l_{i1}l_{r1} + l_{i2}l_{r2} + \cdots + l_{ir-1}l_{rr-1} + l_{ir}l_{rr} \quad \Rightarrow \quad l_{ir} = \frac{\left(a_{ir} - \sum_{k=1}^{r-1} l_{ik}l_{rk} \right)}{l_{rr}}$$

Calcul Numeric

Cursul 5

2022

Anca Ignat

Algoritmul de eliminare Gauss fără schimbare de ecuații $\Leftrightarrow$ descompunere LU

Presupunem că la fiecare pas al algoritmului de eliminare Gauss pivotul este nenul ($\boldsymbol{a_{rr}^{(r-1)} \neq 0}$), deci nu e nevoie de schimbare de ecuații.

Algoritmul se poate scrie astfel:

$$\textbf{for } r = 1, \ldots, n-1$$

$$\textbf{for } i = r+1, \ldots, n$$

- $f = -\dfrac{a_{ir}}{a_{rr}};$

$// \; E_i = E_i + f * E_r$

- $\textbf{for } j = r+1, \ldots, n$

$$a_{ij} = a_{ij} + f * a_{rj};$$

- $a_{ir} = 0;$
- $b_i = b_i + f * b_r;$

Considerăm vectorul și matricea:

$$t^{(r)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ t_{r+1}^{(r)} \\ \vdots \\ t_n^{(r)} \end{pmatrix} \in \mathbb{R}^n \quad , \quad T_r := I_n + t^{(r)} e_r^T \in \mathbb{R}^{n \times n}$$

$$t^{(r)}e_r^T = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ t_{r+1}^{(r)} \\ \vdots \\ t_n^{(r)} \end{pmatrix} \begin{pmatrix} 0 & \cdots & 1 & 0 \cdots 0 \end{pmatrix} = \begin{pmatrix} & & & \overset{\text{col r}}{\overbrace{}} & \\ 0 & 0 & \cdots & 0 & \cdots 0 \\ \vdots & & & & \\ 0 & 0 & \cdots & 0 & \cdots 0 \\ 0 & 0 & \cdots & t_{r+1}^{(r)} & \cdots 0 - \text{lin}\,(r+1) \\ \vdots & & & & \\ 0 & 0 & \cdots & t_n^{(r)} & \cdots 0 \end{pmatrix}$$

Matricea $\boldsymbol{T_r}$ este matrice triunghiulară inferior cu ***1*** pe diagonala principală:

$$\begin{matrix} & & & & \textbf{col } \boldsymbol{r} & \\ \boldsymbol{T_r} = & \begin{pmatrix} \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & & & & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{t}_{\boldsymbol{r+1}}^{(\boldsymbol{r})} & \cdots & \mathbf{0} \\ \vdots & & & & & \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{t}_{\boldsymbol{n}}^{(\boldsymbol{r})} & \cdots & \mathbf{1} \end{pmatrix} \end{matrix}$$

Inversa matricei $\boldsymbol{T_r}$ este

$$\boldsymbol{T_r^{-1} = I_n - t^{(r)} e_r^T}\ .$$

$$\boldsymbol{T_r\ T_r^{-1} = (I_n + t^{(r)} e_r^T)(I_n - t^{(r)} e_r^T) = I_n - t^{(r)} e_r^T + t^{(r)} e_r^T - t^{(r)} e_r^T t^{(r)} e_r^T =}$$
$$\boldsymbol{= I_n - t^{(r)}\ t_r^{(r)} e_r^T = I_n \quad (0 = t_r^{(r)} = e_r^T t^{(r)})}$$

Dacă $\boldsymbol{A}$ este o matrice oarecare, vrem să vedem cum se poate construi matricea $\boldsymbol{B=T_r\ A}$ fără a face înmulțire matricială. Vom studia legătura între liniile matricelor $\boldsymbol{A}$ și $\boldsymbol{B}$.

$$e_i^T B = e_i^T (T_r A) = e_i^T (I_n + t^{(r)} e_r^T) A = e_i^T A + e_i^T t^{(r)} e_r^T A =$$
$$= e_i^T A + t_i^{(r)} (e_r^T A)$$

Linia i a noii matrice B se obține din linia i a matricei A la care se adaugă linia r a matricei A înmulțită cu factorul $t_i^{(r)}$.

$$e_i^T B = \begin{cases} e_i^T A & i = 1, \ldots, r \ (t_i^{(r)} = 0) \\ e_i^T A + t_i^{(r)} (e_r^T A) & i = r+1, \ldots, n \end{cases} .$$

Operația T_rA descrie Pasul r al algoritmului de eliminare Gauss dacă:

$$t_{r+1}^{(r)} = -\frac{a_{r+1r}^{(r)}}{a_{rr}^{(r)}}, \cdots, t_i^{(r)} = -\frac{a_{ir}^{(r)}}{a_{rr}^{(r)}}, \cdots, t_n^{(r)} = -\frac{a_{nr}^{(r)}}{a_{rr}^{(r)}}.$$

Algoritmul de eliminare Gauss fără schimbare de ecuații poate fi descris astfel:

$$T_{n-1} \cdots T_2 T_1 A = U \quad \text{cu} \quad T_r = I_n + t^{(r)} e_r^T,$$

$$t^{(r)} = \left(0 \cdots 0 \ (-\frac{a_{r+1r}^{(r)}}{a_{rr}^{(r)}}) \cdots (-\frac{a_{ir}^{(r)}}{a_{rr}^{(r)}}) \cdots (-\frac{a_{nr}^{(r)}}{a_{rr}^{(r)}}) \right)^T.$$

Avem:

$$A = T_1^{-1}T_2^{-1}\cdots T_{n-1}^{-1}\,U = LU \;\;,\;\; L := T_1^{-1}T_2^{-1}\cdots T_{n-1}^{-1}$$

$$T_1^{-1}T_2^{-1} = (I_n - t^{(1)}e_1^T)(I_n - t^{(2)}e_2^T) = I_n - t^{(1)}e_1^T - t^{(2)}e_2^T + t^{(1)}e_1^T t^{(2)}e_2^T =$$

$$= I_n - t^{(1)}e_1^T - t^{(2)}e_2^T + t^{(1)}t_1^{(2)}e_2^T = I_n - t^{(1)}e_1^T - t^{(2)}e_2^T \;\; (t_1^{(2)} = 0)$$

Prin inducție se arată că:

$$L = T_1^{-1}T_2^{-1}\cdots T_{n-1}^{-1} = I_n - t^{(1)}e_1^T - t^{(2)}e_2^T - \cdots - t^{(n-1)}e_{n-1}^T$$

$$L=\begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ \frac{a_{21}}{a_{11}} & 1 & \cdots & 0 & \cdots & 0 \\ \frac{a_{31}}{a_{11}} & \frac{a_{32}^{(1)}}{a_{22}^{(1)}} & \cdots & 0 & \cdots & 0 \\ \vdots & & & & & \\ \frac{a_{r1}}{a_{11}} & \frac{a_{r2}^{(1)}}{a_{22}^{(1)}} & \cdots & 1 & \cdots & 0 \\ \frac{a_{r+11}}{a_{11}} & \frac{a_{r+12}^{(1)}}{a_{22}^{(1)}} & \cdots & \frac{a_{r+1r}^{(r-1)}}{a_{rr}^{(r-1)}} & \cdots & 0 \\ \vdots & & & & & \\ \frac{a_{n1}}{a_{11}} & \frac{a_{n2}^{(1)}}{a_{22}^{(1)}} & \cdots & \frac{a_{nr}^{(r-1)}}{a_{rr}^{(r-1)}} & \cdots & 1 \end{pmatrix}$$

Descompuneri QR

Definiție

Se numește ***matrice ortogonală***, o matrice $Q \in \mathbb{R}^{n \times n}$ care satisface relația:

$$Q^T Q = QQ^T = I_n \quad (Q^{-1} = Q^T) .$$

Matricele ortogonale au următoarele proprietăți:

- Dacă Q este matrice ortogonală atunci şi matricea transpusă Q^T este ortogonală.

$$Q^T Q = Q^T (Q^T)^T = QQ^T = (Q^T)^T Q^T = I_n$$

- Dacă Q_1 şi Q_2 sunt matrice ortogonale atunci Q_1Q_2 este tot matrice ortogonală.

$$(Q_1Q_2)^T(Q_1Q_2) = Q_2^T Q_1^T Q_1 Q_2 = Q_2^T I\, Q_2 = I_n$$
$$(Q_1Q_2)(Q_1Q_2)^T = Q_1 Q_2 Q_2^T Q_1^T = Q_1^T I\, Q_1 = I_n$$

- Dacă $Q \in \mathbb{R}^{n\times n}$ este matrice ortogonală şi $x \in \mathbb{R}^n$ atunci $\| Qx \|_2 = \| x \|_2$.

$$\| Qx \|_2^2 = (Qx, Qx) = \left(x, Q^T Qx \right) = (x, x) = \| x \|_2^2, \quad \| \cdot \|_2 \geq 0 \Rightarrow$$
$$\| Qx \|_2 = \| x \|_2$$

Fie $\boldsymbol{A}$ o matrice reală pătratică de dimensiune $\boldsymbol{n}$. Pp. că avem:

$$\boldsymbol{A = QR}$$

unde $\boldsymbol{Q}$ este o matrice ortogonală iar $\boldsymbol{R}$ este o matrice superior triunghiulară.

$$\boldsymbol{Ax = b \quad \Leftrightarrow \quad QRx = b \quad \Leftrightarrow \quad Q^T QRx = Q^T b \quad \Leftrightarrow \quad Rx = Q^T b}$$

Algoritmul lui Householder

Matrice de reflexie - $P \in \mathbb{R}^{n \times n}$ de forma:

$$P = I_n - 2vv^T \ , \quad v \in \mathbb{R}^n \ , \quad \| v \|_2 = \sqrt{\sum_{j=1}^{n} | v_j |^2} = 1$$

$$vv^T = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} (v_1, v_2, \ldots, v_n) = \begin{pmatrix} v_1^2 & v_1 v_2 & \ldots & v_1 v_n \\ v_2 v_1 & v_2^2 & \ldots & v_2 v_n \\ \vdots & & & \\ v_n v_1 & v_n v_2 & \ldots & v_n^2 \end{pmatrix}.$$

Matricele de reflexie sunt :

simetrice - $\boldsymbol{P = P^T}$ şi

ortogonale - $\boldsymbol{PP^T = P^T P = P^2 = I_n}$.

$$\boldsymbol{P^T = (I_n - 2vv^T)^T = I_n - 2(vv^T)^T = I_n - 2(v^T)^T v^T = I_n - 2vv^T = P}$$

$$P^2 = (I_n - 2vv^T)(I_n - 2vv^T) = I_n - 2vv^T - 2vv^T + 4(vv^T)(vv^T) =$$

$$= I_n - 4vv^T + 4v(v^T v)v^T = I_n - 4vv^T + 4v\|v\|_2^2 v^T =$$

$$= I_n - 4vv^T + 4vv^T = I_n \quad (\|v\|_2 = 1)$$

$$n = 2, \quad v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - 2\begin{pmatrix} 1 \\ 0 \end{pmatrix}(1 \quad 0) = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad x \in \mathbb{R}^2, \; y = Px$$

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -x_1 \\ x_2 \end{pmatrix}$$

Vectorul $y=Px$ este reflectatul vectorului x în raport cu axa Ox_2.

Algoritmul care folosește matricele de reflexie pentru a obține o descompunere $\boldsymbol{QR}$ pentru o matrice $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{n \times n}}$ a fost descris de Alston S. Householder în articolul "*Unitary triangularization of a nonsymmetric matrix*" apărut în Journal of the Assoc. of Computing Machinery 5 (1958), 339-342.

Transformarea matricei $\boldsymbol{A}$ într-una superior triunghiulară se face în $\boldsymbol{(n-1)}$ pași, la fiecare pas folosindu-se o matrice de reflexie.

Pas 1: $\boldsymbol{A^{(1)} = P_1 A}$ (matricea $\boldsymbol{P_1}$ se alege astfel încât col. ***1*** să fie transformată în formă superior triunghiulară)

Pas 2: $\boldsymbol{A^{(2)} = P_2 A^{(1)} = P_2(P_1 A)}$ ($\boldsymbol{P_2}$ transformă col. ***2*** în formă superior triunghiulară, fără să schimbe col. ***1***)

Pas *r*: $\boldsymbol{A^{(r)} = P_r A^{(r-1)} = P_r(P_{r-1} \ldots P_1 A)}$ (se transformă col ***r*** în formă superior triunghiular fără să schimbe primele ***(r-1)*** coloane)

Descompunerea $\boldsymbol{QR}$ construită cu algoritmul Householder este următoarea:

$$\boldsymbol{P_{n-1} \cdots P_r \cdots P_2 P_1 A = \tilde{Q} A = R}$$

unde

$$\boldsymbol{\tilde{Q} = P_{n-1} \cdots P_r \cdots P_2 P_1}$$

$\boldsymbol{\tilde{Q}}$ este matrice ortogonală ca produs de matrice ortogonale.

$$\boldsymbol{\tilde{Q}A = R \quad \Leftrightarrow \quad \tilde{Q}^T Q A = \tilde{Q}^T R \quad \Leftrightarrow \quad A = \tilde{Q}^T R = QR}$$

$$\boldsymbol{Q = \tilde{Q}^T = (P_{n-1} \cdots P_r \cdots P_2 P_1)^T = P_1 P_2 \cdots P_r \cdots P_{n-1}}$$

Pasul *r*

La intrarea în pasul ***r*** matricea ***A*** are forma:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1r} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2r} & \cdots & a_{2n} \\ \vdots & & & & & \\ 0 & 0 & \cdots & a_{rr} & \cdots & a_{rn} \\ 0 & 0 & \cdots & a_{r+1r} & \cdots & a_{r+1n} \\ \vdots & & & & & \\ 0 & 0 & \cdots & a_{ir} & \cdots & a_{in} \\ \vdots & & & & & \\ 0 & 0 & \cdots & a_{nr} & \cdots & a_{nn} \end{pmatrix}$$

Pasul $\boldsymbol{r}$ constă în:

$$\boldsymbol{A := P_r A}$$

$$\boldsymbol{P_r = I_n - 2v^r (v^r)^T \ , \quad v^r \in R^n \ , \quad \| v^r \|_2 = 1}$$

unde vectorul $\boldsymbol{v^r}$ se alege astfel ca matricea $\boldsymbol{A}$ să aibă şi coloana $\boldsymbol{r}$ în formă superior triunghiulară:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1r} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2r} & \cdots & a_{2n} \\ \vdots & & & & & \\ 0 & 0 & \cdots & a_{rr} & \cdots & a_{rn} \\ 0 & 0 & \cdots & 0 & \cdots & a_{r+1n} \\ \vdots & & & & & \\ 0 & 0 & \cdots & 0 & \cdots & a_{in} \\ \vdots & & & & & \\ 0 & 0 & \cdots & 0 & \cdots & a_{nn} \end{pmatrix}$$

Calculul matricei P_r

Pentru simplitate vom nota $P_r=P$, $v^r=v$.

$$Ae_r = \begin{pmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{r-1r} \\ a_{rr} \\ a_{r+1r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix} \rightarrow (PA)e_r = \bar{A}e_r = \begin{pmatrix} \bar{a}_{1r} = a_{1r} \\ \bar{a}_{2r} = a_{2r} \\ \vdots \\ \bar{a}_{r-1r} = a_{r-1r} \\ \bar{a}_{rr} = k \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Aplicând proprietatea matricelor ortogonale :

$$Q \in \mathbb{R}^{n\times n},\ x \in \mathbb{R}^{n},\ \| Qx \|_2 = \| x \|_2$$

pentru matricea $Q=P$ şi vectorul $x=Ae_r$ avem:

$$\| PAe_r \|_2^2 = a_{1r}^2 + a_{2r}^2 + \cdots + a_{r-1r}^2 + k^2 =$$

$$\| Ae_r \|^2 = a_{1r}^2 + a_{2r}^2 + \cdots + a_{r-1r}^2 + a_{rr}^2 + a_{r+1r}^2 + \cdots + a_{ir}^2 + \cdots + a_{nr}^2$$

Din relația de mai sus rezultă:

$$k^2 = \sigma = a_{rr}^2 + a_{r+1r}^2 + \cdots + a_{ir}^2 + \cdots + a_{nr}^2 = \sum_{i=r}^{n} a_{ir}^2 \Rightarrow k = \pm\sqrt{\sigma}$$

Determinarea vectorului $\boldsymbol{v}$ ce definește matricea $\boldsymbol{P:}$

$$(\boldsymbol{PA})\boldsymbol{e}_r = (\boldsymbol{I}_n - 2\boldsymbol{v}\boldsymbol{v}^T)(\boldsymbol{A}\boldsymbol{e}_r) = \boldsymbol{A}\boldsymbol{e}_r - 2(\boldsymbol{v}\boldsymbol{v}^T)(\boldsymbol{A}\boldsymbol{e}_r) =$$

$$= \boldsymbol{A}\boldsymbol{e}_r - 2\boldsymbol{v}(\boldsymbol{v}^T(\boldsymbol{A}\boldsymbol{e}_r)) = \boldsymbol{A}\boldsymbol{e}_r - (2\alpha)\,\boldsymbol{v} = \boldsymbol{A}\boldsymbol{e}_r - \boldsymbol{u}$$

unde cu α și $\boldsymbol{u}$ am notat:

$$\alpha := v^T(Ae_r) = \big((Ae_r), v\big)_{\mathbb{R}^n} = \left(\begin{pmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix}, \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_i \\ \vdots \\ v_n \end{pmatrix}\right)_{\mathbb{R}^n} =$$

$$= a_{1r}v_1 + a_{2r}v_2 + \cdots + a_{ir}v_i + \cdots + a_{nr}v_n$$

$$u := (2\alpha)\, v = Ae_r - (PA)e_r = \begin{pmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{r-1r} \\ a_{rr} \\ a_{r+1r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix} - \begin{pmatrix} a_{1r} \\ a_{2r} \\ \vdots \\ a_{r-1r} \\ k \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix}$$

Cu aceste notații, matricea $\boldsymbol{P}$ devine:

$$\boldsymbol{P} = \boldsymbol{I}_n - 2(\frac{1}{2\alpha})\boldsymbol{u}\,(\frac{1}{2\alpha})\,\boldsymbol{u}^T = \boldsymbol{I}_n - (\frac{1}{2\alpha^2})\boldsymbol{u}\boldsymbol{u}^T = \boldsymbol{I}_n - \frac{1}{\beta}\boldsymbol{u}\boldsymbol{u}^T$$

$$\beta := 2\alpha^2$$

Pentru a cunoaște matricea $\boldsymbol{P}$ trebuie să mai determinăm constanta β. Din condiția:

$$\|\boldsymbol{v}\|_2^2 = 1 \Rightarrow \|\frac{1}{2\alpha}\boldsymbol{u}\|_2^2 = 1 \Rightarrow \frac{1}{4\alpha^2}\|\boldsymbol{u}\|_2^2 = 1 \Rightarrow 2\beta = \|\boldsymbol{u}\|_2^2$$

$$\| u \|_2^2 = \| \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix} \|_2^2 = (a_{rr} - k)^2 + a_{r+1r}^2 + \cdots + a_{ir}^2 + \cdots + a_{nr}^2 =$$

$$= a_{rr}^2 + a_{r+1r}^2 + \cdots + a_{ir}^2 + \cdots + a_{nr}^2 - 2ka_{rr} + k^2 =$$

$$= \sigma - 2ka_{rr} + \sigma = 2(\sigma - ka_{rr})$$

de unde obținem:

$$\beta = \sigma - k a_{rr}$$

Vom alege semnul constantei k astfel încât β să fie cât mai mare posibil deoarece constanta β apare în operația de împărțire. Avem:

$$\beta \ \text{"mare"} \ \rightarrow \ \beta = \sigma - k a_{rr} \geq \sigma \quad (\sigma \geq 0) \quad \rightarrow$$

$$\text{semn}\, k = -\text{semn}\, a_{rr}$$

Ce înseamnă $\beta = 0$?

$$\beta = \frac{1}{2}\|u\|_2^2 = 0 \rightarrow \|u\|_2^2 = 0 \rightarrow u = 0 \rightarrow$$

$$\rightarrow a_{rr} = k, a_{r+1r} = 0, \ldots, a_{ir} = 0, \ldots, a_{nr} = 0$$

Cum $a_{rr}=k$ şi **semn** k = **-semn** a_{rr} obținem:

$$a_{ir} = 0, \quad \forall i = r, \ldots, n$$

adică avem coloana r deja în formă superior triunghiulară, se poate trece la pasul următor. În acest caz matricea A este singulară.

Ne interesează cum se efectuează operația $\boldsymbol{A=P_rA}$ fără a face înmulțire matricială. Vom pune în evidentă schimbările în raport cu coloanele.

$$(PA)e_j = \text{noua col. } j \text{ a matr. } A = (I_n - \frac{1}{\beta}uu^T)(Ae_j) =$$

$$= Ae_j - \frac{1}{\beta}(uu^T)(Ae_j) = Ae_j - \frac{1}{\beta}u(u^T(Ae_j)) =$$

$$= Ae_j - \frac{\gamma_j}{\beta}u$$

$$\gamma_j := u^T(Ae_j) = \left(\begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{rj} \\ a_{r+1j} \\ \vdots \\ a_{nj} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ a_{rr}-k \\ a_{r+1r} \\ \vdots \\ a_{nr} \end{pmatrix}\right)_{\mathbb{R}^n} =$$

$$= a_{rj}(a_{rr}-k)+\cdots+a_{ij}a_{ir}+\cdots+a_{nj}a_{nr} = \sum_{i=1}^{n} u_i a_{ij} = \sum_{i=r}^{n} u_i a_{ij}$$

$$(u_i = 0, i = 1,\ldots,r-1, u_r = a_{rr}-k, u_i = a_{ir}, i = r+1,\ldots,n)$$

Noua coloană ***j*** se obține din vechea coloană ***j*** din care scădem vectorul ***u*** înmulțit cu constanta γ_j. Ne interesează ca primelor ***(r-1)*** coloane să nu li se schimbe forma superior triunghiulară deja obținută.

Pentru $j=1,\ldots,(r-1)$ avem:

$$\gamma_j := u^T(Ae_j) = \left(\begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{jj} \\ a_{j+1j} = 0 \\ \vdots \\ a_{rj} = 0 \\ \vdots \\ a_{nj} = 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{nr} \end{pmatrix}\right)_{\mathbb{R}^n} =$$

$$= a_{1j}0 + \cdots + a_{jj}0 + \cdots + 0(a_{rr} - k) + \cdots + 0a_{ir} + \cdots + 0a_{nr} = 0$$

Din faptul că $\gamma_j = 0,\ j = 1,...,r-1$ rezultă că primele ***(r-1)*** coloane ale matricei $\boldsymbol{A}$ nu se schimbă când facem operația $\boldsymbol{A=P_rA}$, rămân în formă superior triunghiulară.

Algoritmul de trecere de la matricea $\boldsymbol{A}$ la matricea $\boldsymbol{P_rA}$ este următorul:

$$(P_r A)e_j = \begin{cases} Ae_j & \text{pentru } j = 1,\ldots,r-1 \\ (a_{1r}, a_{2r}, \ldots, a_{r-1r}, k, 0, \ldots, 0)^T & \text{pentru } j = r \\ Ae_j - \dfrac{\gamma_j}{\beta} u & \text{pentru } j = r+1,\ldots,n \end{cases}$$

$$\gamma_j = (Ae_j, u)_{\mathbb{R}^n} = \sum_{i=r}^{n} u_i a_{ij}$$

$$u_i = 0, \quad i = 1,\ldots,r-1, \quad u_r = a_{rr} - k, \quad u_i = a_{ir}, \quad i = r+1,\ldots,n$$

Operaţia de transformare a vectorului termenilor liberi

$b := P_r b$:

$$P_r b = (I_n - \frac{1}{\beta}(uu^T))b = b - \frac{1}{\beta}(uu^T)b = b - \frac{1}{\beta}u(u^T b) = b - \frac{\gamma}{\beta}u$$

$$\gamma = u^T b = (b,u)_{\mathbb{R}^n} = \sum_{i=r}^{n} u_i b_i$$

Algoritmul lui Householder

$\widetilde{Q} = I_n$;

for $r = 1, \cdots, n-1$

- **calculează matricea** P_r **(constanta** β **şi vectorul** u**)**
- $A = P_r * A$;
- $b = P_r * b$;
- $\widetilde{Q} = P_r * \widetilde{Q}$;

La sfârșitul acestui algoritm, în matricea A vom avea matricea superior triunghiulară R, în vectorul b vom avea $Q^T b^{\text{init}}$, b^{init} este vectorul inițial al termenilor liberi, iar matricea $\widetilde{Q}$ va conține matricea Q^T din factorizarea QR a matricei A.

Algoritmul Householder detaliat

$\widetilde{Q} = I_n$;

for $r = 1, \ldots, n-1$

// **construcţia matricii** P_r **– constanta** β **şi vectorul** u

- $\sigma = \sum_{i=r}^{n} a_{ir}^2$;
- **if** $(\sigma \le \varepsilon)$ **break** ; // $r = r+1 \leftrightarrow P_r = I_n$ (A **singulară**)
- $k = \sqrt{\sigma}$;
- **if** $(a_{rr} > 0)$ $k = -k$;
- $\beta = \sigma - k\, a_{rr}$;
- $u_r = a_{rr} - k$; $u_i = a_{ir}$, $i = r+1, \ldots, n$;

// $A = P_r * A$

// transformarea coloanelor $j = r+1, \ldots, n$

- for $j = r+1, \ldots, n$

  $* \gamma = (\gamma_j / \beta) = (Ae_j, u) / \beta = (\sum_{i=r}^{n} u_i a_{ij}) / \beta;$

  * for $i = r, \ldots, n$

    $a_{ij} = a_{ij} - \gamma * u_i;$

// transformarea coloanei r a matricii A

- $a_{rr} = k;\ a_{ir} = 0,\ \ i = r+1, \ldots, n;$

// $b = P_r * b$

- $\gamma = (\gamma / \beta) = (b, u) / \beta = (\sum_{i=r}^{n} u_i b_i) / \beta;$
- for $i = r, \ldots, n \quad b_i = b_i - \gamma * u_i;$

$/ / \widetilde{Q} = P_r * \widetilde{Q}$

- **for** $j = 1, \ldots, n$

  $* \gamma = (\widetilde{Q} e_j, u) / \beta = (\sum_{i=r}^{n} u_i \tilde{q}_{ij}) / \beta;$

  $*$ **for** $i = r, \ldots, n$

  $\tilde{q}_{ij} = \tilde{q}_{ij} - \gamma * u_i;$

Numărul de operații efectuate:

A (adunări, scăderi):

$$(n-1)+2n(n-1)+\frac{n(n-1)(2n-1)}{3}=\frac{(n-1)(n+1)(2n+3)}{3}=$$

$$=\frac{2}{3}n^3+O(n^2)$$

M (înmulțiri, împărțiri):

$$4(n-1)+3n(n-1)+\frac{n(n-1)(2n-1)}{6}=\frac{2}{3}n^3+O(n^2)$$

R (radicali): ***(n-1)***

Algoritmul lui Givens

Fie $\boldsymbol{A}$ o matrice reală pătratică de dimensiune $\boldsymbol{n}$. Pp. că avem:

$$\boldsymbol{A = QR}$$

unde $\boldsymbol{Q}$ este o matrice ortogonală iar $\boldsymbol{R}$ este o matrice superior triunghiulară.

$$\boldsymbol{Ax = b \quad \Leftrightarrow \quad QRx = b \Leftrightarrow Q^T QRx = Q^T b \quad \Leftrightarrow Rx = Q^T b}$$

În cazul algoritmului Givens, pentru a aduce sistemul $\boldsymbol{Ax=b}$ la forma $\boldsymbol{Rx = Q^T b}$ se folosesc matricele de rotație. O matrice de rotație $\boldsymbol{R_{pq}(\theta) = (r_{ij})_{i,j=1,n}}$ are următoarea formă :

$$
R_{pq}(\theta) = \begin{array}{c} \\ \\ \\ \\ p \\ \\ q \\ \\ \\ \end{array}
\begin{array}{c}
\begin{array}{ccccccccccc} & & & & p & & q & & & & \end{array} \\
\begin{pmatrix}
1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & & & & & & & \\
0 & 0 & \cdots & c & \cdots & s & \cdots & 0 \\
\vdots & & & & 1 & & & \\
0 & 0 & \cdots & (-s) & \cdots & c & \cdots & 0 \\
\vdots & & & & & & & \\
0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1
\end{pmatrix}
\end{array}
$$

$$r_{ij} = \begin{cases} 1 & \text{pentru } i = j, \quad i \neq p, i \neq q \\ c & \text{pentru } i = j, \quad i = p, i = q \\ s & \text{pentru } i = p, j = q \\ -s & \text{pentru } i = q, j = p \\ 0 & \text{în rest} \end{cases}$$

unde $p, q \in \{1,...,n\}$ iar c şi s sunt două numere reale care satisfac relația $c^2+s^2=1$. Constantele c şi s pot fi alese astfel încât $c=\cos\theta$, $s=\sin\theta$. Se arată ușor, folosind relația $c^2+s^2=1$, că matricea $R_{pq}(\theta)$ este ortogonală:

$$R_{pq}(\theta)\, R_{pq}^{T}(\theta) = R_{pq}^{T}(\theta) R_{pq}(\theta) = I_n$$

Calculul matricei:

$$\boldsymbol{B = R_{pq}(\theta)A},$$

$\boldsymbol{B}$ se obține din $\boldsymbol{A}$ modificând doar liniile $\boldsymbol{p}$ şi $\boldsymbol{q}$.

Fie

$\boldsymbol{A_i = e_i^T A}$, - linia $\boldsymbol{i}$ a matricei $\boldsymbol{A}$

$\boldsymbol{B_i = e_i^T B}$ - linia $\boldsymbol{i}$ a matricei $\boldsymbol{B}$.

Liniile matricei $\boldsymbol{B}$:

$$\boldsymbol{B_i = A_i \ , \quad i = 1,\ldots,n \ , \quad i \neq p \ , i \neq q}$$

$$\boldsymbol{B_p = cA_p + sA_q}$$

$$\boldsymbol{B_q = -sA_p + cA_q}$$

$$b_{pj} = c\, a_{pj} + s\, a_{qj} \;\; , \;\; j = 1,\ldots,n$$

$$b_{qj} = -s\, a_{pj} + c\, a_{qj} \;\; , \; j = 1,\ldots,n$$

$$b_{ij} = a_{ij} \quad \text{în rest}$$

Calculul matricei :

$$D = A\, R_{pq}^{T}(\theta),$$

D se obține din A modificând doar coloanele p și q.

Notăm Ae_j , De_j – coloana j a matricei A şi respectiv D.

Coloanele matricei $\boldsymbol{D}$:

$$\boldsymbol{D_j = A_j \ , \quad j = 1,\ldots,n \, , \quad j \neq p \, , \, j \neq q}$$

$$\boldsymbol{De_p = cAe_p + sAe_q}$$

$$\boldsymbol{De_q = -sAe_p + cAe_q}$$

$$\boldsymbol{d_{ip} = c\, a_{ip} + s\, a_{iq} \ , \ i = 1,\ldots,n}$$

$$\boldsymbol{d_{iq} = -\, s\, a_{ip} + c\, a_{iq} \ , i = 1,\ldots,n}$$

$$\boldsymbol{d_{ij} = a_{ij}} \quad \text{în rest}$$

Algoritmul lui Givens se desfășoară în ***(n-1)*** pași - la pasul ***r*** se transformă coloana ***r*** a matricei ***A*** în formă superior triunghiulară fără a modifica primele ***(r-1)*** coloane.

Pasul 1

Intrare: matricea $\boldsymbol{A}$, vectorul $\boldsymbol{b}$

Ieșire: matricea $\boldsymbol{A^{(1)}}$ (cu prima coloană în formă superior triunghiulară), $\boldsymbol{b^{(1)}}$

Se efectuează următoarele operații de înmulțire cu matrice de rotație:

$$\boldsymbol{R_{1n}(\theta_{1n}) \cdots R_{13}(\theta_{13}) R_{12}(\theta_{12})\, A = A^{(1)}}$$

$$\boldsymbol{R_{1n}(\theta_{1n}) \cdots R_{13}(\theta_{13}) R_{12}(\theta_{12})\, b = b^{(1)}}$$

Unghiurile $\boldsymbol{\theta_{1i}}$ (constantele $\boldsymbol{c_{1i}}$ și $\boldsymbol{s_{1i}}$) se aleg astfel ca elementul de pe poziția $\boldsymbol{(i,1)}$ din matricea rezultat să devină $\boldsymbol{0}$.

Pasul r

Intrare: matricea $\boldsymbol{A^{(r-1)}}$ (are primele $\boldsymbol{(r-1)}$ coloane în formă superior triunghiulară), $\boldsymbol{b^{(r-1)}}$

Ieșire: matricea $\boldsymbol{A^{(r)}}$ (cu primele $\boldsymbol{r}$ coloane în formă superior triunghiulară), $\boldsymbol{b^{(r)}}$

La acest pas matricea $\boldsymbol{A^{(r-1)}}$ şi vectorul $\boldsymbol{b^{(r-1)}}$ se transformă astfel:

$$\boldsymbol{R_{rn}(\theta_{rn}) \cdots R_{ri}(\theta_{ri}) \cdots R_{rr+1}(\theta_{rr+1})\, A^{(r-1)} = A^{(r)}}$$

$$\boldsymbol{R_{rn}(\theta_{rn}) \cdots R_{ri}(\theta_{ri}) \cdots R_{rr+1}(\theta_{rr+1})\, b^{(r-1)} = b^{(r)}}$$

unde elementele $c = c_{ri}$ şi $s = s_{ri}$ din matricele de rotație se aleg astfel ca după înmulțirea cu $R_{ri}(\theta_{ri}), i = r+1,\ldots,n$ elementul ***(i,r)*** să devină ***0***.

Considerăm operația $B = R_{ri}(\theta_{ri})A$, unde θ_{ri} se alege astfel ca $b_{ir} = 0$:

$$b_{rj} = c\, a_{rj} + s\, a_{ij} \ ,$$

$$b_{ij} = -s\, a_{rj} + c\, a_{ij} \ , j = 1,\ldots,n$$

$$(b_{kl} = a_{kl} \quad \text{în rest})$$

$$(j = r) \quad b_{ir} = -s\, a_{rr} + c\, a_{ir}$$

Cea mai simplă alegere pentru c și s astfel ca să obținem $b_{ir}=0$ este:

$$c = f\, a_{rr} \quad , \quad s = f\, a_{ir} \quad ,$$

$$f \text{ ales astfel ca } c^2 + s^2 = 1 \quad \Rightarrow \quad f = \frac{1}{\sqrt{a_{rr}^2 + a_{ir}^2}}$$

$$c = \frac{a_{rr}}{\sqrt{a_{rr}^2 + a_{ir}^2}} \quad , \quad s = \frac{a_{ir}}{\sqrt{a_{rr}^2 + a_{ir}^2}}$$

$$\sqrt{a_{rr}^2 + a_{ir}^2} = 0 \;\Rightarrow a_{rr} = a_{ir} = 0 \;\Rightarrow\; c = 1\,,\; s = 0$$

$$(R_{ir}(\theta) = I_n)$$

Deci elementul de pe poziția ***(i,r)*** este deja nul. Nu avem ce schimba în matricea ***A*.**

Operația $\boldsymbol{B = R_{ri}(\theta_{ri})A}$, nu afectează forma superior triunghiulară a primelor ***(r-1)*** coloane. În matricea ***B*** aceste coloane vor continua să fie în formă superior triunghiulară.

$$\boldsymbol{b_{rj} = c\,a_{rj} + s\,a_{ij} = 0}\,, \quad \boldsymbol{b_{ij} = -s\,a_{rj} + c\,a_{ij} = 0}\ \boldsymbol{, j = 1,\ldots,r-1}$$

deoarece $\boldsymbol{a_{rj} = a_{ij} = 0}$

Înmulțirea $\boldsymbol{B = R_{ri}(\theta_{ri})A}$ nu schimbă decât liniile ***r*** și ***i*** ale matricei ***B***. În concluzie, operația $\boldsymbol{B = R_{ri}(\theta_{ri})A}$ nu schimbă elementele nule deja obținute, ci doar face ca elementul de pe poziția ***(i,r)*** să devină ***0***.

Algoritmul lui Givens poate fi descris astfel:

$$\boldsymbol{R_{n-1n}(\theta_{n-1n})\cdots R_{rn}(\theta_{rn})\cdots R_{r\,r+1}(\theta_{r\,r+1})\cdots R_{1n}(\theta_{1n})\cdots R_{12}(\theta_{12})\,A = R}$$

$$\boldsymbol{R_{n-1n}(\theta_{n-1n})\cdots R_{rn}(\theta_{rn})\cdots R_{r\,r+1}(\theta_{r\,r+1})\cdots R_{1n}(\theta_{1n})\cdots R_{12}(\theta_{12})b = \bar{b} = Q^T b}$$

Notăm cu $\widetilde{Q}$ următoarea matrice:

$$\boldsymbol{\widetilde{Q} = R_{n-1n}(\theta_{n-1n})\cdots R_{rn}(\theta_{rn})\cdots R_{r\,r+1}(\theta_{r\,r+1})\cdots R_{1n}(\theta_{1n})\cdots R_{12}(\theta_{12})}$$

Matricea $\widetilde{Q}$ este matrice ortogonală ca produs de matrice ortogonale. Descompunerea QR a matricei A este următoarea:

$$\widetilde{Q}\,A = R \quad (\widetilde{Q}^{-1} = \widetilde{Q}^{T}) \Rightarrow A = \widetilde{Q}^{T} R = QR$$

$$Q = \widetilde{Q}^{T} = \left(R_{n-1n}(\theta_{n-1n}) \cdots R_{rn}(\theta_{rn}) \cdots R_{r\,r+1}(\theta_{r\,r+1}) \cdots R_{1n}(\theta_{1n}) \cdots R_{12}(\theta_{12}) \right)^{T}$$

$$= R_{12}^{T}(\theta_{12}) \cdots R_{1n}^{T}(\theta_{1n}) \cdots R_{r\,r+1}^{T}(\theta_{r\,r+1}) \cdots R_{rn}^{T}(\theta_{rn}) \cdots R_{n-1n}^{T}(\theta_{n-1n})$$

Pe scurt, ***algoritmul lui Givens*** este următorul:

$$\widetilde{Q} = I_n;$$

$$\textbf{for } r = 1, \cdots, n-1$$

$$\quad \textbf{for } i = r+1, \cdots, n$$

$$\quad \begin{cases} A = R_{ri}(\theta_{ri}) * A; \\ b = R_{ri}(\theta_{ri}) * b; \\ \widetilde{Q} = R_{ri}(\theta_{ri}) * \widetilde{Q}; \end{cases}$$

$\widetilde{Q} = I_n;$

for $r = 1,\ldots,n-1$

  for $i = r+1,\ldots,n$

    // **construcţia matricii** $R_{ri}(\theta_{ri})$ **– constantele** c **şi** s

    - $f = \sqrt{a_{rr}^2 + a_{ir}^2};$
    - **if** $(f \le \varepsilon) \{ c = 1; s = 0; \}$ // $R_{ri}(\theta_{ri}) = I$

      else $\{ c = a_{rr} / f ; s = a_{ir} / f ; \}$

    // $A = R_{ri}(\theta_{ri}) * A$

    - **for** $j = r+1,\ldots,n$

$$\begin{cases} a_{rj} = c * a_{rj} + s * a_{ij}; \\ a_{ij} = -s * a_{rj}^{vechi} + c * a_{ij}; \end{cases}$$

- $a_{ir} = 0;\ a_{rr} = f\ ;$

$//\ b = R_{ri}(\theta_{ri}) * b$

- $b_r = c * b_r + s * b_i;$
- $b_i = -s * b_r^{vechi} + c * b_i;$

$//\ \tilde{Q} = R_{ri}(\theta_{ri}) * \tilde{Q}$

- for $j = 1, \ldots, n$

$$\begin{cases} \tilde{q}_{rj} = c * \tilde{q}_{rj} + s * \tilde{q}_{ij}; \\ \tilde{q}_{ij} = -s * \tilde{q}_{rj}^{\sim vechi} + c * \tilde{q}_{ij}; \end{cases}$$

La sfârșitul acestui algoritm, în matricea $\boldsymbol{A}$ vom avea matricea superior triunghiulară $\boldsymbol{R}$, în vectorul $\boldsymbol{b}$ vom avea $\widetilde{\boldsymbol{Q}}\boldsymbol{b}^{\text{init}}$($\boldsymbol{b}^{\text{init}}$ - vectorul termenilor liberi inițial), iar matricea $\widetilde{\boldsymbol{Q}}$ va conține matricea $\boldsymbol{Q}^{\boldsymbol{T}}$ din factorizarea $\boldsymbol{QR}$ a matricei $\boldsymbol{A}$.
Numărând operațiile efectuate (exceptând calculul matricei $\widetilde{\boldsymbol{Q}}$) obținem:

$$\frac{\boldsymbol{n(n-1)}}{\boldsymbol{2}}\ \textbf{radicali}$$

$$\frac{\boldsymbol{n(n-1)(4n+7)}}{\boldsymbol{6}} = \frac{\boldsymbol{2}}{\boldsymbol{3}}\boldsymbol{n^3} + \boldsymbol{O(n^2)} - \textbf{adunări/scăderi}$$

$$\frac{\boldsymbol{2n(n-1)(2n+5)}}{\boldsymbol{3}} = \frac{\boldsymbol{4}}{\boldsymbol{3}}\boldsymbol{n^3} + \boldsymbol{O(n^2)}\ \textbf{înmulţiri/împărţiri}$$

QR – algoritmul Gram Schmidt

$$A \in \mathbb{R}^{n\times n} \quad \text{cu} \quad \det A \neq 0$$

$$A=QR$$

$$\begin{bmatrix} a^1 & a^2 & \cdots & a^n \end{bmatrix} = \begin{bmatrix} q^1 & q^2 & \cdots & q^n \end{bmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & & \\ 0 & 0 & \cdots & r_{nn} \end{pmatrix} \quad (1)$$

$a^j = Ae_j$ – coloana j a matricei A

$q^j = Qe_j$ – coloana j a matricei Q

Relația (1) poate fi rescrisă astfel:

$$\begin{cases} r_{11}q^1 = a^1 \\ r_{12}q^1 + r_{22}q^2 = a^2 \\ \vdots \\ r_{1p}q^1 + \cdots + r_{jp}q^j + \cdots + r_{pp}q^p = a^p \\ \vdots \\ r_{1n}q^1 + \cdots + r_{jn}q^j + \cdots + r_{nn}q^n = a^n \end{cases} \qquad (2)$$

Algoritmul de calcul al descompunerii $\boldsymbol{QR}$ cu metoda Gram-Schmidt se desfășoară în $\boldsymbol{n}$ pași, la fiecare pas calculându-se:

- coloana $\boldsymbol{p}$ din matricea $\boldsymbol{R}$
- coloana $\boldsymbol{p}$ din matricea $\boldsymbol{Q}$

Avem:

$$\textbf{det } A \neq 0,\ A = QR,\ Q - \textbf{ ortogonală} \Rightarrow \textbf{ det } R \neq 0\ (r_{ii} \neq 0\ \forall i)$$

Pasul 1

Se folosește prima ecuație a sistemului (2)

$$r_{11}q^1 = a^1$$

Se face produsul scalar al acestei ecuații cu vectorii q^1 şi a^1. Se folosește proprietatea coloanelor matricelor ortogonale:

$$\left(q^i, q^j\right)_{\mathbb{R}^n} = \begin{cases} \| q^i \|_2^2 = 1 & \text{pentru } i = j \\ 0 & \text{pentru } i \neq j \end{cases}$$

$$q^1 = \frac{1}{r_{11}} a^1 \quad \text{şi } \left\| q^1 \right\|_2 = 1$$

$$r_{11} = \pm \left\| a^1 \right\|_2 \ , \quad q^1 = \frac{1}{r_{11}} a^1 \ (r_{11} \neq 0 \text{ deoarece det } A \neq 0)$$

Pasul p

Se folosește ecuația $\boldsymbol{p}$ sistemului (2):

$$r_{1p}q^1 + \cdots + r_{jp}q^j + \cdots + r_{pp}q^p = a^p$$

La acest pas se cunosc deja coloanele $q^1, q^2, \ldots, q^{p-1}$. Se face, pe rând, produsul scalar al acestei ecuații cu vectorii:

$$q^1, q^2, \ldots, q^{p-1}$$

$$q^p \text{ şi } a^p.$$

$$\left(\sum_{k=1}^{p} r_{kp}q^k, q^j \right)_{\mathbb{R}^n} = \left(a^p, q^j \right)_{\mathbb{R}^n} \quad j = 1, \ldots, p-1$$

$$\left(\sum_{k=1}^{p} r_{kp} q^k, q^j\right)_{\mathbb{R}^n} = \sum_{\substack{k=1 \\ k\neq j}}^{p} r_{kp}\left(q^k, q^j\right)_{\mathbb{R}^n} + r_{jp}\left(q^j, q^j\right)_{\mathbb{R}^n} = r_{jp}$$

$$r_{jp} = \left(a^p, q^j\right)_{\mathbb{R}^n} \quad j = 1, \ldots, p-1$$

Avem:

$$q^p = \frac{1}{r_{pp}}\left(a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1}\right)$$

$$\| q^p \|_2^2 = 1 = \frac{1}{r_{pp}^2} \| \left(a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1}\right) \|_2^2$$

Obținem:

$$r_{pp} = \pm \left\| a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1} \right\|_2$$

$$q^p = \frac{1}{r_{pp}} \left(a^p - r_{1p} q^1 - \cdots - r_{p-1\,p} q^{p-1} \right) = \frac{1}{r_{pp}} \left(a^p - \sum_{j=1}^{p-1} r_{jp} q^j \right)$$

Algoritmul Gram Schmidt modificat

for $i = 1,n$

$v^i = a^i;$

for $i = 1, n$

$r_{ii} = \|v^i\|_2;$

$q^i = (1/ r_{ii}) v^i;$

for $j = (i + 1) , n$

$r_{ij} = (q^i , v^j);$

$v^j = v^j - r_{ij}q^i;$

M: $\frac{(n^3 + 3n^2)}{2}$ **A:** $\frac{(n^3 + n^2 - 2)}{2}$

Calcul Numeric

Cursul 6

2022

Anca Ignat

Metode iterative pentru rezolvarea sistemelor de ecuaţii liniare

$$\boldsymbol{Ax = b}\,, \quad \boldsymbol{A} \in \mathbb{R}^{n \times n}\,, \quad \boldsymbol{b} \in \mathbb{R}^{n} \qquad (1)$$

- se presupune cunoscut că $\boldsymbol{A}$ este nesingulară, $\mathbf{det}\ \boldsymbol{A \neq 0}$;
- soluţia exactă a sistemului (1) se notează cu $\boldsymbol{x}^*$:

$$\boldsymbol{x}^* = \boldsymbol{A}^{-1}\boldsymbol{b} \qquad (2)$$

- $\boldsymbol{n}$ - dimensiunea sistemului este "*mare*";
- $\boldsymbol{A}$ este matrice rară - cu "*puţine*" elemente $\boldsymbol{a_{ij} \neq 0}$;

- pentru a aproxima soluţia $\boldsymbol{x}^*$ matricea $\boldsymbol{A}$ nu se schimbă (transformă) ci doar se folosesc elementele nenule ale matricei ;
- se construieşte un şir de vectori $\{\boldsymbol{x}^{(k)}\} \subseteq \mathbb{R}^n$, şir care în anumite cazuri, converge la $\boldsymbol{x}^*$:

$$\boldsymbol{x}^{(k)} \to \boldsymbol{x}^* \quad \textbf{pentru } \boldsymbol{k} \to \infty$$

O schemă generală de deducere a unei metode iterative

Fie descompunerea:

$$\boldsymbol{A = B - C}\ , \quad \boldsymbol{B, C} \in \boldsymbol{R^{n\times n}}\ , \quad \boldsymbol{B}\ \text{“}\textit{uşor}\text{”}\ \text{inversabilă} \qquad (3)$$

Ce înseamnă $\boldsymbol{B}$ "*uşor*" inversabilă ? Sistemul liniar, având ca matrice a sistemului matricea $\boldsymbol{B}$:

$$\boldsymbol{Bx = f}$$

se rezolvă ‘uşor’ (adică repede) – ca în cazul sistemelor cu matrice diagonale sau triunghiulare, de exemplu.

$$Ax^* = b \leftrightarrow Bx^* - Cx^* = b \leftrightarrow$$

$$Bx^* = Cx^* + b \leftrightarrow x^* = B^{-1}Cx^* + B^{-1}b = Mx^* + d$$

unde

$$M := B^{-1}C \in \mathbb{R}^{n\times n}, \quad d := B^{-1}b \in \mathbb{R}^n \tag{4}$$

Şirul $\{x^{(k)}\}$ se construieşte astfel:

$$x^{(k+1)} := Mx^{(k)} + d, k = 0,1,2,\ldots x^{(0)} \in \mathbb{R}^n \text{ ales arbitrar} \tag{5}$$

Vectorul $x^{(k+1)}$ poate fi privit şi ca soluţia sistemului liniar:

$$Bx = f \text{ cu } f := Cx^{(k)} + b \tag{6}$$

Cunoscând vectorul $\boldsymbol{x^{(k)}}$, următorul element din şir, $\boldsymbol{x^{(k+1)}}$, se poate construi fie utilizând relaţia (5) (dacă putem construi matricea $\boldsymbol{M}$ explicit), fie rezolvând sistemul liniar (6).

Matricea $\boldsymbol{M}$ poartă numele de ***matricea iteraţiei*** iar vectorul $\boldsymbol{x^{(0)}} \in \mathbb{R}^n$ se numeşte ***iteraţia iniţială***.

Ne punem problema convergenţei şirului $\boldsymbol{x^{(k)}}$:

$$\boldsymbol{x^{(k)} \to x^*} \quad , \quad \boldsymbol{k \to \infty}$$

Se ştie că această convergenţă nu are loc pentru orice matrice $\boldsymbol{B}$. Avem următorul rezultat general de convergenţă.

Fie $A \in \mathbb{R}^{n\times n}$ **, $\{A^k\}$ un şir de matrici.**

$$A^k \to 0_{n\times n}\,,\, k \to \infty \;\Leftrightarrow\; \left\|A^k\right\| \to 0\,,\, k \to \infty.$$

Propoziţie

Fie $A \in \mathbb{R}^{n\times n}$. Atunci:

$$A^k \to 0\,,\, k \to \infty \;\Leftrightarrow\; \rho(A) < 1.$$

Dacă există o normă matriceală naturală pentru care $\|A\| < 1$ atunci:

$$A^k \to 0 \text{ pentru } k \to \infty.$$

$$(n = 1 \to a \in \mathbb{R}\,,\, a^k \to 0 \text{ pentru } k \to \infty \;\Leftrightarrow\; |a| < 1.)$$

Propoziție

Fie $A \in \mathbb{R}^{n\times n}$. Seria $\sum_{k=0}^{\infty} A^k$ converge dacă şi numai dacă raza spectrală a matricei A este subunitară:

$$\sum_{k=0}^{n} A^k = S \quad \Leftrightarrow \quad \rho(A) < 1.$$

Dacă există o normă a matricei A astfel încât $\|A\| < 1$ atunci seria converge. În cazul convergenței avem :

$$\sum_{k=0}^{\infty} A^k = S = (I - A)^{-1}.$$

Teorema de convergenţă

Fie $A \in \mathbb{R}^{n\times n}$ o matrice nesingulară şi $B, C \in \mathbb{R}^{n\times n}$, $\det B \neq 0$, astfel ca $A=B-C$. Fie $x^{(0)} \in \mathbb{R}^n$ un vector oarecare şi $\{x^{(k)}\}$ şirul de vectori dat de relaţia (5) cu M şi d daţi de (4). Atunci:

$$x^{(k)} \to x^*, k \to \infty, \ \forall x^{(0)} \Leftrightarrow \rho(M) < 1 \qquad (7)$$

unde $\rho(M) = \max\{|\lambda|; \lambda - \text{valoare proprie a matricii } M\}$ este raza spectrală a matricei M. Dacă există o normă matricială naturală astfel ca $\|M\| < 1$ atunci şirul $\{x^{(k)}\}$ converge la soluția x^* a sistemului (1).

$$\|M\| < 1 \Rightarrow \quad x^{(k)} \to x^*, k \to \infty, \forall x^{(0)}. \qquad (8)$$

Demonstrație: Scăzând relațiile (5) şi $x^* = Mx^* + d$ obţinem:

$$x^{(k+1)} - x^* = M(x^{(k)} - x^*), k = 0,1,2,\ldots$$

Avem:

$$x^{(p)} - x^* = M(x^{(p-1)} - x^*) = M^2(x^{(p-2)} - x^*) = \cdots = M^p(x^{(0)} - x^*)$$

$$x^{(p)} - x^* = M^p(x^{(0)} - x^*), \forall p$$

Prin urmare:

$$x^{(p)} \to x^*, p \to \infty \Leftrightarrow M^p \to 0, p \to \infty$$

$$M^p \to 0, p \to \infty \Leftrightarrow \rho(M) < 1$$

Dacă:

$$\| M \| < 1 \Rightarrow M^p \to 0, p \to \infty \Rightarrow x^{(p)} \to x^*, p \to \infty \quad \forall x^{(0)}$$

Evaluarea erorii absolute $\| x^{(k)} - x^* \|$

Presupunem $\|M\|<1$ (şirul $\{x^{(k)}\}$ converge la x^*).
Avem din (5):

$$x^{(l+1)} = Mx^{(l)} + d$$

$$x^{(l)} = Mx^{(l-1)} + d$$

$$x^{(l+1)} - x^{(l)} = M(x^{(l)} - x^{(l-1)}) \quad \forall l$$

Pentru orice k, j, folosind relaţiile de mai sus, avem:

$$x^{(k+j+1)} - x^{(k+j)} = M(x^{(k+j)} - x^{(k+j-1)}) = \cdots = M^j(x^{(k+1)} - x^{(k)}) \; \forall k, j$$

Aplicând succesiv relaţia precedentă obţinem:

$$x^{(k+p)} - x^{(k)} = x^{(k+p)} - x^{(k+p-1)} + x^{(k+p-1)} - x^{(k+p-2)} + \cdots +$$

$$+ x^{(k+2)} - x^{(k+1)} + x^{(k+1)} - x^{(k)} =$$

$$= \sum_{j=0}^{p-1} (x^{(k+j+1)} - x^{(k+j)})$$

$$x^{(k+p)} - x^{(k)} = \sum_{j=0}^{p-1} (x^{(k+j+1)} - x^{(k+j)}) = (\sum_{j=0}^{p-1} M^j)(x^{(k+1)} - x^{(k)})$$

Făcând $p \to \infty$ obţinem:

$$x^* - x^{(k)} = (\sum_{j=0}^{\infty} M^j) M (x^{(k)} - x^{(k-1)})$$

$$\| M \| < 1 \Rightarrow \sum_{j=0}^{\infty} M^j = (I_n - M)^{-1}$$

Mai avem şi evaluarea:

$$\| M \| < 1 \quad \Rightarrow \quad \frac{1}{1 + \| M \|} \leq \| (I_n - M)^{-1} \| \leq \frac{1}{1 - \| M \|}$$

Prin urmare:

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

Această relaţie ne spune că din punct de vedere practic putem opri algoritmul atunci când diferenţa dintre două iteraţii succesive devine suficient de mică, acest lucru asigurând apropierea de soluţie.

În continuare vedea cum se memorează matricele rare și vom particulariza matricea ***B***.

Memorarea matricelor rare

- se memorează doar valorile nenule şi suficiente informaţii despre indici astfel ca să se poată reconstitui complet matricea

Pp. că matricea ***A*** are ***NN*** elemente nenule.

Memorare comprimată pe linii

Se folosesc 3 vectori:

valori – vector de numere reale de dimensiune ***NN***
ind_col – vector de indici de dimensiune ***NN***
inceput_linii – vector de întregi de dimensiune ***n+1***

În vectorul ***valori*** se memorează elementele nenule ale matricii ***A*** în ordinea liniilor iar în vectorul ***ind_col*** se memorează indicii de coloană ai elementelor din ***valori***. În vectorul ***inceput_linii*** se stochează indicele/poziţia în vectorul ***valori / ind_col*** al/a primului element de pe linia ***i*** memorat în vectorii ***valori / ind_col***.

- ***inceput_linii(n+1) = NN+1***
- ***inceput_linii(i+1) – inceput_linii(i)*** =
  numărul de elemente nenule de pe linia ***i, i=1,n***

$$A = \begin{pmatrix} 102.5 & 0.0 & 2.5 & 0.0 & 0.0 \\ 3.5 & 104.88 & 1.05 & 0.0 & 0.33 \\ 0.0 & 0.0 & 100.0 & 0.0 & 0.0 \\ 0.0 & 1.3 & 0.0 & 101.3 & 0.0 \\ 0.73 & 0.0 & 0.0 & 1.5 & 102.23 \end{pmatrix}$$

$n=5$, $NN=12$

valori = (102.5, 2.5, 0.33, 1.05, 104.88, 3.5, 100.0, 101.3, 1.3, 1.5, 0.73, 102.23)

ind_col = (1, 3, 5, 3, 2, 1, 3, 4, 2, 4, 1, 5)

inceput_linii = (1, 3, 7, 8, 10, 13)

Dacă se ştie că matricea are maxim ***n_max*** elemente nenule pe fiecare linie se pot folosi 2 matrice pentru memorarea rară:

valori – matrice de numere reale de dimensiune ***n*** × ***n_max***
ind_col – matrice de indici de dimensiune ***n*** × ***n_max***

În matricea ***valori*** se memorează pe linia ***i*** elementele nenule de pe linia ***i*** a matricei ***A*** iar în matricea ***ind_col*** se memorează indicii de coloană ai elementelor corespunzătoare din matricea ***valori***.

$$A = \begin{pmatrix} 102.5 & 0.0 & 2.5 & 0.0 & 0.0 \\ 0.0 & 104.88 & 1.05 & 0.0 & 0.33 \\ 0.0 & 0.0 & 100.0 & 0.0 & 0.0 \\ 0.0 & 1.3 & 0.0 & 101.3 & 0.0 \\ 0.73 & 0.0 & 0.0 & 1.5 & 102.23 \end{pmatrix}$$

$$valori = \begin{pmatrix} 102.5 & 2.5 & 0 \\ 104.88 & 1.05 & 0.33 \\ 100.0 & 0 & 0 \\ 101.3 & 1.3 & 0 \\ 102.23 & 1.5 & 0.73 \end{pmatrix} \qquad ind\_col = \begin{pmatrix} 1 & 3 & 0 \\ 2 & 3 & 5 \\ 3 & 0 & 0 \\ 4 & 2 & 0 \\ 5 & 4 & 1 \end{pmatrix}$$

Diagonalele matricei $\boldsymbol{A}$:

$$\boldsymbol{d_0 : (a_{11}, a_{22}, \ldots, a_{nn})}$$
$$\boldsymbol{d_1 : (a_{12}, a_{23}, \ldots, a_{n-1n})}$$
$$\boldsymbol{d_{-1} : (a_{21}, a_{32}, \ldots, a_{nn-1})}$$
$$\boldsymbol{d_2 : (a_{13}, a_{24}, \ldots, a_{n-2n})}$$
$$\boldsymbol{d_{-2} : (a_{31}, a_{42}, \ldots, a_{nn-2})}$$
$$\vdots$$

Pentru matricele care au elementele nenule plasate pe câteva din diagonalele matricei $\boldsymbol{A}$ ($\boldsymbol{n\_d}$ diagonale cu elemente nenule) se pot folosi pentru memorare o matrice şi un vector:

diag – matrice cu numere reale de dimensiune $n \times n\_d$
diag_no – vector de întregi de dimensiune ***n_d***

În matricea ***diag*** se memorează pe coloane diagonalele cu elemente nenule iar în ***diag_no*** este specificat numărul diagonalei care e memorat în coloana ***j*** a matricei ***diag***.

$$diag(i, j) = a_{i\ i+diag\_no(j)}$$

$$A = \begin{pmatrix} 20.5 & 2.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 40.5 & 3.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 100.0 & 0.0 & 0.0 \\ 0.0 & 2.3 & 0.0 & 101.5 & 4.0 \\ 0.0 & 0.0 & 3.0 & 0.0 & 102.5 \end{pmatrix}$$

$$diag = \begin{pmatrix} * & 20.5 & 2.0 \\ * & 40.5 & 3.0 \\ 1.0 & 100.0 & 0.0 \\ 2.3 & 101.5 & 4.0 \\ 3.0 & 102.5 & * \end{pmatrix} \quad diag\_no = (-2, 0, 1)$$

Alte tipuri de memorări rare:
http://netlib.org/linalg/html_templates/node90.html

Metoda Jacobi pentru rezolvarea sistemelor liniare

Fie sistemul:

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

cu

$$\det A \neq 0 \quad , \quad a_{ii} \neq 0\,, i = 1,2,\ldots,n$$

Alegem:

$$B = \operatorname{diag}[a_{11}, a_{22}, \ldots, a_{nn}] = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix}$$

$$\det B = a_{11}a_{22}\cdots a_{nn} \neq 0$$

$$B^{-1} = \operatorname{diag}\left[\frac{1}{a_{11}}, \frac{1}{a_{22}}, \ldots, \frac{1}{a_{nn}}\right] = \begin{pmatrix} \frac{1}{a_{11}} & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{a_{nn}} \end{pmatrix}$$

Matricea C este:

$$C = B - A = \begin{pmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ -a_{21} & 0 & -a_{23} & \cdots & -a_{2n} \\ -a_{31} & -a_{32} & 0 & \cdots & -a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & -a_{n3} & \cdots & 0 \end{pmatrix}$$

$$C = (c_{ij}) \in \mathbb{R}^{n \times n} \qquad c_{ij} = \begin{cases} -a_{ij} & \text{dacă } i \neq j \\ 0 & \text{dacă } i = j \end{cases}$$

Matricea iteraţiei se poate calcula şi are forma:

$$M := B^{-1}C = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \dots & -\frac{a_{2n}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0 & \dots & -\frac{a_{3n}}{a_{33}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \dots & 0 \end{pmatrix}$$

$$M = (m_{ij}) \in \mathbb{R}^{n \times n} \qquad m_{ij} = \begin{cases} -\left(\dfrac{a_{ij}}{a_{ii}}\right) & \text{dacă } i \neq j \\ 0 & \text{dacă } i = j \end{cases}$$

Construim vectorul g:

$$g := Mx^{(k)} \in \mathbb{R}^n \quad , \quad Mx^{(k)} = (g_i)_{i=1}^n$$

Componentele vectorului g sunt:

$$g_i = \sum_{j=1}^{n} m_{ij} x_j^{(k)} = -\sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k)} = -\left(\sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)}\right) / a_{ii} \quad , i = 1, \ldots, n$$

Vectorul $\boldsymbol{d}$ este:

$$\boldsymbol{d = B^{-1}b = (d_i)_{i=1}^{n} \in \mathbb{R}^n , \quad d_i = \frac{b_i}{a_{ii}} , i = 1,\ldots,n}$$

Şirul $\{x^{(k)}\} \subseteq \mathbb{R}^n$ se construieşte folosind formula:

$$\boldsymbol{x^{(k+1)} = Mx^{(k)} + d \quad \leftrightarrow \quad x_i^{(k+1)} = g_i + d_i , i = 1,\ldots,n}$$

$$\boldsymbol{x_i^{(k+1)} = \frac{\left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right)}{a_{ii}} , \quad i = 1,\ldots,n}$$

$$x_i^{(k+1)} = \frac{(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)})}{a_{ii}} \quad , \quad i = 1,\ldots,n \qquad (1)$$

Formula (1) descrie ***metoda lui Jacobi*** de aproximare a soluţiei unui sistem liniar.

Condiţii suficiente de convergenţă

Propoziţia 1

$$\| M \| < 1 \quad \Rightarrow \quad x^{(k)} \to x^{*}, k \to \infty.$$

Demonstraţie. Fie x^{*} soluţia sistemului $Ax=b$. Din relaţia $A=B-C$ rezultă $Bx^{*} = Cx^{*} + b$ sau $x^{*} = Mx^{*} + d$.
Procesul iterativ $x^{(k+1)} = Mx^{(k)} + d$ conduce la relaţia:

$$\left\| x^{*} - x^{(k+1)} \right\| = \left\| M\left(x^{*} - x^{(k)} \right) \right\| \le \| M \| \left\| x^{*} - x^{(k)} \right\| \le \cdots \le \| M \|^{k+1} \left\| x^{*} - x^{(0)} \right\|$$

În continuare vom aplica această propoziţie pentru diverse norme.

- Din $\| M \|_F = (\sum_{i=1}^{n} \sum_{j=1}^{n} m_{ij}^2)^{\frac{1}{2}} < 1$ deducem:

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{a_{ij}}{a_{ii}} \right)^2 < 1 \quad \Rightarrow \quad x^{(k)} \to x^* \, , k \to \infty$$

- Din $\| M \|_1 = \max\{\sum_{i=1}^{n} | m_{ij} |; j = 1,\ldots,n\} < 1$ deducem:

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} \left(\frac{|a_{ij}|}{|a_{ii}|} \right) < 1 \quad \forall j = 1,\ldots,n \quad \Rightarrow \quad x^{(k)} \to x^{*}, k \to \infty$$

- ***(Criteriul dominanţei diagonalei pe linii)***

Din $\| M \|_\infty = \max\{\sum_{j=1}^{n} | m_{ij} |; i = 1,\ldots,n\} < 1$ deducem:

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} \left(\frac{|a_{ij}|}{|a_{ii}|} \right) < 1 \quad \forall i = 1,\ldots,n \quad \Rightarrow \quad x^{(k)} \to x^{*}, k \to \infty$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| < |a_{ii}| \quad \forall i = 1,\ldots,n \Rightarrow \lim_{k \to \infty} x^{(k)} = x^{*}$$

- ***(Criteriul dominanţei diagonalei pe coloane)***

$$\sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}| < |a_{jj}| \quad \forall j = 1,\ldots,n \Rightarrow \|M\|_1 < 1 \Rightarrow \lim_{k\to\infty} x^{(k)} = x^*$$

Metoda Gauss-Seidel pentru rezolvarea sistemelor liniare

Considerăm din nou sistemul liniar:

$$\boldsymbol{Ax = b}\,, \quad \boldsymbol{A} \in \mathbb{R}^{n\times n}\,, \quad \boldsymbol{b} \in \mathbb{R}^{n}$$

cu

$$\boldsymbol{\det A \neq 0} \quad , \quad \boldsymbol{a_{ii} \neq 0}\,, \boldsymbol{i = 1, 2, \ldots, n}$$

Putem deduce metoda Gauss-Seidel din metoda lui Jacobi astfel:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} \; x_j^{(k)} \; - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}\right) / a_{ii} \;, i = 1,\ldots,n \; - \; \textbf{Jacobi}$$

$$\updownarrow$$

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} \; x_j^{(k+1)} \; - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}\right) / a_{ii}, i = 1,\ldots,n - \textbf{Gauss-Seidel}$$

Când calculăm $x_i^{(k+1)}$ cunoaştem deja $x_1^{(k+1)},\cdots, x_{i-1}^{(k+1)}$ şi putem folosi aceste valori în prima sumă.

Deducerea metodei Gauss-Seidel din schema generală se face luând:

$$B = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$$

$$B = (b_{ij}) \in \mathbb{R}^{n \times n} \quad b_{ij} = \begin{cases} a_{ij} & \text{dacă } j \leq i \\ 0 & \text{dacă } j > i \end{cases}$$

Matricea B este nesingulară ($a_{ii} \neq 0, \forall i$):

$$\det B = a_{11} a_{22} \cdots a_{nn} \neq 0$$

Matricea C este:

$$C = B - A = \begin{pmatrix} 0 & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ 0 & 0 & -a_{23} & \cdots & -a_{2n} \\ 0 & 0 & 0 & \cdots & -a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -a_{n-1n} \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$C = (c_{ij}) \in \mathbb{R}^{n \times n} \quad c_{ij} = \begin{cases} -a_{ij} & \text{dacă } i < j \\ 0 & \text{dacă } i \geq j \end{cases}$$

În cazul metodei Gauss-Seidel, vectorul $x^{(k+1)}$ se obţine din $x^{(k)}$ rezolvând sistemul inferior triunghiular (7) din schema generală:

$$Bx = Cx^{(k)} + b = f \qquad (1)$$

Soluţia sistemului (1) este dată de formula:

$$x_i = \left(f_i - \sum_{j=1}^{i-1} b_{ij} x_j \right) / b_{ii} = \left(f_i - \sum_{j=1}^{i-1} a_{ij} x_j \right) / a_{ii} , \quad i = 1, 2, \ldots, n \qquad (2)$$

Vectorul f este:

$$f_i = (Cx^{(k)})_i + b_i\ , \quad i = 1,2,\ldots,n \tag{3}$$

$$(Cx^{(k)})_i = \sum_{j=1}^{n} c_{ij} x_j^{(k)} = -\sum_{j=i+1}^{n} a_{ij} x_j^{(k)}\ , \forall i = 1,\ldots,n \tag{4}$$

Folosind formula de rezolvare a sistemelor inferior triunghiulare (2), din relaţiile (3) şi (4) avem:

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

Condiţii suficiente de convergenţă pentru metoda Gauss-Seidel

Propoziţia 1

Dacă matricea $\boldsymbol{A}$ este astfel încât:

$$\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\left(\frac{a_{ij}}{a_{ii}}\right)^2 < 1$$

atunci are loc convergenţa şirului construit cu metoda Gauss-Seidel la soluţia sistemului $\boldsymbol{Ax=b}$:

$$x^{(k)} \to x^{*}\ ,\ k \to \infty \quad \forall x^{(0)} \in \mathbb{R}^n$$

Propoziţia 2 ***(Criteriul dominanţei diagonalei pe linii)***

Dacă matricea $\boldsymbol{A}$ este astfel încât:

$$\sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| < |a_{ii}| \quad \forall i = 1,\ldots,n$$

atunci:

$$x^{(k)} \to x^{*}\ , k \to \infty \quad \forall x^{(0)} \in \mathbb{R}^{n}$$

Propoziţia 3 ***(Criteriul dominanţei diagonalei pe coloane)***

Dacă matricea $\boldsymbol{A}$ este astfel încât:

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} |a_{ij}| < |a_{jj}| \quad \forall j = 1,\ldots,n$$

atunci metoda Gauss-Seidel converge:

$$\lim_{k \to \infty} x^{(k)} = x^{*}, \quad \forall x^{(0)}$$

Calcul Numeric

Cursul 7

2022

Anca Ignat

Metode iterative pentru matrice simetrice şi pozitiv definite

Considerăm cazul sistemelor liniare cu matricea sistemului simetrică şi pozitiv definită:

$$A = A^T - \text{matrice simetrică} - a_{ij} = a_{ji} \quad \forall i, j = 1, 2, \ldots n$$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix} = A^T = \begin{pmatrix} a_{11} & a_{21} & a_{31} & \cdots & a_{n1} \\ a_{12} & a_{22} & a_{32} & \cdots & a_{n2} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{pmatrix}$$

$$A = A^T \quad \Rightarrow \quad A = L + D + L^T$$

$$D = \mathrm{diag}[a_{11}, a_{22}, \ldots, a_{nn}] = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{pmatrix} \quad L^T = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1n} \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Definiţii

Matricea $A \in \mathbb{R}^{n\times n}$ se numeşte ***pozitiv semidefinită*** ($A \geq 0$):

$$\left(Ax, x\right)_{\mathbb{R}^n} \geq 0 \quad \forall x \in \mathbb{R}^n.$$

Matricea A se numeşte ***pozitiv definită*** ($A > 0$) dacă:

$$\left(Ax, x\right)_{\mathbb{R}^n} > 0 \quad \forall x \in \mathbb{R}^n, x \neq 0.$$

Propoziţie

Dacă matricea $\boldsymbol{A \in \mathbb{R}^{n\times n}}$ este pozitiv definită atunci matricea $\boldsymbol{A}$ este nesingulară.

Demonstraţie: Presupunem prin reducere la absurd că matricea $\boldsymbol{A}$ este pozitiv definită şi singulară. Atunci, sistemul de ecuaţii liniare $\boldsymbol{Ax=0}$ are pe lângă soluţia banală $\boldsymbol{x=0}$ şi o soluţie $\boldsymbol{x^0 \neq 0}$. Avem:

$$\boldsymbol{x^0 \neq 0 \;\Rightarrow\; 0 < \left(Ax^0, x^0\right) = \left(0, x^0\right) = 0}$$ **contradicţie!**

$$\boldsymbol{A > 0 \;\Rightarrow\; a_{ii} = \left(Ae_i, e_i\right) > 0 \;\; \forall i = 1,\ldots,n}$$

Lemă

Fie $A \in \mathbb{R}^{n \times n}$ o matrice simetrică şi $B \in \mathbb{R}^{n \times n}$ o matrice nesingulară astfel încât matricea $P = B + B^T - A$ este pozitiv definită. Fie matricea $M = I_n - B^{-1}A$. Atunci raza spectrală a matricei M este strict subunitară dacă şi numai dacă matricea A este pozitiv definită:

$$\rho(M) < 1 \quad \Leftrightarrow \quad A > 0$$

Teoremă

Fie $A \in \mathbb{R}^{n \times n}$ o matrice simetrică, nesingulară, cu diagonala pozitivă, $a_{ii} > 0, \ \forall \ i = 1,\ldots,n$ şi $b \in \mathbb{R}^n$ vectorul termenilor liberi. Atunci metoda lui Gauss-Seidel generează şiruri convergente la soluţia $x^* = A^{-1}b, \forall\, x^{(0)}$ dacă şi numai dacă A este pozitiv definită .

Demonstraţie: Din teorema de convergenţă avem:

$$x^{(k)} \to x^{*}\ , \quad k \to \infty \quad \Leftrightarrow \quad \rho(M) < 1$$

Dacă matricea A se scrie sub forma:

$$A = L + D + L^{T}$$

matricele B şi C sunt date de:

$$B = L + D \quad , \quad C = B - A = -L^{T}$$

Matricea iteraţiei M este:

$$M = B^{-1}C = B^{-1}(B - A) = I_n - B^{-1}A$$

Încercăm să aplicăm **Lema** de mai sus. Pentru aceasta verificăm dacă matricea $\boldsymbol{P}$ este pozitiv definită:

$$\boldsymbol{P = B + B^T - A = L + D + (L + D)^T - L - D - L^T = D}$$

$$\boldsymbol{(Px, x)_{\mathbb{R}^n} = (Dx, x)_{\mathbb{R}^n} = ((a_{ii} x_i)_i, (x_i)_i)_{\mathbb{R}^n} = \sum_{i=1}^{n} a_{ii} x_i^2}$$

$$\boldsymbol{a_{ii} > 0 \quad \forall i \Rightarrow (Px, x)_{\mathbb{R}^n} > 0 \quad \forall x \in \mathbb{R}^n, x \neq 0 \Rightarrow P > 0}$$

Putem aplica **Lema** de unde deducem convergenţa şirului construit cu metoda Gauss-Seidel doar în cazul în care matricea $\boldsymbol{A}$ este pozitiv definită:

$$\boldsymbol{x^{(k)} \to x^*, \quad k \to \infty \Leftrightarrow \rho(M) < 1 \Leftrightarrow A}$$ **pozitiv definită**

Metodele relaxării

Fie $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ o matrice reală pătratică de dimensiune n, simetrică, $\boldsymbol{A}=\boldsymbol{A}^{\boldsymbol{T}}$ şi pozitiv definită, $\boldsymbol{A} > \boldsymbol{0}$ şi $\boldsymbol{b} \in \mathbb{R}^{n}$ un vector real. Considerăm sistemul de ecuaţii liniare:

$$\boldsymbol{Ax} = \boldsymbol{b}$$

Deoarece matricea A este pozitiv definită sistemul de mai sus are soluţie unică, $\boldsymbol{x}^{*} = \boldsymbol{A}^{-1}\boldsymbol{b}$. Vom considera funcţia $\boldsymbol{f} : \mathbb{R}^{n} \to \mathbb{R}_{+}$:

$$\boldsymbol{f}(\boldsymbol{y}) = \left(\boldsymbol{A}(\boldsymbol{x}^{*} - \boldsymbol{y}), \boldsymbol{x}^{*} - \boldsymbol{y} \right)_{\mathbb{R}^{n}} , \ \boldsymbol{y} \in \mathbb{R}^{n}$$

Din faptul că matricea $\boldsymbol{A}$ este pozitiv definită avem:

$$\boldsymbol{f(y) \geq 0,\ \forall y \in \mathbb{R}^n \quad şi\ f(y) > 0 = f(x^*)\ ,\forall y \neq x^*}$$

Prin urmare $\boldsymbol{x^*}$ este şi unica soluţie a problemei de minimizare:

$$\boldsymbol{\min\left\{ f(y);\ y \in \mathbb{R}^n \right\} = 0 = f(x^*)}$$

Vom căuta soluţia sistemului $\boldsymbol{Ax=b}$, $\boldsymbol{x^* = A^{-1}b}$ ca fiind soluţia problemei de minimizare de mai sus folosind o metodă de tip relaxare de forma:

$$y^{(0)} \in \mathbb{R}^n - \text{dat}, \quad y^{(k+1)} = y^{(k)} + c_k e_l \ , \ \ l = l_k \ \ , \ \ k = 0,1,\ldots$$

$$y_j^{(k+1)} = y_j^{(k)} \ , \ \ \forall \ j \neq l \ , \quad y_l^{(k+1)} = y_l^{(k)} + c_k$$

Constanta c_k se determină astfel încât $f(y^{(k+1)}) < f(y^{(k)})$ în speranţa că şirul $y^{(k)}$ astfel construit converge la x^*. Notăm cu

$r^{(k)} = b - Ay^{(k)}$ vectorul reziduu.

Avem:

$$r^{(k)} = b - Ay^{(k)} = Ax^* - Ay^{(k)} = A(x^* - y^{(k)})$$

$$f(y^{(k+1)}) = f(y^{(k)}) - 2\, c_k\, r_l^{(k)} + c_k^2 a_{ll}$$

Pentru ca $f(y^{(k+1)}) < f(y^{(k)})$ este necesar şi suficient să alegem c_k astfel ca:

$$c_k^2 a_{ll} - 2 c_k r_l^{(k)} < 0 \quad \Leftrightarrow \quad (a_{ll} > 0)\, c_k \in \left(0, 2\frac{r_l^{(k)}}{a_{ll}}\right) \text{sau} \left(2\frac{r_l^{(k)}}{a_{ll}}, 0\right)$$

$$c_k = \omega_k \frac{r_l^{(k)}}{a_{ll}} \quad , \quad \text{cu } \omega_k \in (0,2)$$

Metoda de relaxare obţinută este următoarea:

$$y^{(0)} \in \mathbb{R}^n - \text{dat}, \; y^{(k+1)} = y^{(k)} + \omega_k \frac{r_l^{(k)}}{a_{ll}} e_l \, , \quad k = 0,1,\ldots \, , \quad \omega_k \in (0,2)$$

Pentru a aproxima $\boldsymbol{x}^*$ se deduce o clasă de metode numite ***metodele relaxării succesive***. Aceste metode se obţin aplicând metodele de relaxare de mai sus. Vom considera:

$$\omega_k = \omega, \forall k$$

Vom construi un şir $\left\{ \boldsymbol{x}^{(k)} \right\} \subseteq \mathbb{R}^n$ astfel:

$$x^{(0)} = y^{(0)} \text{ un vector din } \mathbb{R}^n \text{ dat}$$

$$l = 1 \quad y^{(1)} = y^{(0)} + \omega \frac{r_1^{(0)}}{a_{11}} e_1$$

$$l = 2 \quad y^{(2)} = y^{(1)} + \omega \frac{r_2^{(1)}}{a_{22}} e_2$$

$$\vdots$$

$$l = n \quad y^{(n)} = y^{(n-1)} + \omega \frac{r_n^{(n-1)}}{a_{nn}} e_n$$

$$x^{(1)} = y^{(n)}$$

Trecerea de la iteraţia k la iteraţia următoare se face astfel:

$$x^{(k)} = y^{(kn)}$$

$$l = 1 \quad y^{(kn+1)} = y^{(kn)} + \omega \frac{r_1^{(kn)}}{a_{11}} e_1$$

$$l = 2 \quad y^{(kn+2)} = y^{(kn+1)} + \omega \frac{r_2^{(kn+1)}}{a_{22}} e_2$$

$$\vdots$$

$$l = n \quad y^{(kn+n)} = y^{(kn+n-1)} + \omega \frac{r_n^{(kn+n-1)}}{a_{nn}} e_n$$

$$x^{(k+1)} = y^{((k+1)n)}, \; k = 0,1,2,\ldots$$

Acum putem scrie dependenţa vectorului $x^{(k+1)}$ de $x^{(k)}$:

$x^{(0)} \in \mathbb{R}^n$, $\omega \in (0,2)$ **date,**

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

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

$$k = 0,1,2,\ldots$$

Metodele de mai sus poartă numele de ***metodele relaxării succesive***. Pentru $\omega = 1$ obţinem metoda Gauss-Seidel.

- $\mathbf{0 < \omega < 1}$ metodele se numesc de ***sub-relaxare*** şi pot fi folosite în cazul când metoda Gauss-Seidel diverge.

- $\mathbf{1 < \omega < 2}$ metodele se numesc de ***supra-relaxare*** şi pot fi folosite pentru accelerarea convergenţei în cazul când metoda Gauss-Seidel converge.

Rearanjând formulele de mai sus avem:

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

$$= \left(C\, x^{(k)} \right)_i + \left(b \right)_i$$

Matricea $\boldsymbol{A}$ fiind simetrică, poate fi scrisă sub forma:

$$\boldsymbol{A = L + D + L^T} \quad \textbf{cu} \quad \boldsymbol{L} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & 0 & \mathbf{0} \\ \boldsymbol{a_{21}} & \mathbf{0} & \cdots & 0 & \mathbf{0} \\ \vdots & & & & \\ \boldsymbol{a_{n1}} & \boldsymbol{a_{n2}} & \cdots & \boldsymbol{a_{n\,n-1}} & \mathbf{0} \end{pmatrix},$$

$$\boldsymbol{D} = \mathbf{diag}\left[\boldsymbol{a_{11}}, \boldsymbol{a_{22}}, \ldots, \boldsymbol{a_{nn}}\right]$$

Cu aceste notaţii, matricile $\boldsymbol{B}$ şi $\boldsymbol{C}$ de mai sus pot fi scrise astfel:

$$\boldsymbol{B = L + \frac{1}{\omega} D} \qquad , \qquad \boldsymbol{C = \frac{1-\omega}{\omega} D - L^T}$$

Vom verifică dacă metodele relaxării succesive se înscriu în clasa generală de metode iterative, adică vom verifica dacă $\boldsymbol{A=B-C}$:

$$\boldsymbol{B - C = L + \frac{1}{\omega} D - \frac{1-\omega}{\omega} D + L^T = L + D + L^T = A}$$

Convergenţa şirului $\boldsymbol{x^{(k)}}$ la soluţia $\boldsymbol{x^* = A^{-1} b}$?

Teoremă

Fie o matrice $A \in \mathbb{R}^{n \times n}$, simetrică, $A=A^T$ cu $\det A \neq 0$, $a_{ii}>0$, $\forall i = 1,\ldots,n$, $b \in \mathbb{R}^n$ un vector real şi $\omega \in (0,2)$. Atunci şirul $x^{(k)}$ construit cu o metoda de relaxare succesivă converge la soluţia x^* a sistemului liniar $Ax=b$ oricare ar fi iteraţia iniţială $x^{(0)}$ dacă şi numai dacă matricea A este pozitiv definită.

$$x^{(k)} \to x^*, k \to \infty, \forall\, x^{(0)} \quad \Leftrightarrow \quad (Ax, x) > 0, \forall x \in \mathbb{R}^n, x \neq 0$$

Demonstraţie: Vom verifica dacă raza spectrală a matricei iteraţiei este subunitară folosind **Lema.** Avem:

$$M = B^{-1}C = B^{-1}\left(B - A\right) = I_n - B^{-1}A$$

$$B = L + \frac{1}{\omega}D \quad , \ \det B = \frac{1}{\omega^n} a_{11}a_{22}\cdots a_{nn} \neq 0 \quad (a_{ii} > 0 \, , \ \forall i)$$

Matricea A este simetrică iar matricea B este nesingulară. Pentru a fi îndeplinite ipotezele Lemei trebuie să verificăm că matricea P este pozitiv definită:

$$P = B + B^T - A = L + \frac{1}{\omega}D + L^T + \frac{1}{\omega}D - L - D - L^T = \frac{2-\omega}{\omega}D$$

$$\left(Px, x\right) = \frac{(2-\omega)}{\omega} \sum_{i=1}^{n} a_{ii} x_i^2 > 0\,, \;\; x \neq 0 \;\; (a_{ii} > 0, \forall i) \Leftrightarrow$$

$$\Leftrightarrow \frac{(2-\omega)}{\omega} > 0 \;\; \Leftrightarrow \omega \in \left(0, 2\right)$$

Toate ipotezele lemei sunt îndeplinite, prin urmare avem convergența dorită.

Valori şi vectori proprii
(eigenvalues, eigenvectors)

Definiţie

Fie $A \in \mathbb{R}^{n \times n}$. Numărul complex $\lambda \in \mathbb{C}$ se numeşte ***valoare proprie*** a matricei A dacă există un vector $u \in \mathbb{C}^n$, $u \neq 0$ astfel ca:

$$Au = \lambda u$$

Vectorul u se numeşte ***vector propriu*** asociat valorii proprii λ.

Pentru existenţa vectorului $u \neq 0$ este necesar şi suficient ca matricea $(\lambda I_n - A)$ să fie singulară , adică $\det(\lambda I_n - A) = 0$.

Polinomul de grad n:

$$p_A(\lambda) = \det(\lambda I_n - A)$$

se numeşte ***polinom caracteristic*** al matricei A.

Propoziţia 1

Fie rădăcinile polinomului caracteristic $\lambda_1, \lambda_2, ..., \lambda_n$ distincte, $\lambda_i \neq \lambda_j$ pentru $1 \leq i < j \leq n$ şi $u_1, u_2, \ldots, u_n$ vectorii proprii corespunzători. Atunci $u_1, u_2, \ldots, u_n$ sunt liniar independenţi. (demonstraţia se face prin inducţie)

Propoziţia 2

Fie valorile proprii λ_i ale matricei $A \in \mathbb{R}^{n\times n}$ distincte. Atunci există o matrice nesingulară T astfel ca:

$$T^{-1}AT = \mathbf{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n].$$

Demonstraţie. Fie $u_1, u_2, \ldots, u_n$ vectorii proprii ai matricei A. Considerăm matricea T ale cărei coloane sunt vectorii proprii u_i, $T = [u_1 \quad u_2 \quad \ldots \quad u_n]$. Deoarece vectorii proprii sunt liniar independenţi conform propoziţiei 1 rezultă că matricea T este nesingulară. Vom avea:

$$AT = \left[Au_1\ Au_2 \ldots Au_n\right] = \left[\lambda_1 u_1\ \lambda_2 u_2 \ldots \lambda_n u_n\right] = T.\mathbf{diag}\left[\lambda_1\ \lambda_2 \ldots \lambda_n\right]$$

Înmulţind la stânga cu T^{-1} obţinem concluzia propoziţiei 2.

Definiţie

Matricile A şi B sunt ***asemenea*** (notaţie $A \sim B$) dacă şi numai dacă există o matrice nesingulară T ($\det T \neq 0$) astfel ca:

$$A = T\,B\,T^{-1}$$

Propoziţia 3

$$A \sim B \;\Rightarrow\; p_A(\lambda) = p_B(\lambda).$$

Demonstraţie.

$$p_A(\lambda) = \det(\lambda I_n - A) = \det\left(\lambda I_n - TBT^{-1}\right) = \det\left(\lambda TT^{-1} - TBT^{-1}\right)$$

$$= \det\left(T\left(\lambda I_n - B\right)T^{-1}\right) = \det(T)\det\left(\lambda I_n - B\right)\det\left(T^{-1}\right) = p_B(\lambda)$$

Propoziţia 3 ne spune că matricele asemenea au același polinom caracteristic şi aceleași valori proprii.

Teorema lui Gershgorin

Fie $A \in \mathbb{R}^{n \times n}$ şi $\lambda \in \mathbb{C}$ o valoare proprie oarecare a matricei A. Atunci:

$$\exists i_0 \in \{1, 2, \ldots, n\} \text{ astfel încât } |\lambda - a_{i_0 i_0}| \leq r_{i_0}, \; r_{i_0} = \sum_{\substack{j=1 \\ j \neq i_0}}^{n} |a_{i_0 j}|.$$

(Valoarea proprie λ se află în cercul din planul complex de centru $a_{i_0 i_0}$ şi rază r_{i_0}.)

Demonstraţie. Fie $\lambda \in \mathbb{C}$ o valoare proprie a matricei A şi $u \neq 0$ un vector propriu asociat valorii proprii λ, $Au = \lambda u$.
Avem:

$$\lambda u_i = a_{ii} u_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} u_j \Leftrightarrow (\lambda - a_{ii}) u_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} u_j, i = 1, \ldots, n.$$

Fie i_0 astfel ca $|u_{i_0}| = \|u\|_\infty = \max\{|u_k|; k = 1, \ldots, n\} > 0$ ($u \neq 0$).
Vom avea:

$$|\lambda - a_{i_0 i_0}| = \left| \sum_{\substack{j=1 \\ j \neq i_0}}^{n} a_{i_0 j} \frac{u_j}{u_{i_0}} \right| \leq \sum_{\substack{j=1 \\ j \leq i_0}}^{n} |a_{i_0 j}| \frac{|u_j|}{|u_{i_0}|} \leq r_{i_0}$$, ţinând seama că $\frac{|u_j|}{|u_{i_0}|} \leq 1$.

Observaţie. Presupunem că matricea $\boldsymbol{A}$ are $\boldsymbol{n}$ vectori proprii liniar independenţi $\boldsymbol{u^1, u^2, \ldots, u^n}$ asociaţi valorilor proprii $\boldsymbol{\lambda_1, \lambda_2, \ldots, \lambda_n}$ Fie $\boldsymbol{U = \left[u^1\ u^2 \ldots u^n \right]}$. Datorită independenţei vectorilor $\boldsymbol{u^k}$ rezultă că matricea $\boldsymbol{U}$ este nesingulară şi avem:

$$\Lambda = \operatorname{diag}\left[\lambda_1, \lambda_2, \ldots, \lambda_n\right] \qquad \boldsymbol{U^{-1} A U = \Lambda.}$$

Considerăm matricea perturbată:

$$\boldsymbol{A(\varepsilon) = A + \varepsilon B.}$$

$$\boldsymbol{U^{-1} A(\varepsilon) U = \Lambda + \varepsilon U^{-1} B U = \Lambda + \varepsilon C}\ .$$

$\boldsymbol{A(\varepsilon) \sim U^{-1} A(\varepsilon)\, U \Rightarrow}$ au aceleaşi valori proprii $\boldsymbol{\lambda_i(\varepsilon)}$

$$\left|\lambda(\varepsilon) - \lambda_i - \varepsilon c_{ii}\right| \le \varepsilon \sum_{\substack{j=1 \\ j \ne i}}^{n} \left|c_{ij}\right| \ \Rightarrow\ \left|\lambda(\varepsilon) - \lambda_i\right| = \mathcal{O}(\varepsilon).$$

Metoda puterii pentru matrice simetrice

Propoziţie

Fie $A \in \mathbb{R}^{n\times n}, A = A^T$. Atunci toate valorile proprii ale matricei A sunt numere reale.

Demonstraţie. Fie $\lambda \in \mathbb{C}$ şi $u \in \mathbb{C}^n, u \neq 0$, $Au = \lambda u$.

Considerăm produsul scalar:

$$(Au,u)_{\mathbb{C}^n} = \lambda (u,u)_{\mathbb{C}^n} = \lambda \|u\|_2^2.$$

$$(Au,u)_{\mathbb{C}^n} = \left(u, A^T u\right)_{\mathbb{C}^n} = (u, Au)_{\mathbb{C}^n} = \overline{(Au,u)}_{\mathbb{C}^n} \Rightarrow (Au,u)_{\mathbb{C}^n} \in \mathbb{R}$$

$$\lambda = \frac{(Au,u)_{\mathbb{C}^n}}{\|u\|_2^2} \in \mathbb{R}.$$

Propoziţie

Fie $A \in \mathbb{R}^{n\times n}$, $A = A^T$. Atunci există o bază ortonormată de vectori proprii ai matricei A, $\{u^1, u^2, ..., u^n\}$:

$$\left(u^i, u^j\right)_{\mathbb{R}^n} = \delta_{ij} = \begin{cases} 1 & \text{dacă } i = j \\ 0 & \text{dacă } i \neq j \end{cases}$$

Echivalent, putem scrie ca există vectori proprii $\{u^1, u^2, ..., u^n\}$ asociaţi valorilor proprii reale $\{\lambda_1, \lambda_2, ..., \lambda_n\}$ atfel ca:

$$AU = U\Lambda \quad \Leftrightarrow \quad U^T AU = \Lambda$$

cu $\Lambda = \text{diag}[\lambda_1, \lambda_2, ..., \lambda_n]$ şi $U = [u^1\, u^2 \cdots u^n]$ matrice ortogonală.

Definiţie

Se numeşte *coeficient Rayleigh* al vectorului $\boldsymbol{u \in \mathbb{R}^n}$ pentru matricea $\boldsymbol{A}$ următoarea mărime scalară:

$$\boldsymbol{r(u) = \frac{u^T Au}{u^T u} = \frac{\left(Au, u\right)_{\mathbb{R}^n}}{\left(u, u\right)_{\mathbb{R}^n}} = \frac{\left(Au, u\right)_{\mathbb{R}^n}}{\| u \|_2^2}}$$

Se verifică uşor că dacă $\boldsymbol{u \in \mathbb{R}^n}$ este vector propriu al matricei $\boldsymbol{A}$ asociat valorii proprii $\boldsymbol{\lambda}$ atunci $\boldsymbol{r(u) = \lambda}$.

Fie $\boldsymbol{A \in \mathbb{R}^{n \times n}, A = A^T}$. Matricea are valori proprii reale $\boldsymbol{\lambda_1, \lambda_2, \ldots, \lambda_n}$. Presupunem în plus că:

$$\boldsymbol{|\lambda_1| > |\lambda_2| \geq \cdots |\lambda_n| \geq 0}$$

Metoda puterii este un algoritm care aproximează valoarea proprie de modul maxim λ_1 şi un vector propriu asociat.

Se porneşte de la un vector nenul de normă euclidiană 1, $u^{(0)} \in \mathbb{R}^n$, $\| u^{(0)} \|_2 = 1$ şi se construieşte următorul şir de vectori de normă euclidiană *1*:

$$u^{(0)},\ u^{(1)} = \frac{1}{\| Au^{(0)} \|_2} Au^{(0)},\ \ u^{(2)} = \frac{1}{\| Au^{(1)} \|_2} Au^{(1)}, \ldots,$$

$$u^{(k)} = \frac{1}{\| Au^{(k-1)} \|_2} Au^{(k-1)}, \ldots$$

În anumite condiţii acest şir converge la un vector propriu asociat valorii proprii λ_1, iar coeficienţii Rayleigh corespunzători converg către λ_1.

Teoremă

Fie $A \in \mathbb{R}^{n \times n}, A = A^T$ o matrice simetrică pentru care valorile proprii îndeplinesc condiţia:

$$|\lambda_1| > |\lambda_2| \geq \cdots |\lambda_n| \geq 0.$$

Dacă $u^{(0)} \in \mathbb{R}^n$, $\| u^{(0)} \|_2 = 1$, $(u^{(0)}, u^1)_{\mathbb{R}^n} \neq 0$ (u^1 vector propriu asociat lui λ_1) atunci:

$$u^{(k)} = \frac{1}{\| A^k u^{(0)} \|_2} A^k u^{(0)} \rightarrow u^1 \text{ (vector propriu asociat lui } \lambda_1)$$

$$r(u^{(k)}) \rightarrow \lambda_1$$

Demonstraţie. Fie $\{u^1, u^2, \ldots, u^n\}$ vectori proprii asociaţi valorilor proprii $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ care formează o bază ortonormată în $\mathbb{R}^n$. Avem:

$$u^{(0)} = a_1 u^1 + a_2 u^2 + \cdots + a_n u^n \ , \ a_i \in \mathbb{R}$$

Deoarece $(u^{(0)}, u^1)_{\mathbb{R}^n} \neq 0$ rezultă că $a_1 \neq 0$. Din construcţia şirului $u^{(k)}$ deducem că există o constantă c_k astfel ca:

$$u^{(k)} = c_k A^k u^{(0)} =$$

$$= c_k A^k (a_1 u^1 + a_2 u^2 + \cdots + a_n u^n) =$$

$$= c_k (a_1 \lambda_1^k u^1 + a_2 \lambda_2^k u^2 + \cdots + a_n \lambda_n^k u^n) =$$

$$= c_k \lambda_1^k \left[a_1 u^1 + a_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k u^2 + \cdots + a_n \left(\frac{\lambda_n}{\lambda_1} \right)^k u^n \right]$$

Din această ultimă relaţie, din faptul că λ_1 este valoare proprie dominantă şi $a_1 \neq 0$ deducem că pentru k suficient de mare vectorul $u^{(k)}$ se aliniază după vectorul propriu u^1:

$$u^{(k)} \approx c_k \lambda_1^k a_1 u^1$$

Metoda puterii

$u^{(0)} \in \mathbb{R}^n$, $\| u^{(0)} \|_2 = 1$;

$k = 0$;

do

- $k++$;
- $w = Au^{(k-1)}$;
- $u^{(k)} = \frac{1}{\| w \|_2} w$;
- $\lambda_k = r(u^{(k)}) = \left(Au^{(k)}, u^{(k)} \right)_{\mathbb{R}^n}$;

while ($\| Au^{(k)} - \lambda_k u^{(k)} \| > \varepsilon$ şi $k \leq k_{\max}$);

Metoda iteraţiei inverse

Considerăm o matrice simetrică $\boldsymbol{A \in \mathbb{R}^{n\times n}}$, $\boldsymbol{A = A^T}$ şi $\boldsymbol{\mu \in \mathbb{R}}$ un număr real care nu este valoare proprie a matricei $\boldsymbol{A}$. Vom folosi metoda puterii pentru a aproxima valoarea proprie a matricei $\boldsymbol{A}$ care este cea mai apropiată de $\boldsymbol{\mu}$ şi un vector propriu asociat.

$\boldsymbol{\mu \neq}$ valoare proprie $\boldsymbol{\rightarrow}$ $\boldsymbol{\det(A - \mu I_n) \neq 0 \rightarrow \exists\,(A - \mu I_n)^{-1}}$

Fie $\boldsymbol{\{\lambda_1, \lambda_2, \ldots, \lambda_n\}}$ valorile proprii reale ale matricei $\boldsymbol{A}$.

Valorile proprii ale matricei $(A-\mu I_n)^{-1}$ sunt:

$$\left\{\frac{1}{(\lambda_1-\mu)},\frac{1}{(\lambda_2-\mu)},\cdots,\frac{1}{(\lambda_n-\mu)}\right\}$$

Matricele A şi $(A-\mu I_n)^{-1}$ au aceiași vectori proprii. Să presupunem că λ_I este valoarea proprie cea mai apropiată de μ (şi singura). Atunci:

$$\frac{1}{|\lambda_I-\mu|} > \frac{1}{|\lambda_j-\mu|} \quad \forall\, j \neq I$$

Această relaţie sugerează ideea aplicării metodei puterii matricei $(A - \mu I_n)^{-1}$ pentru a aproxima valoarea proprie $(\lambda_I - \mu)^{-1}$ şi a unui vector propriu asociat. Algoritmul duce la aproximarea valorii proprii cea mai apropiată de μ, λ_I şi a unui vector propriu asociat acestei autovalori, u^I.

Metoda iteraţiei inverse

$u^{(0)} \in \mathbb{R}^n$, $\| u^{(0)} \|_2 = 1$;

$k = 0$;

do

- $k++$;
- Se rezolvă sistemul $(A - \mu I_n)w = u^{(k-1)}$;
- $u^{(k)} = \dfrac{1}{\| w \|_2} w$;
- $\lambda_k = r(u^{(k)}) = \left(Au^{(k)}, u^{(k)} \right)_{\mathbb{R}^n}$;

while ($\| Au^{(k)} - \lambda_k u^{(k)} \| > \varepsilon$ şi $k \le k_{\max}$);

Calcul Numeric

Cursul 8

2022

Anca Ignat

Forma superioară Hessenberg

Spunem că o matrice $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ este în ***formă superioară Hessenberg*** dacă:

$$\boldsymbol{h_{ij} = 0\,,\, i = 1,\ldots,n\,,\, j = 1,\ldots,i-2}$$

O matrice în formă Hessenberg arată astfel:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \cdots h_{1n-1} & h_{1n} \\ h_{21} & h_{22} & h_{23} \cdots h_{2n-1} & h_{2n} \\ 0 & h_{32} & h_{33} \cdots h_{3n-1} & h_{3n} \\ 0 & 0 & h_{43} \cdots h_{4n-1} & h_{4n} \\ \vdots & & & \\ 0 & 0 & 0 \cdots h_{nn-1} & h_{nn} \end{pmatrix}$$

Ne interesează un algoritm care să transforme o matrice pătratică $\boldsymbol{A}$ oarecare într-o matrice Hessenberg superioară $\boldsymbol{H}$ care să aibă aceleaşi valori proprii:

$A \rightarrow H$ **a.î.** $\mathbf{H} \sim A$, $H = \tilde{P} A \tilde{P}^{-1}$, $\tilde{P}$ **matrice nesingulară**

Algoritmul este o adaptare a algoritmului lui Housholder şi se desfăşoară în ***(n-2)*** paşi, folosind matricile de reflexie pentru a transforma matricea.

Pas 1

se efectuează operaţiile $A = P_1 A P_1$ (matricea P_1 se alege astfel încât coloana 1 să fie transformată în formă superior Hessenberg)

Pas 2

$$A = P_2 A P_2 = P_2 (P_1 A^{init}) P_2$$

(P_2 transformă coloana 2 în formă superior Hessenberg fără să schimbe coloana 1)

Pas *r*

$$A = P_r A P_r = P_r (P_{r-1} \cdots P_1 A^{init} P_1 \cdots P_{r-1}) P_r$$

(se transformă coloana r în formă superior Hessenberg fără să schimbe primele $(r-1)$ coloane)

Pasul r ($r=1,2,\ldots,n-2$)

La intrarea în pasul r matricea A are primele $(r-1)$ coloane în formă superior Hessenberg. La ieşirea din pasul r matricea A va avea primele r coloane în formă superior Hessenberg:

$$A_{ies} = P_r \, A_{intr} \, P_r \;\; , \;\; A_{ies} \sim A_{intr}$$

$$P_r = I_n - 2v^r (v^r)^T \;\; , \;\; v^r \in R^n \;\; , \;\; \| v^r \|_2 = 1$$

Vectorul v^r se alege astfel ca matricea A_{ies} să aibă coloana r în formă superior Hessenberg şi să nu schimbe primele $(r-1)$ coloane ale matricii A_{intr} .

Calculul matricii P_r

$$P = I_n - \frac{1}{\beta} uu^T$$

$$\beta = \sigma - k \cdot a_{r+1r}$$

$$k^2 = \sigma = a_{r+1r}^2 + \cdots + a_{ir}^2 + \cdots + a_{nr}^2 = \sum_{i=r+1}^{n} a_{ir}^2 \quad \Rightarrow \quad k = \pm\sqrt{\sigma}$$

$$\operatorname{semn} k = -\operatorname{semn} a_{r+1r}$$

$$u := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{r+1r} - k \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix}$$

$$\beta = 0 \rightarrow \; r = r + 1 \;\; (P = I_n)$$

Algoritmul de trecere de la matricea $\boldsymbol{A}$ la matricea $\boldsymbol{P_r A}$ este următorul:

$$(P_r A)e_j = \begin{cases} Ae_j & \text{pentru } j = 1,\ldots,r-1 \\ (a_{1r}, a_{2r}, \ldots, a_{rr}, k, 0, \ldots, 0)^T & \text{pentru } j = r \\ Ae_j - \dfrac{\gamma_j}{\beta} u & \text{pentru } j = r+1,\ldots,n \end{cases}$$

$$\gamma_j = (Ae_j, u)_{\mathbb{R}^n} = \sum_{i=r+1}^{n} u_i a_{ij}$$

$$u_i = 0, \quad i = 1,\ldots,r \; , \; u_{r+1} = a_{r+1r} - k \; , \; u_i = a_{ir} \; , \quad i = r+2,\ldots,n$$

Vom descrie în continuare cum se efectuează operaţia $\boldsymbol{A:=AP_r}$ fără a face înmulţire matricială (matricea $\boldsymbol{A}$ este cea obţinută mai sus având primele $\boldsymbol{r}$ coloane în formă superior Hessenberg).

Vom arata că această operaţie nu schimbă forma superior Hessenberg obţinută. Vom pune în evidenţă transformările liniilor matricii $\boldsymbol{A}$. Pentru $\boldsymbol{i=1,...,n}$ avem:

$$\boldsymbol{e_i^T(AP)} = \text{ noua linie } \boldsymbol{i} \text{ a matricii } \boldsymbol{AP} = (\boldsymbol{e_i^T A})(\boldsymbol{I_n} - \frac{\mathbf{1}}{\boldsymbol{\beta}}\boldsymbol{uu^T}) =$$

$$= \boldsymbol{e_i^T A} - \frac{\mathbf{1}}{\boldsymbol{\beta}}(\boldsymbol{e_i^T A})\boldsymbol{u}\,\boldsymbol{u^T} = \boldsymbol{e_i^T A} - \frac{\boldsymbol{\gamma_i}}{\boldsymbol{\beta}}\boldsymbol{u^T}$$

unde

$$\gamma_i = (e_i^T A)u = a_{ir+1}u_{r+1} + \cdots + a_{in}u_n$$

Elementele liniei i se schimbă astfel:

$$a_{ij} = a_{ij} - \frac{\gamma_i}{\beta}u_j \ , \ j = r+1,\ldots,n \ , \quad i = 1,\ldots,n$$

Operaţia $A{:=}AP_r$ nu modifică primele r coloane ale matricii A, ele rămânând în formă superior Hessenberg.

Algoritmul de obţinere a formei superior Hessenberg

for $r = 1,\ldots,n-2$

// construcţia matricii P_r – constanta β şi vectorul u

- $\sigma = \sum_{i=r+1}^{n} a_{ir}^2;$
- if $(\sigma \le \varepsilon)$ break ; // $r = r+1 \leftrightarrow P_r = I_n$
- $k = \sqrt{\sigma};$
- if $(a_{r+1r} > 0)\ k = -k;$
- $\beta = \sigma - k\, a_{r+1r};$
- $u_{r+1} = a_{r+1r} - k;\ u_i = a_{ir}\ , i = r+2,\ldots,n;$

// $A = P_r * A$

// transformarea coloanelor $j = r+1, \ldots, n$

- for $j = r+1, \ldots, n$

$$\begin{cases} \gamma = (\gamma_j / \beta) = (Ae_j, u) / \beta = (\sum_{i=r+1}^{n} u_i a_{ij}) / \beta; \\ \text{for } i = r+1, \ldots, n \\ \quad a_{ij} = a_{ij} - \gamma * u_i; \end{cases}$$

// transformarea coloanei r a matricii A

- $a_{r+1r} = k$; $a_{ir} = 0$, $i = r+2, \ldots, n$;

// $A = A * P_r$

// transformarea liniilor $i = 1, \ldots, n$

- for $i = 1, \ldots, n$

$$\begin{cases} \gamma = (\gamma_i / \beta) = ((e_i^T A)u) / \beta = (\sum_{j=r+1}^{n} u_j a_{ij}) / \beta; \\ \text{for } j = r+1, \ldots, n \\ \quad a_{ij} = a_{ij} - \gamma * u_j; \end{cases}$$

Algoritmul *QR* de aproximare a valorilor proprii ale unei matrice oarecare

Prezentăm în continuare cel mai folosit algoritm de aproximare a valorilor proprii pentru matrice pătratice oarecare.

Spunem că o matrice $\boldsymbol{S} \in \mathbb{R}^{n \times n}$ este în ***formă Schur reală*** dacă matricea $\boldsymbol{S}$ este în formă superior Hessenberg şi în plus este bloc-diagonală:

$$S = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ 0 & S_{22} & \cdots & S_{2p} \\ \vdots & & & \\ 0 & 0 & \cdots & S_{pp} \end{pmatrix}$$

blocurile S_{ii} sunt astfel ca:

- $S_{ii} \in \mathbb{R}$ - este valoare proprie reală
- $S_{ii} \in \mathbb{R}^{2\times 2}$ - este bloc corespunzător valorilor proprii complexe

Valorile proprii corespunzătoare blocului $S_{ii} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbb{R}^{2\times 2}$ sunt rădăcinile ecuaţiei:

$$\begin{vmatrix} \lambda - a & -b \\ -c & \lambda - d \end{vmatrix} = (\lambda - a)(\lambda - d) - bc = \lambda^2 - (a+d)\lambda + ad - bc = 0$$

Se presupune că această ecuaţie de gardul 2 are rădăcini complexe.

Algoritmul QR de aproximare a valorilor proprii construieşte un şir de matrici $A^{(k)} \in \mathbb{R}^{n \times n}$, matrici asemenea cu matricea A, $A^{(k)} \sim A, \forall k$, şir care converge la o matrice în formă Schur reală, $A^{(k)} \to S, k \to \infty$. Matricea limită S este asemenea cu matricea A, valorile prorii ale matricii S fiind uşor de calculat. Şirul $A^{(k)}$ se construieşte astfel:

$A^{(0)} := A$, $A^{(0)} = Q_0 R_0$ (descomp. QR calc. pentru matricea $A^{(0)}$)

$A^{(1)} := R_0 Q_0$, $A^{(1)} = Q_1 R_1$ (descomp. QR calc. pentru matricea $A^{(1)}$)

$A^{(2)} := R_1 Q_1$

$\vdots$

$A^{(k)} = Q_k R_k$ (descomp. QR calc. pentru matricea $A^{(k)}$),

$A^{(k+1)} := R_k Q_k$, $k = 0,1,2,\ldots$

Matricile Q_k sunt matrici ortogonale ($Q_k^{-1} = Q_k^T$) iar matricile R_k sunt superior triunghiulare.

Matricile $A^{(k)}$ şi $A^{(k+1)}$ sunt asemenea:

$$Q_k^T * \mid A^{(k)} = Q_k R_k \Rightarrow R_k = Q_k^T A^{(k)}$$

$$A^{(k+1)} = R_k Q_k = Q_k^T A^{(k)} Q_k \quad \Rightarrow A^{(k+1)} \sim A^{(k)}, \forall k$$

Matricile şirului construit sunt toate asemenea prin urmare au aceleaşi valori proprii anume cele ale matricii iniţiale $A = A^{(0)}$:

$$A = A^{(0)} \sim A^{(1)} \sim \cdots \sim A^{(k)} \sim \cdots \sim S$$

Dacă matricea $A^{(k)}$ este în formă superioară Hessenberg, atunci descompunerea QR realizată cu algoritmul lui Givens se simplifică. Reamintim algoritmul lui Givens:

$$\boldsymbol{R_{n-1n}(\theta_{n-1n})\cdots R_{pn}(\theta_{pn})\cdots R_{pp+1}(\theta_{pp+1})\cdots R_{1n}(\theta_{1n})\cdots R_{12}(\theta_{12})\,A = R}$$

Dacă matricea $\boldsymbol{A}$ este în formă Hessenberg în algoritmul lui Givens, din cele $\dfrac{\boldsymbol{n(n-1)}}{\boldsymbol{2}}$ înmulţiri cu matrici de rotaţie rămân doar ***(n-1)***:

$$\boldsymbol{R_{n-1n}(\theta_{n-1n})\cdots R_{pp+1}(\theta_{pp+1})\cdots R_{23}(\theta_{23})R_{12}(\theta_{12})\,A = R}.$$

Problema care se pune este dacă pornind cu o matrice în formă Hessenberg, toate matricile şirului rămân în formă Hessenberg:

$\boldsymbol{A^{(k)}}$ **(**în formă Hessenberg**)** $\boldsymbol{= H = QR}$ **(**cu Givens**)** $\boldsymbol{\Rightarrow ?}$

$\boldsymbol{A^{(k+1)} = \bar{H} = RQ = Q^T A^{(k)} Q = Q^T HQ}$ – este tot în formă Hessenberg ?

Avem:

$$\boldsymbol{\bar{H} = Q^T HQ = R\, R_{12}^T(\theta_{12}) \cdots R_{rr+1}^T(\theta_{rr+1}) \cdots R_{n-1n}^T(\theta_{n-1n})}$$

Notăm cu:

$$\boldsymbol{\bar{R} = R\, R_{12}^T(\theta_{12})}$$

pentru care avem:

$$\begin{cases} \overline{r}_{i1} = cr_{i1} + sr_{i2} \ , \forall i \\ \overline{r}_{i2} = -sr_{i1} + cr_{i2} \ , \forall i \end{cases} + \begin{cases} r_{i1} = 0, i = 2,\ldots,n \\ r_{i2} = 0, i = 3,\ldots,n \end{cases} \Rightarrow \begin{cases} \overline{r}_{i1} = 0, i = 3,\ldots,n \\ \overline{r}_{i2} = 0, i = 3,\ldots,n \end{cases}$$

deci coloana ***1*** se transformă în formă Hessenberg iar coloana ***2*** rămâne în formă suprior triunghiulară.

La pasul p avem:

$$\left(R\, R_{12}^T(\theta_{12}) \cdots R_{p-1\,p}^T(\theta_{p-1\,p}) \right) R_{pp+1}^T(\theta_{pp+1}) = \tilde{R}\, R_{pp+1}^T(\theta_{pp+1}) = \overline{R} \ ,$$

$$\tilde{R} = R\, R_{12}^T(\theta_{12}) \cdots R_{p-1\,p}^T(\theta_{p-1\,p})$$

matricea $\tilde{R}$ are primele ***(p-1)*** coloane în formă Hessenberg iar restul coloanelor sunt în formă superior triunghiulară. Vom arata că la acest pas matricea $\bar{R}$ va avea primele p coloane în formă Hessenberg iar restul coloanelor în formă superior triunghiulară. Operaţia $\bar{R} := \tilde{R}\, R^{T}_{pp+1}(\theta_{pp+1})$ presupune doar schimbarea elementelor coloanelor p şi $p+1$:

$$\begin{cases} \bar{r}_{ip} = c\tilde{r}_{ip} + s\tilde{r}_{ip+1} \ , \forall i \\ \bar{r}_{ip+1} = -s\tilde{r}_{ip} + c\tilde{r}_{ip+1} \ , \forall i \end{cases} + \begin{cases} \tilde{r}_{ip} = 0, \ i = p+1,\ldots,n \\ \tilde{r}_{ip+1} = 0, \ i = p+2,\ldots,n \end{cases} \Rightarrow$$

.

$$\begin{cases} \bar{r}_{ip} = 0, \ i = p+2,\ldots,n \\ \bar{r}_{ip+1} = 0, \ i = p+2,\ldots,n \end{cases}$$

Observăm din relaţia de mai sus că în matricea $\bar{\boldsymbol{R}}$ coloana $\boldsymbol{p}$ are formă Hessenberg iar coloana $\boldsymbol{p+1}$ rămâne în formă superior triunghiulară (celelalte elemente din matrice nu se modifică).

Prin urmare după pasul $\boldsymbol{n-1}$ matricea $\bar{\boldsymbol{H}} = \boldsymbol{A}^{(k+1)}$ este în formă superioară Hessenberg. Algoritmul $\boldsymbol{QR}$ de aproximare a valorilor proprii folosind descompunerea Givens păstrează forma Hessenberg.

Algoritmul QR pentru valori proprii

/ / se aduce matricea $\boldsymbol{A}$ la forma Hessenberg

- $\boldsymbol{A = \bar{Q}\, A\, \bar{Q}^T;}$
- $\boldsymbol{k = 0;}$
- while **(** $\boldsymbol{A \neq}$ forma Schur reală **)**

  - $\boldsymbol{A = QR;}$ / / se calculează cu algoritmul Givens
  - $\boldsymbol{A = RQ}$ sau $\boldsymbol{Q^T AQ;}$
  - $\boldsymbol{k = k + 1;}$

În practică se presupune că matricea $\boldsymbol{A}$ este în **formă Hessenberg neredusă**, adică:

$$a_{ii-1} \neq 0 \;\; \forall i = 2,\ldots,n$$

Dacă matricea nu este în formă neredusă, problema se decuplează:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix} \quad , \; p = n-1 \text{ sau } n-2$$

$$\begin{matrix} p & n-p \end{matrix}$$

Algoritmului *QR* cu deplasare ("*shift*") simplă

Algoritmul cu deplasare simplă este următorul:

- $\boldsymbol{A = \bar{Q}\, A\, \bar{Q}^T}$**;** / / aducerea la forma Hessenberg neredusă
- $\boldsymbol{k = 0;}$
- while **(** $\boldsymbol{A \neq}$ forma Schur reală **)**

  - $\boldsymbol{A - d_k\ I_n = QR;}$ **/ /** se calc**.** cu alg. Givens
  - $\boldsymbol{A := RQ + d_k\ I_n;}$
  - $\boldsymbol{k = k + 1;}$

$d_k \in \mathbb{R}$ sunt constantele de deplasare.

Dacă $A - d\,I_n = QR$ ($A^{(k)}$) şi $\overline{A} = RQ + d\,I_n$ ($A^{(k+1)}$), se pune problema dacă cele două matrice sunt asemenea ($A \sim \overline{A}$) (şirul de matrice construit cu pasul QR cu deplasare simplă au aceleaşi valori proprii).

$$\overline{A} = Q^T QRQ + d\,Q^T Q = Q^T (QR + d\,I_n)Q = Q^T AQ \Rightarrow \overline{A} \sim A$$

Varianta cu deplasare se efectuează pentru a accelera convergenţa algoritmului. Dacă $\lambda_1, \lambda_2, \ldots, \lambda_n$ sunt valorile proprii ale matricii A ordonate astfel ca:

$$|\lambda_1 - d| \geq |\lambda_2 - d| \geq \cdots \geq |\lambda_n - d|$$

Rapiditatea cu care $a_{p+1\,p}^{(k)} \to 0\,,k \to \infty$ este dată de rata de convergenţa a expresiei $\left|\frac{\lambda_{p+1}-d}{\lambda_p-d}\right|^k$. Dacă se alege $d \approx \lambda_n$ convergenţa $a_{n-1\,n}^{(k)} \to 0$ este rapidă. Avem următorul rezultat:

Teoremă

Fie d o valoare proprie a unei matrice Hessenberg nereduse H. Dacă $\overline{H} = RQ + d\,I_n$, cu $H - d\,I_n = QR$ descompunerea QR a matricei $H - d\,I_n = QR$. Atunci:

$$\overline{h}_{nn-1} = 0\ ,\ \overline{h}_{nn} = d$$

Algoritmul ***QR*** cu deplasare simplă găseşte valoarea proprie ***d*** într-un singur pas. Euristic s-a constatat că la fiecare pas, cea mai bună aproximare a unei valori proprii este $a_{nn}^{(k)}$.

$$d_k = a_{nn}^{(k)}$$

Algoritmul QR cu deplasare simplă

- $A = \bar{Q}\, A\, \bar{Q}^T$; / / aducerea la forma Hessenberg neredusă
- $k = 0$;
- while ($A \neq$ forma Schur reală)

  - $A - a_{nn}\, I_n = QR$; / / se calc. cu algoritmul Givens
  - $A := RQ + a_{nn}\, I_n$;
  - $k = k + 1$;

Algoritmului *QR* cu deplasare (“*shift*”) dublă

În cazul când valorile proprii a_1, a_2 corespunzătoare blocului:

$$G = \begin{bmatrix} a_{pp} & a_{pn} \\ a_{np} & a_{nn} \end{bmatrix}, \quad p = n-1$$

sunt complexe, $a_1, a_2 \in \mathbb{C}$, abordarea cu deplasare simplă nu mai asigură accelerarea convergenţei. Avem:

$$\det(\lambda I_2 - G) = (\lambda - a_1)(\lambda - a_2) = (\lambda - a_{pp})(\lambda - a_{nn}) - a_{pn}a_{np} =$$

$$= \lambda^2 - (a_1 + a_2)\lambda + a_1 a_2 = \lambda^2 - (a_{pp} + a_{nn})\lambda + a_{pp}a_{nn} - a_{pn}a_{np}$$

$$a_1 + a_2 = a_{pp} + a_{nn} = trace(G) \quad , \quad a_1 a_2 = a_{pp}a_{nn} - a_{pn}a_{np} = \det G$$

Algoritmul ***QR*** cu deplasare dublă constă în trecerea de la matricea $A = A^{(k)}$ la matricea $A_2 = A^{(k+1)}$ realizând doi paşi cu deplasare simplă :

$A \rightarrow A_1$ (deplasare simplă a_1), $A_1 \rightarrow A_2$ (deplasare simplă a_2)

$$A - a_1 I_n = Q_1 R_1$$

$$A_1 = R_1 Q_1 + a_1 I_n$$

$$A_1 - a_2 I_n = Q_2 R_2$$

$$A_2 = R_2 Q_2 + a_2 I_n$$

Fie matricea :

$$M := (Q_1 Q_2)(R_2 R_1) = Q_1 (Q_2 R_2) R_1 = Q_1 (A_1 - a_2\ I_n) R_1 =$$

$$= Q_1 (Q_1^T A Q_1 - a_2\ I_n) R_1 = Q_1 Q_1^T A Q_1 R_1 - a_2 Q_1 R_1 =$$

$$= (A - a_2\ I_n) Q_1 R_1 = (A - a_2\ I_n)(A - a_1\ I_n)$$

$$M = (Q_1 Q_2)(R_2 R_1) = (A - a_2\ I_n)(A - a_1\ I_n) =$$

$$= A^2 - (a_1 + a_2) A + a_1 a_2\ I_n$$

Avem următoarele relaţii de asemănare:

$$A \sim A_1 = Q_1^T A Q_1 \sim A_2 = Q_2^T A_1 Q_2 = Q_2^T Q_1^T A Q_1 Q_2 = \left(Q_1 Q_2\right)^T A \left(Q_1 Q_2\right)$$

$$A_2 = \left(Q_1 Q_2\right)^T A \left(Q_1 Q_2\right) = Q^T A Q \quad , \quad Q := Q_1 Q_2$$

Matricea $\boldsymbol{Q}$ care asigură trecerea de la matricea $\boldsymbol{A}$ la matricea $\boldsymbol{A_2}$ este matricea ortogonală din descompunerea $\boldsymbol{QR}$ a matricii $\boldsymbol{M = \left(A - a_2\, I_n\right)\left(A - a_1\, I_n\right)}$. Pasul $\boldsymbol{QR}$ cu deplasare dublă se face urmând etapele:

1) se calculează matricea $\boldsymbol{M = A^2 - s\,A + qI_n}$ cu

$\boldsymbol{s = a_1 + a_2 = a_{pp} + a_{nn}}$, $\boldsymbol{q = a_1 a_2 = a_{pp}\,a_{nn} - a_{pn} a_{np}}$;

2) se calculează descompunerea $\boldsymbol{QR}$ a matricii $\boldsymbol{M}$;

3) $\boldsymbol{A_2 := Q^T A Q}$.

Vectori proprii

Considerăm două matrici asemenea $\boldsymbol{A}$ şi $\boldsymbol{B}$:

$$\boldsymbol{A \sim B} \quad \Leftrightarrow \quad \boldsymbol{A = PBP^{-1}} \;\; , \;\; \boldsymbol{P} \quad \text{matrice nesingulară}$$

Ştim că cele două matrici au acelaşi polinom caracteristic, $\boldsymbol{p_A(\lambda) \equiv p_B(\lambda)}$, deci au aceleaşi valori proprii. Ne interesează care este legătura între vectorii proprii asociaţi aceleiaşi valori proprii. Fie $\boldsymbol{u}$ vector propriu asociat valorii proprii $\boldsymbol{\lambda}$ pentru matricea $\boldsymbol{A}$ şi $\boldsymbol{w}$ vector propriu asociat valorii proprii $\boldsymbol{\lambda}$ pentru matricea $\boldsymbol{B}$. Care este relaţia între $\boldsymbol{u}$ şi $\boldsymbol{w}$?

$$Au = \lambda u\,,\; Bw = \lambda w\,,\;\; A = PBP^{-1} \;\Rightarrow\; PBP^{-1}u = \lambda u \;\Rightarrow$$

$$BP^{-1}u = \lambda\, P^{-1}u \;\Rightarrow\; w = P^{-1}u \;\;,\; u = Pw$$

Dacă se aplică algoritmul ***QR*** unei matrici simetrice, forma

Schur reală la care se ajunge este o matrice diagonală:

$$S \;=\; \Lambda = \mathbf{diag}[\lambda_1, \lambda_2, ..., \lambda_n]$$

Legătura dintre matricea simetrică iniţială *A* şi matricea diagonală este de forma:

$$S = \Lambda = \mathbf{diag}[\lambda_1, \lambda_2, ..., \lambda_n] = U^T A\, U$$

unde ***U*** este o matrice ortogonală, coloanele matricii ***U*** fiind vectori proprii asociaţi valorilor proprii reale $\lambda_1, \lambda_2, ..., \lambda_n$.
Matricea ***U*** se poate calcula astfel:

Algoritmul QR pentru matrici simetrice (valori +vectori proprii)

// **se aduce matricea** A **la forma Hessenberg**

- $A = \bar{Q}\, A\, \bar{Q}^T$;
- $U = \bar{Q}^T$;
- $k = 0$;
- **while (** $A \neq$ **matrice diagonală** **)**

  - $A = QR$; // **se calculează cu algoritmul Givens**
  - $A = RQ$ **sau** Q^T **AQ;**
  - $U = UQ$;
  - $k = k + 1$;

Descompunerea după valori singulare
(Singular Value Decomposition)

Teoremă

Fie $A \in \mathbb{R}^{m \times n}$. Atunci există o matrice ortogonală pătratică de dimensiune m, $U \in \mathbb{R}^{m \times m}$, o matrice ortogonală pătratică de dimensiune n, $V \in \mathbb{R}^{n \times n}$ şi constantele pozitive:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0, \qquad r \leq \min\{m,n\} \text{ a.î.}$$

$$A = U \Sigma V^T, \quad \Sigma \in \mathbb{R}^{m \times n}, \Sigma = \begin{bmatrix} D & \mathbf{0}_{r \times (n-r)} \\ \mathbf{0}_{(m-r) \times r} & \mathbf{0}_{(m-r) \times (n-r)} \end{bmatrix}, \qquad (1)$$

$$D \in \mathbb{R}^{r \times r}, D = \mathbf{diag}\left[\sigma_1, \ldots, \sigma_r\right]$$

Constanta r este chiar rangul matricii A, $r = \mathbf{rang}(A)$.
Constantele $\sigma_1, \sigma_2, \ldots, \sigma_r$ poartă numele de ***valori singulare*** ale matricii A.
Folosind relaţia (1) avem:

$$A^T = \left(U\Sigma V^T\right)^T = V\Sigma^T U^T,$$

$$AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T = U\Lambda_m U^T \ ,$$

$$\Lambda_m = \Sigma\Sigma^T = \begin{bmatrix} D^2 & \mathbf{0}_{r\times(m-r)} \\ \mathbf{0}_{(m-r)\times r} & \mathbf{0}_{(m-r)\times(m-r)} \end{bmatrix} \in \mathbb{R}^{m\times m}$$

$$A^T A = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T\Sigma V^T = V\Lambda_n V^T \ ,$$

$$\Lambda_n = \Sigma^T\Sigma = \begin{bmatrix} D^2 & \mathbf{0}_{r\times(n-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(n-r)} \end{bmatrix} \in \mathbb{R}^{n\times n}$$

Ţinând cont de ortogonalitatea matricilor U şi V, putem rescrie relaţiile de mai sus astfel:

$$(AA^T)U = U\,\Lambda_m \ , \ \Lambda_m = \mathbf{diag}\left[\sigma_1^2,\sigma_2^2,\ldots,\sigma_r^2,0,\ldots,0\right] \in \mathbb{R}^{m\times m}$$

$$(A^T A)V = V\,\Lambda_n \ , \ \Lambda_n = \mathbf{diag}\left[\sigma_1^2,\sigma_2^2,\ldots,\sigma_r^2,0,\ldots,0\right] \in \mathbb{R}^{n\times n}$$

Din aceste relaţii deducem că $\boldsymbol{\sigma_1^2, \sigma_2^2, \ldots, \sigma_r^2}$ sunt valorile proprii strict pozitive ale matricilor $\boldsymbol{AA^T}$ şi/sau $\boldsymbol{A^TA}$ iar matricile $\boldsymbol{U}$ şi $\boldsymbol{V}$ sunt matrici ale căror coloane sunt vectorii proprii asociaţi (cei ce formează baze ortonormate). Matricile $\boldsymbol{AA^T}$ şi $\boldsymbol{A^TA}$ sunt matrici simetrice:

$$\left(AA^T\right)^T = \left(A^T\right)^T A^T = AA^T \quad , \quad \left(A^TA\right)^T = A^T\left(A^T\right)^T = A^TA$$

şi au toate valorile proprii nenegative:

$$\left(AA^T\right)u=\lambda u\Rightarrow\left(AA^Tu,u\right)=\left(\lambda u,u\right)\Rightarrow$$

$$\lambda=\frac{\left(A^Tu,A^Tu,\right)}{\left(u,u\right)}=\frac{\left\|A^Tu\right\|_2^2}{\left\|u\right\|_2^2}\geq 0$$

Putem folosi descompunerea după valori singulare pentru a defini pseudo-inversa unei matrici oarecare $A\in\mathbb{R}^{m\times n}$ ($n\neq m$).

$$A=U\Sigma V^T\ ,\ A^{-1}=_?\left(U\Sigma V^T\right)^{-1}=\left(V^T\right)^{-1}\Sigma_?^{-1}U^{-1}=V\Sigma_?^{-1}U^T$$

Rămâne de definit matricea $\Sigma_?^{-1}$. Urmând acest raţionament se defineşte ***pseudoinversa Moore-Penrose*** a unei matrici

$\boldsymbol{A} \in \mathbb{R}^{m \times n}$ astfel:

$$\boldsymbol{A}^{\boldsymbol{I}} = \boldsymbol{V}\boldsymbol{\Sigma}^{\boldsymbol{I}}\boldsymbol{U}^{\boldsymbol{T}} \quad , \; \boldsymbol{A}^{\boldsymbol{I}} \in \mathbb{R}^{n \times m} \quad , \quad \boldsymbol{\Sigma}^{\boldsymbol{I}} = \begin{bmatrix} \boldsymbol{D}^{-1} & 0_{r \times (m-r)} \\ \mathbf{0}_{(n-r) \times r} & 0_{(n-r) \times (m-r)} \end{bmatrix} \in \mathbb{R}^{n \times m},$$

$$\boldsymbol{D}^{-1} \in \mathbb{R}^{r \times r}, \boldsymbol{D}^{-1} = \mathbf{diag}\left[\frac{\mathbf{1}}{\sigma_1}, \ldots, \frac{\mathbf{1}}{\sigma_r}\right].$$

Pseudoinversa definită mai sus satisface următoarele proprietăţi:

$$\left(\boldsymbol{A}^{\boldsymbol{I}}\right)^{\boldsymbol{I}} = \boldsymbol{A} \quad , \; \forall \boldsymbol{A} \in \mathbb{R}^{m \times n} \quad ; \; \left(\boldsymbol{A}^{\boldsymbol{T}}\right)^{\boldsymbol{I}} = \left(\boldsymbol{A}^{\boldsymbol{I}}\right)^{\boldsymbol{T}}, \; \forall \boldsymbol{A} \in \mathbb{R}^{m \times n}$$

$$\boldsymbol{A}\boldsymbol{A}^{\boldsymbol{I}}\boldsymbol{A} = \boldsymbol{A} \qquad , \qquad \boldsymbol{A}^{\boldsymbol{I}}\boldsymbol{A}\boldsymbol{A}^{\boldsymbol{I}} = \boldsymbol{A}^{\boldsymbol{I}}$$

Există o proprietate care nu mai este satisfăcută de psudoinversă deşi este respectată de inversa clasică:

$$\exists\, \boldsymbol{A, B} \ \textbf{a.î.} \quad \left(\boldsymbol{AB}\right)^{\boldsymbol{I}} \neq \boldsymbol{B}^{\boldsymbol{I}} \boldsymbol{A}^{\boldsymbol{I}}.$$

Descompunerea după valori singulare poate fi utilizată şi pentru rezolvarea sistemelor liniare cu matrici oarecare ($\boldsymbol{m \neq n}$)

$$\boldsymbol{Ax = b} \quad , \quad \boldsymbol{A} \in \mathbb{R}^{\boldsymbol{m \times n}} \quad , \quad \boldsymbol{b} \in \mathbb{R}^{\boldsymbol{m}} \quad , \quad \boldsymbol{x} := \boldsymbol{A}^{\boldsymbol{I}} \boldsymbol{b} \in \mathbb{R}^{\boldsymbol{n}}.$$

Problema celor mai mici pătrate

$$A \in \mathbb{R}^{m \times n} \ , \ b \in \mathbb{R}^{m} \ , \ Ax = b \ , \ x \in \mathbb{R}^{n} \qquad \textbf{(1)}$$

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
&\vdots \\
&a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i \\
&\vdots \\
&a_{m1}x_1 + a_{m2}x_2 \cdots + a_{mn}x_n = b_m
\end{aligned}$$

Sistemul are soluţii clasice dacă:

$$\textbf{rang } A = \textbf{rang } [A \mid b]$$

- $m < n$ - o infinitate de soluţii
- $m \geq n$
  - dacă **rang** A = **rang** $[A \mid b]$ soluţii clasice
  - dacă **rang** $A \neq$ **rang** $[A \mid b]$ soluţii în sensul celor mai mici pătrate

Vectorul reziduu:

$$r(x) = b - Ax \in \mathbb{R}^m$$

Vectorul $x \in \mathbb{R}^n$ se numeşte ***soluţie în sensul celor mai mici pătrate*** pentru sistemul (1) dacă este soluţia următoarei probleme de optimizare:

$$\min\{\|r(x)\|_2^2 = \|b - Ax\|_2^2;\ x \in \mathbb{R}^n\} \qquad (LSP)$$

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \in \mathbb{R}^{3\times 2} \quad , \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \ m = 3, \ n = 2$$

$$\text{rang } A = 2 \neq \text{rang } [A \mid b] = 3$$

Sistemul:

$$\begin{aligned} x_1 + 4x_2 &= 0 \\ 2x_1 + 5x_2 &= 0 \\ 3x_1 + 6x_2 &= 1 \end{aligned} \qquad (2)$$

nu are soluţie clasică (nu există x_1, x_2 care să satisfacă toate cele 3 ecuaţii simultan). Vectorul reziduu are forma:

$$r(x)=b-Ax=\begin{pmatrix}0\\0\\1\end{pmatrix}-\begin{bmatrix}1&4\\2&5\\3&6\end{bmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}=\begin{pmatrix}-x_1-4x_2\\-2x_1-5x_2\\1-3x_1-6x_2\end{pmatrix}$$

Soluţia în sensul celor mai mici pătrate a acestui sistem este definită ca soluţia problemei de optimizare:

$$\min\{(-x_1-4x_2)^2+(-2x_1-5x_2)^2+(1-3x_1-6x_2)^2;\ x_1,x_2\in\mathbb{R}\}$$

$$\min\{1-6x_1-12x_2+64x_1x_2+14x_1^2+77x_2^2;\ x_1,x_2\in\mathbb{R}\}$$

Această problemă de minimizare are soluţia:

$$x_1 = \frac{13}{18}, \qquad x_2 = -\frac{2}{9}, \qquad \| r(x) \|_2^2 = \frac{1}{6}$$

şi este soluţia în sensul celor mai mici pătrate a sistemului (2).

$$\mathbf{range}(\mathrm{A}) = \{ \mathrm{y} \in \mathbb{R}^m ; y = a_1 A^1 + a_2 A^2 + \cdots + a_n A^n, a_i \in \mathbb{R}, i = 1, n \}$$

$A = \left[A^1 \; A^2 \cdots A^n \right]$, $A^i \in \mathbb{R}^m$ **sunt coloanele matricii** A

Teoremă

Fie $\boldsymbol{A \in \mathbb{R}^{m \times n}\ (m \geq n)}$, $\boldsymbol{b \in \mathbb{R}^{m}}$. Un vector $\boldsymbol{x \in \mathbb{R}^{n}}$ minimizează norma euclidiană a vectorului reziduu $\boldsymbol{\|r(x)\|_2 = \|b - Ax\|_2}$, rezolvând problema ***(LSP)***, dacă şi numai dacă:

$$\boldsymbol{r(x) \perp \text{range}(A) \quad \Leftrightarrow \quad A^T r(x) = 0}$$

sau echivalent

$$\boldsymbol{A^T A x = A^T b} \qquad \textbf{(3)}$$

Sistemul (3) poartă numele de sistemul de ***ecuaţii normale***.

Este un sistem pătratic de dimensiune $\boldsymbol{n}$, matricea sistemului $\boldsymbol{A^T A \in \mathbb{R}^{n \times n}}$ este simetrică. Sistemul de ecuaţii normale (3) este nesingular dacă şi numai dacă $\mathbf{rang}\boldsymbol{A} = \boldsymbol{n}$, în acest caz soluţia $\boldsymbol{x}$ a sistemului (3) este unică.

$$\det A^T A \neq 0 \Leftrightarrow \operatorname{rang} A = n$$

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}, \quad A^T A = \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}, \quad A^T b = \begin{pmatrix} 3 \\ 6 \end{pmatrix}$$

$$\begin{aligned} 14x_1 + 32x_2 &= 3 \\ 32x_1 + 77x_2 &= 6 \end{aligned} \Rightarrow x_1 = \frac{13}{18}, \quad x_2 = -\frac{2}{9}$$

Pseudo-inversa matricii A

Presupunem că A are **rang** $A = n$. Atunci pseudo-inversa poate fi definită ca:

$$A^{+} = \left(A^{T} A\right)^{-1} A^{T} \in \mathbb{R}^{n \times m} \qquad (A^{+} = A^{I} \quad ?)$$

$$A^{+} = \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}^{-1} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Rezolvarea sistemului de ecuaţii normale

1) Folosind factorizarea Cholesky (descompunere $\boldsymbol{LU}$) pentru matrici simetrice:

   $\boldsymbol{A^T A = LL^T}$, $\boldsymbol{L \in \mathbb{R}^{n \times n}}$ **matrice inferior triunghiulară**

- Se calculează matricea $\boldsymbol{A^T A}$ şi vectorul $\boldsymbol{A^T b}$;
- Se calculează factorizarea Cholesky a matricii $\boldsymbol{A^T A = LL^T}$;
- Se rezolvă sistemul inferior triunghiular $\boldsymbol{Ly = A^T b}$ pentru $\boldsymbol{y}$;
- Se rezolvă sistemul superior triunghiular $\boldsymbol{L^T x = y}$ pentru $\boldsymbol{x}$;

2) Se calculează descompunerea $\boldsymbol{QR}$ (cu algoritmul lui

Householder adaptat) pentru matricea $\boldsymbol{A}$:

$\boldsymbol{A = QR}$, $\boldsymbol{Q \in \mathbb{R}^{m \times m}}$ **matrice ortogonală** , $\boldsymbol{R \in \mathbb{R}^{m \times n}}$,

$$\boldsymbol{R = \begin{bmatrix} \bar{R} \in \mathbb{R}^{n \times n} \\ 0_{(m-n) \times n} \end{bmatrix}} \quad , \quad \boldsymbol{\bar{R}} \text{ – matrice superior triunghiulară}$$

- Se calculează factorizarea $\boldsymbol{QR}$ modificată a matricii $\boldsymbol{A}$;
- Se calculează vectorul $\boldsymbol{Q^T b}$;
- Se rezolvă sistemul sup. triunghiular $\boldsymbol{\bar{R}x = \left(Q^T b\right)_{i=1,n}}$;

3) Se foloseşte desc. după valori singulare a matricii $\boldsymbol{A}$

$$\boldsymbol{A = U\Sigma V^T}, \ \boldsymbol{\Sigma} \in \mathbb{R}^{m \times n} \ , \boldsymbol{U} \in \mathbb{R}^{m \times m} \ , \ \boldsymbol{V} \in \mathbb{R}^{n \times n}$$

- Se calculează SVD pentru matricea $\boldsymbol{A=U\Sigma V^T}$;
- Se calculează vectorul $\boldsymbol{U^T b}$;
- Se rezolvă sistemul diagonal $\boldsymbol{\Sigma w = U^T b}$ pentru $\boldsymbol{w}$;
- Soluţia este $\boldsymbol{x=Vw}$;

1), 2) sau 3) ? → se recomandă **2)**

Calcul Numeric

Cursul 9

2022

Anca Ignat

Interpolare numerică

Presupunem că despre o funcţie f cunoaştem doar valorile într-un număr finit de puncte. Pornind de la aceste date, dorim să aproximăm funcţia f într-un alt punct.

x	x_0	x_1	x_2	...	x_{n-1}	x_n
f	y_0	y_1	y_2	...	y_{n-1}	y_n

În tabelul de mai sus $f(x_i) = y_i$, $i=0,1,\ldots,n$ şi $x_i \neq x_j$, $i \neq j$.

Dat un punct $x \neq x_i$, $i=0,1,\ldots,n$ dorim să aproximăm $f(x)$ cunoscând cele $(n+1)$ perechi (x_i, y_i), $i=0,\ldots,n$. Punctele x_i se numesc ***noduri de interpolare***.

Polinomul de interpolare Lagrange

Notăm cu Π_n mulțimea polinoamelor de grad cel mult n. Dimensiunea acestui spațiu este $n+1$, baza uzuală fiind dată de polinoamele $1, x, x^2,\ldots, x^n$. Vom considera o altă bază în acest spațiu. Se consideră polinoamele p_i:

$$p_i \in \Pi_n \text{ astfel ca } p_i(x_j) = \begin{cases} 0 & \text{pentru } j \neq i \\ 1 & \text{pentru } j = i \end{cases}, \; j = 0,\ldots,n \,, \; i = 0,\ldots,n$$

Din relația $p_i(x_j)=0, \; \forall j \neq i$ şi faptul că p_i este de grad n rezultă că $x_0, x_1,\ldots,x_{i-1}, x_{i+1},\ldots,x_n$ sunt cele n rădăcini ale polinomului p_i.

Avem:

$$p_i(x) = c_i(x - x_0)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n),$$
$$c_i \in \mathbb{R}, i = 0,\ldots,n$$

Constanta c_i se determină din relaţia $p_i(x_i) = 1$:

$$p_i(x_i) = 1 = c_i(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n) \Rightarrow$$
$$c_i = \frac{1}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}$$

Polinoamele p_i au forma:

$$p_i(x) = \frac{(x - x_0)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)} = \prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j}\right)$$

$$i = 0, \ldots, n$$

Propoziţie

Polinoamele $p_0, p_1, \ldots, p_n$ formează o bază în Π_n.

Demonstraţie: Vom arăta că cele $n+1$ polinoame sunt liniar independente:

$$q(x) = a_0 p_0(x) + a_1 p_1(x) + \cdots + a_n p_n(x) = 0\,, \forall x$$

$$\Rightarrow a_0 = a_1 = \cdots = a_n = 0$$

Vom face pe rând $x = x_0$, $x = x_1$,…, $x = x_n$ în polinomul q:

$$x = x_0 \quad q(x_0) = a_0 p_0(x_0) + a_1 p_1(x_0) + \cdots + a_n p_n(x_0) =$$

$$= a_0 1 + a_1 0 + \cdots + a_n 0 = a_0 = 0 \quad \Rightarrow a_0 = 0$$

$$x = x_1 \quad q(x_1) = 0 \quad \Rightarrow \quad a_1 = 0$$

$$\vdots$$

$$x = x_k \quad q(x_k) = a_0 p_0(x_k) + \cdots + a_k p_k(x_k) + \cdots + a_n p_n(x_k) =$$

$$= a_0 0 + \cdots + a_k 1 + \cdots + a_n 0 = a_k = 0 \Rightarrow a_k = 0$$

$$\vdots$$

$$x = x_n \quad q(x_n) = 0 \quad \Rightarrow \quad a_n = 0$$

Toate constantele a_i sunt nule deci polinoamele $\{p_i; i = 0,\ldots,n\}$ formează o bază în Π_n.

Pentru a aproxima funcţia f pornind de la tabelul de mai sus, vom construi un polinom $l_n \in \Pi_n$ a.î. să satisfacă ***condiţiile de interpolare***:

$$l_n \in \Pi_n \quad , \quad l_n(x_i) = y_i \quad , \quad \forall i = 0,\ldots,n \qquad (1)$$

Odată construit acest polinom, vom aproxima $f(x)$ prin $l_n(x)$, $f(x) \approx l_n(x)$

Vom scrie polinomul l_n în raport cu noua bază $\{p_i; i = 0,\ldots,n\}$, deci există constantele reale $a_0, a_1,\ldots,a_n$ astfel ca:

$$l_n(x) = \sum_{i=0}^{n} a_i p_i(x)$$

Constantele a_k se determină astfel:

$$y_k = l_n(x_k) = a_0 p_0(x_k) + \cdots + a_k p_k(x_k) + \cdots + a_n p_n(x_k) =$$

$$= a_0 0 + \cdots + a_k 1 + \cdots + a_n 0 = a_k \Rightarrow a_k = y_k$$

Prin urmare un polinom de grad n care îndeplinesc condiţiile de interpolare (1) este:

$$l_n(x) = \sum_{i=0}^{n} y_i p_i(x) = \sum_{i=0}^{n} y_i \frac{(x - x_0)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_0)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}$$

$$= \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j \neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j}\right) \qquad (2)$$

Polinomul din formula (2) se numeşte ***polinomul de interpolare Lagrange***.

Propoziţie

Polinomul l_n dat de formula (2) este unicul polinom de grad n care îndeplineşte condiţiile de interpolare (1).

Demonstraţie: Presupunem că mai există un polinom $q \in \Pi_n$ care îndeplineşte condiţiile (1):

$$q \in \Pi_n \quad , \; q(x_i) = y_i \quad , \quad \forall i = 0,\ldots,n$$

Fie polinomul $p(x)=l_n(x)-q(x) \in \Pi_n$.

$$p(x_k) = l_n(x_k) - q(x_k) = y_k - y_k = 0 \ , \forall k = 0,\ldots,n$$

Polinomul p are ca rădăcini toate nodurile de interpolare.
Polinomul p este polinom de grad cel mult n şi are $(n+1)$ rădăcini distincte ($x_i \neq x_j, \forall \ i \neq j$). Acest polinom nu poate fi decât polinomul identic nul:

$$p(x) = l_n(x) - q(x) \equiv 0 \quad \forall x \ , \quad l_n(x) = q(x) \quad \forall x$$

Polinomul l_n este unicul care satisface (2).

Fie w_{n+1} polinomul de grand $(n+1)$ care are ca rădăcini nodurile de interpolare:

$$w_{n+1}(x) = (x - x_0)(x - x_1)\cdots(x - x_n) \in \Pi_{n+1}$$

Fie $a = \min\{x_0, x_1, \ldots, x_n\}$, $b = \max\{x_0, x_1, \ldots, x_n\}$.

Teorema restului (eroarea la interpolarea Lagrange)

Fie $f \in C^{n+1}[a,b]$ şi $\bar{x} \in [a,b],\ \bar{x} \neq x_i\ , \forall i = 0,\ldots,n.$

Atunci există un punct $y \in [a,b],\ y = y(x_0, x_1, \ldots, x_n, \bar{x})$ (punctul y depinde de nodurile de interpolare x_i şi de punctul $\bar{x}$) astfel că eroarea la interpolarea numerică este dată de:

$$f(\bar{x}) - l_n(\bar{x}) = \frac{f^{(n+1)}(y)}{(n+1)!} w_{n+1}(\bar{x}) \qquad (3)$$

Demonstraţie: Considerăm funcţia F:

$$F(x) := f(x) - l_n(x) - c w_{n+1}(x)$$

Constanta reală c este aleasă astfel ca $F(\bar{x}) = 0$ adică:

$$c = \frac{f(\bar{x}) - l_n(\bar{x})}{w_{n+1}(\bar{x})},\ (\ x \neq x_i\ \ \forall i\ ) \Rightarrow\ w_{n+1}(\bar{x}) \neq 0\ ) \qquad (4)$$

Funcţia f fiind de clasă C^{n+1} pe intervalul $[a,b]$ rezultă că şi funcţia F este din $C^{n+1}[a,b]$. Avem:

$$F(x_i) = f(x_i) - l_n(x_i) - cw_{n+1}(x_i) = y_i - y_i - c\,0 = 0\ \ , \forall i = 0,\ldots,n$$

Funcţia F are $(n+2)$ zerouri, $x_0, x_1, \ldots, x_n, \bar{x}$. Aplicând succesiv Teorema lui Rolle rezultă că F' are $(n+1)$ zerouri, F'' are n zerouri,…, $F^{(n+1)}$ are 1 zero în intervalul $[a,b]$. Vom nota această rădăcină a lui $F^{(n+1)}$ cu y. Punctul y depinde de zerourile iniţiale $x_0, x_1, \ldots, x_n, \bar{x}$ şi:

$$y = y(x_0, x_1, \ldots, x_n, \bar{x}) \in [a,b] \quad \text{a.î.} \quad F^{(n+1)}(y) = 0. \qquad (5)$$

Derivata de ordinul ***(n+1)*** a funcţiei ***F*** se calculează astfel:

$$F^{(n+1)}(x) = f^{(n+1)}(x) - l_n^{(n+1)}(x) - c\, w_{n+1}^{(n+1)}(x) =$$

$$= f^{(n+1)}(x) - 0 - c(n+1)! = f^{(n+1)}(x) - c(n+1)! \qquad (6)$$

(derivata de ordin ***(n+1)*** a polinomului de grad ***n*** l_n este ***0***).

Din relaţiile (4), (5) şi (6) rezultă că:

$$c = \frac{f^{(n+1)}(y)}{(n+1)!} = \frac{f(\bar{x}) - l_n(\bar{x})}{w_{n+1}(\bar{x})} \Rightarrow f(\bar{x}) - l_n(\bar{x}) = \frac{f^{(n+1)}(y)}{(n+1)!} w_{n+1}(\bar{x})$$

Forma Newton a polinomului de interpolare Lagrange

Fie $l_k(x, x_0, x_1,\ldots, x_k, f)$ polinomul de interpolare Lagrange pentru funcţia f pe sistemul de noduri distincte $\{x_0, x_1,\ldots, x_k\}$.

Propoziţie

Fie $l_{k-1}(x, x_0, x_1,\ldots, x_{k-1}, f)$, $l_{k-1}(x, x_1, x_2,\ldots, x_k, f) \in \Pi_{k-1}$ polinoamele de interpolare Lagrange pentru funcţia f pe sistemele de noduri $\{x_0, x_1,\ldots, x_{k-1}\}$ şi respectiv $\{x_1, x_2,\ldots, x_k\}$. Atunci:

$$l_k(x, x_0, x_1,\ldots, x_k, f) =$$

$$= \frac{(x - x_0) l_{k-1}(x, x_1, x_2,\ldots, x_k, f) - (x - x_k) l_{k-1}(x, x_0, x_1,\ldots, x_{k-1}, f)}{x_k - x_0} \qquad (1)$$

Demonstraţie: Exerciţiu.

Considerăm următoarele probleme de interpolare pentru f:

$$\{(x_0, y_0), (x_1, y_1), \ldots, (x_{k-1}, y_{k-1})\} \rightarrow l_{k-1}(x, x_0, x_1, \ldots, x_{k-1}, f)$$

$$\{(x_0, y_0), (x_1, y_1), \ldots, (x_k, y_k)\} \rightarrow l_k(x, x_0, x_1, \ldots, x_k, f)$$

Ne interesează să găsim o formulă de trecere rapidă de la polinomul de interpolare pe k noduri la cel care are un nod în plus. Deoarece polinomul de grad cel mult k:

$$q(x) = l_k(x, x_0, x_1, \ldots, x_k, f) - l_{k-1}(x, x_0, x_1, \ldots, x_{k-1}, f) \in \Pi_k$$

are ca rădăcini punctele $x_0, x_1, \ldots, x_{k-1}$ $(q(x_i) = y_i - y_i = 0, i=0,\ldots,k-1)$ avem relaţia:

$$l_k(x, x_0, x_1, \ldots, x_k, f) = l_{k-1}(x, x_0, x_1, \ldots, x_{k-1}, f) + A\prod_{j=0}^{k-1}(x - x_j) \quad (2)$$

în care A este dat de relaţia:

$$A = \frac{l_k(x_k, x_0, x_1, \ldots, x_k, f) - l_{k-1}(x_k, x_0, x_1, \ldots, x_{k-1}, f)}{\prod_{j=0}^{k-1}(x_k - x_j)} \quad (3)$$

$$A = \frac{y_k}{\prod_{j=0}^{k-1}(x_k - x_j)} - \frac{\sum_{i=0}^{k-1} y_i \prod_{\substack{j=0 \\ j\neq i}}^{k-1}\left(\frac{x_k - x_j}{x_i - x_j}\right)}{\prod_{j=0}^{k-1}(x_k - x_j)} =$$

$$= \frac{y_k}{\prod_{j=0}^{k-1}(x_k - x_j)} - \sum_{i=0}^{k-1} \frac{y_i}{(x_k - x_i)\prod_{\substack{j=0 \\ j\neq i}}^{k-1}(x_i - x_j)}$$

$$A = \sum_{i=0}^{k} \frac{y_i}{\prod_{\substack{j=0 \\ j \neq i}}^{k} (x_i - x_j)} \qquad (4)$$

$$\left[x_k \right]_f = y_k = f(x_k) \ ,$$

$$\left[x_0, x_1, \ldots, x_k \right]_f = \frac{\left[x_1, x_2, \ldots, x_k \right]_f - \left[x_0, x_1, \ldots, x_{k-1} \right]_f}{x_k - x_0}$$

numită ***diferenţă divizată de ordin k a funcţiei f pe nodurile*** $\{x_0, x_1, \ldots, x_k\}$.

Propoziție

$$\left[x_0, x_1, \ldots, x_k\right]_f = \sum_{i=0}^{k} \frac{y_i}{\prod\limits_{\substack{j=0 \\ j \neq i}}^{k} (x_i - x_j)} = \sum_{i=0}^{k} \frac{y_i}{\left(w_{n+1}(x_k)\right)'}$$

pentru orice sistem de noduri $\{x_0, x_1, \ldots, x_k\}$ şi orice k.

Demonstraţie: Se face prin inducţie. Pentru $k=1$ avem:

$$\left[x_0, x_1\right]_f = \frac{y_0}{x_0 - x_1} + \frac{y_1}{x_1 - x_0} = \frac{\left[x_1\right]_f - \left[x_0\right]_f}{x_1 - x_0}$$

Presupunem că relaţia (8) este valabilă pentru orice k şi pentru orice sistem de noduri $\{x_0, x_1, \ldots, x_k\}$. Pentru $k+1$ folosim relaţia de recurenţă şi apoi aplicăm ipoteza inductivă:

$$[x_0, x_1, \ldots, x_{k+1}]_f = \frac{[x_1, x_1, \ldots, x_{k+1}]_f - [x_0, x_2, \ldots, x_k]_f}{x_{k+1} - x_0} =$$

$$= \frac{1}{x_{k+1} - x_0} \left(\sum_{i=1}^{k+1} \frac{y_i}{\prod\limits_{\substack{j=1 \\ j \neq i}}^{k+1} (x_i - x_j)} - \sum_{i=0}^{k} \frac{y_i}{\prod\limits_{\substack{j=0 \\ j \neq i}}^{k} (x_i - x_j)} \right) =$$

$$= \frac{1}{x_{k+1} - x_0} \left\{ - \frac{y_0}{\prod\limits_{\substack{j=0 \\ j \neq 0}}^{k} (x_0 - x_j)} + \frac{y_{k+1}}{\prod\limits_{\substack{j=1 \\ j \neq k+1}}^{k+1} (x_{k+1} - x_j)} + \right.$$

$$\left. + \sum_{i=1}^{k} \left[\frac{y_i}{\prod\limits_{\substack{j=1 \\ j \neq i}}^{k} (x_i - x_j)} \left(\frac{1}{x_i - x_{k+1}} - \frac{1}{x_i - x_0} \right) \right] \right\} =$$

$$= \frac{y_0}{\prod_{\substack{j=0 \\ j\neq 0}}^{k+1}(x_0 - x_j)} + \frac{y_{k+1}}{\prod_{\substack{j=0 \\ j\neq k+1}}^{k+1}(x_{k+1} - x_j)} + \sum_{i=1}^{k} \frac{y_i}{\prod_{\substack{j=0 \\ j\neq i}}^{k+1}(x_i - x_j)} =$$

$$= \sum_{i=0}^{k+1} \frac{y_i}{\prod_{\substack{j=0 \\ j\neq i}}^{k+1}(x_i - x_j)}$$

Inducţia este completă.

Din definiţie se observă că diferenţa divizată $[x_0, x_1, \ldots, x_k]_f$ nu depinde de ordinea nodurilor $\{x_0, x_1, \ldots, x_k\}$.

Vom nota în continuare cu $l_k(x)$ polinomul de interpolare Lagrange pe nodurile $\{x_0, x_1, \ldots, x_k\}$ pentru funcţia f. Avem:

$$l_n(x) = l_0(x) + [l_1(x) - l_0(x)] + \cdots + [l_k(x) - l_{k-1}(x)] + \cdots + [l_n(x) - l_{n-1}(x)] =$$
$$= y_0 + [x_0, x_1]_f (x - x_0) + \cdots + [x_0, x_1, \ldots, x_k]_f (x - x_0) \cdots (x - x_{k-1}) + \cdots$$
$$+ [x_0, x_1, \ldots, x_n]_f (x - x_0) \cdots (x - x_{n-1})$$

Am obţinut ***forma Newton*** a polinomului de interpolare Lagrange:

$$l_n(x) = y_0 + \left[x_0, x_1\right]_f (x - x_0) + \left[x_0, x_1, x_2\right]_f (x - x_0)(x - x_1) + \cdots +$$
$$+ \left[x_0, x_1, \ldots, x_n\right]_f (x - x_0) \cdots (x - x_{n-1})$$

Schema lui Aitken de calcul a diferențelor divizate

Ne propunem să calculăm diferențele divizate

$$\left[x_0, x_1\right]_f, \left[x_0, x_1, x_2\right]_f, \ldots, \left[x_0, x_1, \ldots, x_n\right]_f$$

necesare construirii polinomului de interpolare Lagrange în forma Newton. Procedeul folosește definiția recursivă a diferențelor divizate și se desfășoară în n pași. La pasul 1 se calculează numai diferențe divizate de ordinul 1:

$$\left[x_0, x_1\right]_f, \left[x_1, x_2\right]_f, \ldots, \left[x_{n-1}, x_n\right]_f.$$

În general, la pasul k se calculează diferențe divizate de ordin k:

$$\left[x_0, x_1, \ldots, x_k\right]_f, \left[x_1, x_2, \ldots, x_{k+1}\right]_f, \ldots, \left[x_{n-k}, x_{n-k+1}, \ldots, x_n\right]_f.$$

La pasul n se calculează o singură diferență divizată de ordin n și anume $\left[x_0, x_1, \ldots, x_n\right]_f$.

		Pas 1	$\cdots$	**Pas k**	$\cdots$	**Pas n**
x_0	y_0					
x_1	y_1	$[x_0, x_1]_f$				
x_2	y_2	$[x_1, x_2]_f$				
$\vdots$						
x_k	y_k	$[x_{k-1}, x_k]_f$		$[x_0, x_1, \ldots, x_k]_f$		
$\vdots$				$\vdots$	$\ddots$	
x_{n-1}	y_{n-1}	$[x_{n-2}, x_{n-1}]_f$		$[x_{n-k-1}, \ldots, x_{n-1}]_f$		
x_n	y_n	$[x_{n-1}, x_n]_f$		$[x_{n-k}, \ldots, x_{n-1}]_f$	$\cdots$	$[x_0, x_1 \ldots, x_n]_f$

Notăm **dd[i,k]=** $\left[x_i, x_{i+1}, \ldots, x_{i+k}\right]_f$ diferenţa divizată de ordin ***k***, pe nodurile consecutive $\left\{x_i, x_{i+1}, \ldots, x_{i+k}\right\}$ ***i=0,…,n-k, k=1,…,n,*** cu ***dd[i,0]=y_i, i=0,…,n***. Schema lui Aitken se implementează astfel:

$$dd[i,0] = y_i,\ i = 0,\ldots,n;$$

for $k = 1,\ldots,n$

for $i = 0,\ldots,n-k$

$$dd[i,k] = \frac{dd[i+1,k-1] - dd[i,k-1]}{x_{i+k} - x_i}$$

Putem face aceleași calcule folosind un singur vector, de exemplu rescriind vectorul y astfel:

$$\textbf{for } k = 1,\ldots,n$$

$$\textbf{for } i = n,\ldots,k$$

$$y_i = \frac{y_i - y_{i-1}}{x_i - x_{i-k}}$$

La finalul acestei secvențe de program, vectorul y va conține elementele:

$$y_0, \left[x_0, x_1\right]_f, \left[x_0, x_1, x_2\right]_f, \ldots, \left[x_0, x_1, \ldots, x_n\right]_f$$

$$(y_k = \left[x_0, x_1, \ldots, x_k\right]_f, \; k=0,\ldots,n).$$

Interpolare Newton pe noduri echidistante

Pp. că nodurile de interpolare sunt echidistante:

$$x_i = x_0 + i\,h \quad , \quad i = 0,1,...,n$$

În relația de mai sus fie se dă h distanța între 2 noduri succesive, fie se precizează primul și ultimul nod, x_0 și x_n iar h se calculează:

$$h = \frac{(x_n - x_0)}{n} \quad .$$

$$\left[x_i, x_{i+1}\right]_f = \frac{f(x_{i+1}) - f(x_i)}{(x_{i+1} - x_i)} = \frac{y_{i+1} - y_i}{h}$$

Se introduce noțiunea de **diferență finită de ordinul 1:**

$$\Delta f(x) = f(x+h) - f(x)$$

Pornind de la această definiție se pot introduce și diferențe finite de ordin superior:

$$\Delta^2 f(x) = \Delta(\Delta f(x)) = \Delta(f(x+h) - f(x)) =$$

$$= f(x+2h) - 2f(x+h) + f(x)$$

și în general se pot introduce recursiv **diferențele finite de ordin** k:

$$\Delta^k f(x) = \Delta(\Delta^{k-1} f(x)) = \Delta^{k-1} f(x+h) - \Delta^{k-1} f(x).$$

Prin inducție după k, se poate deduce formula de calcul a diferențelor finite de ordin k folosind doar valorile funcției f:

$$\Delta^k f(x) = \sum_{i=0}^{k} (-1)^{k-i} C_k^i \, f(x + ih)$$

.

Observație: Dacă funcția f este polinom de grad m atunci $\Delta f(x)$ este polinom de grad $m-1$, $\Delta^2 f(x)$ este polinom de grad $m-2$, ş.a.m.d. Prin urmare:

$$\Delta^k f(x) \equiv 0 \text{ , pentru } k > m \text{ , } f \text{ – polinom de grad } m.$$

Legătura între diferențele divizate şi cele finite:

$$[x_i, x_{i+1}]_f = \frac{f(x_{i+1}) - f(x_i)}{(x_{i+1} - x_i)} = \frac{\Delta f(x_i)}{h}$$

$$[x_i, x_{i+1}, x_{i+2}]_f = \frac{[x_{i+1}, x_{i+2}]_f - [x_i, x_{i+1}]_f}{(x_{i+2} - x_i)} = \frac{\Delta^2 f(x_i)}{2h^2}$$

Prin inducție se poate arăta următoarea legătură între diferențele divizate de ordin k şi cele finite:

$$\left[x_i, x_{i+1}, \ldots, x_{i+k}\right]_f = \frac{\Delta^k f(x_i)}{k!h^k}.$$

Polinoame de interpolare pe noduri echidistante

$$l_n(x) = y_0 + \left[x_0, x_1\right]_f (x - x_0) + \left[x_0, x_1, x_2\right]_f (x - x_0)(x - x_1) + \cdots +$$
$$+ \left[x_0, x_1, \ldots, x_k\right]_f (x - x_0)(x - x_1)\cdots(x - x_{k-1}) + \cdots +$$
$$+ \left[x_0, x_1, \ldots, x_n\right]_f (x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

Consideră că punctul de interpolare este de forma:

$$\overline{x} = x_0 + t\,h$$

și înlocuim diferențele divizate cu diferențe finite în forma Newton a polinomului de interpolare:

$$(\overline{x} - x_0)\cdots(\overline{x} - x_{k-1}) = (x_0 + t\,h - x_0)\cdots(x_0 + t\,h - x_0 - (k-1)h) =$$

$$= h^k t(t-1)\cdots(t-k+1)$$

$$l_n(\overline{x}) = l_n(x_0 + th) = y_0 + \Delta f(x_0)t + \Delta^2 f(x_0)\frac{t(t-1)}{2} + \cdots +$$

$$+ \Delta^k f(x_0)\frac{t(t-1)\cdots(t-k+1)}{k!} + \cdots +$$

$$+ \Delta^n f(x_0)\frac{t(t-1)\cdots(t-n+1)}{n!}$$

Această relație poartă numele de **formula lui Newton progresivă pe noduri echidistante.**

Considerăm polinomul de interpolare Lagrange pe nodurile în ordine inversă $\{x_n, x_{n-1}, ..., x_0\}$:

$$l_n(x) = y_n + \left[x_n, x_{n-1}\right]_f (x - x_n) + \left[x_n, x_{n-1}, x_{n-2}\right]_f (x - x_n)(x - x_{n-1}) +$$
$$+ \cdots + \left[x_n, x_{n-1}, ..., x_0\right]_f (x - x_n)(x - x_{n-1}) \cdots (x - x_0)$$

Dacă punctul de interpolare este de forma:

$$\overline{x} = x_n + th$$

analog ca mai sus obține **formula lui Newton regresivă pe noduri echidistante:**

$$l_n(\bar{x}) = l_n(x_n + th) = y_n + \Delta f(x_{n-1})t + \Delta^2 f(x_{n-2})\frac{t(t+1)}{2} + \cdots +$$

$$+ \Delta^k f(x_{n-k})\frac{t(t+1)\cdots(t+k-1)}{k!} + \cdots +$$

$$+ \Delta^n f(x_0)\frac{t(t+1)\cdots(t+n-1)}{n!}$$

Schema lui Aitken pentru diferenţe finite

Pentru formulele lui Newton progresive/regresive, avem nevoie de calculul următoarelor diferenţe finite:

$\Delta f(x_0)$, $\Delta^2 f(x_0)$, …, $\Delta^k f(x_0)$, …,$\Delta^n f(x_0)$ - pentru formula progresivă

$\Delta f(x_{n-1})$, $\Delta^2 f(x_{n-2})$, …, $\Delta^k f(x_{n-k})$, …, $\Delta^n f(x_0)$ - pentru formula regresivă

Metoda de calcul a acestor diferenţe finite este similră schemei Aitken pentru dieferenţele divizate.

Pas 1	⋯	**Pas k**	⋯	**Pas n**
y_0				
y_1	$\Delta f(x_0)$			
y_2	$\Delta f(x_1)$			
$\vdots$				
y_k	$\Delta f(x_{k-1})$	$\Delta^k f(x_0)$		
$\vdots$		$\vdots$	$\ddots$	
y_{n-1}	$\Delta f(x_{n-2})$	$\Delta^k f(x_{n-k-1})$		
y_n	$\Delta f(x_{n-1})$	$\Delta^k f(x_{n-k})$	⋯	$\Delta^n f(x_0)$

Funcții spline

Fie nodurile:

$$x_i \in [a,b], i = 0,1,\ldots,n,$$

cu

$$a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$$

Se consideră funcția continuă polinomială pe porțiuni:

$$S(x) = P_i(x) \quad \textbf{pentru} \quad x \in [x_i, x_{i+1}] \quad \forall i = 0,\ldots,n-1$$

$$S(x)=\begin{cases} P_0(x), & x\in[x_0,x_1], \\ P_1(x), & x\in[x_1,x_2], \\ P_2(x), & x\in[x_2,x_3], \\ \quad\vdots & \\ P_{n-2}(x), & x\in[x_{n-2},x_{n-1}], \\ P_{n-1}(x), & x\in[x_{n-1},x_n]. \end{cases}$$

$P_i(x)$, $i=0,...,n$ sunt polinoame. O asemenea funcție poartă numele de ***funcție spline***.

Funcții spline liniare continue

Definiție

Funcția $S(x)$ definită mai sus se numește ***funcție spline liniară continuă*** dacă polinoamele $P_i(x), i = 0,\ldots,n-1$ sunt polinoame de gradul 1 și $S(x) \in C[a,b]$, adică:

$$\lim_{\substack{x \to x_i \\ x < x_i}} S(x) = \lim_{\substack{x \to x_i \\ x > x_i}} S(x), i = 1,\ldots,n-1.$$

Fie funcția $f : [a,b] \to \mathbb{R}$ pentru care se cunosc valorile:

$$y_i = f(x_i), i = 0,\ldots,n.$$

Funcția spline liniară de interpolare S pentru funcţia f îndeplinește condițiile de interpolare:

$$S(x_i) = y_i \ , i = 0,\ldots,n.$$

Ținând seamă că polinoamele $P_i(x)$ sunt polinoame de gradul 1 și $S(x)$ este continuă vom avea condițiile:

$$\begin{cases} P_i(x_i) = y_i \ , \\ P_i(x_{i+1}) = y_{i+1}, \quad i = 0,\ldots,n-1, \\ P_i(x) - \text{polinom de gradul 1.} \end{cases}$$

Din aceste condiții rezultă:

$$P_i(x) = \frac{x - x_i}{x_{i+1} - x_i} y_{i+1} + \frac{x_{i+1} - x}{x_{i+1} - x_i} y_i \ , i = 0,\ldots,n-1$$

$$S(x_k) = P_{k-1}(x_k) = P_k(x_k) = y_k \ , k = 1,\ldots,n-1,$$
$$S(x_0) = P_0(x_0) = y_0 \ , S(x_n) = P_{n-1}(x_n) = y_n \ .$$

Funcții spline cubice de clasă C^2

Se consideră sistemul de noduri distincte din intervalul ***[a,b]***:

$$\Delta = \{a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b\}$$

Funcția ***S(x)*** asociată divizării Δ care îndeplinește condițiile :

$$S(x) \in C^2[a,b],$$

polinoamele $P_i(x)$ **au gradul** 3 , $i = 0,\ldots,n-1,$

se numește ***funcție spline cubică***.

Dată fiind o funcție $f:[a,b]\to\mathbb{R}$ cu valorile:

$$y_i = f(x_i),\ i=0,\ldots,n,$$

se consideră funcția spline cubică $S(x)$ de interpolare ce satisface

$$S(x_i)=y_i,\ i=0,\ldots,n.$$

Pentru determinarea funcției spline cubice de interpolare observăm că polinoamele:

$$P_i(x)=\alpha_i x^3+\beta_i x^2+\gamma_i x+\delta_i,\ x\in[x_i,x_{i+1}],\ i=0,\ldots,n-1,$$

implică determinarea a celor $4n$ necunoscute $\{\alpha_i,\beta_i,\gamma_i,\delta_i;\ i=0,\ldots,n-1\}$ pentru care se impun:

$$\begin{cases} n+1 \text{ condiţii din relaţiile de interpolare } S(x_i) = y_i \ , i = 0,\ldots,n, \\ 3(n-1) \text{ condiţii de continuitate pentru } S(x), S'(x) \text{ şi } S''(x) \\ \qquad \text{în nodurile } x_i \ , i = 1,\ldots,n-1 \ , \end{cases}$$

în total ***4n-2*** condiții.

Se pot avea în vedere pentru adăugarea a două condiții suplimentarea următoarele abordări :

- fixarea pantelor în extremitățile intervalului ***[a,b].*** Se presupune că funcția ***f*** este derivabilă şi se cunosc valorile ***f′(a), f′(b)***. Se impun condițiile:
  $$S'(x_0) = P_0'(x_0) = f'(a), \ S'(x_n) = P_{n-1}'(x_n) = f'(b);$$
- periodicitatea primelor două derivate:

$$f'(a) = f'(b) \quad (S'(x_0) = P_0'(x_0) = P_{n-1}'(x_n) = S'(x_n)),$$

$$f''(a) = f''(b) \quad (S''(x_0) = P_0''(x_0) = P_{n-1}''(x_n) = S''(x_n));$$

- anularea derivatei secunde în capetele intervalului:

$$f''(a) = f''(b) = 0$$

$$(S''(x_0) = P_0''(x_0) = 0, S''(x_n) = P_{n-1}''(x_n) = 0).$$

Funcțiile spline care îndeplinesc aceste condiții se numesc ***funcții spline cubice normale***.

- derivata de ordinul al treilea a funcției S este continuă în punctele x_1 și x_{n-1} . Aceasta înseamnă că polinoamele P_0, P_1 respectiv P_{n-2}, P_{n-1} coincid. Acest tip de funcție spline se numește „*not – a – knot*” și este utilizat în MATLAB.

Vom calcula în cele ce urmează funcția spline cubică în cazul în care cunoaștem suplimentar valorile celei de-a doua derivate a funcției f în capetele intervalului de interpolare:

$$a_0 = f''(a),\ a_n = f''(b)\ .$$

Recapitulând, vom avea următoarele condiții:

$$\begin{cases} P_i(x_i) = y_i \ , i = 0,\ldots,n-1, \ P_{n-1}(x_n) = y_n \ - \text{interpolare}, \\ P_{i-1}(x_i) = P_i(x_i), i = 1,\ldots,n-1, - \text{continuitatea funcţiei } S, \\ P'_{i-1}(x_i) = P'_i(x_i), i = 1,\ldots,n-1, - \text{continuitatea primei derivatei}, \\ P''_{i-1}(x_i) = P''_i(x_i), i = 1,\ldots,n-1, - \text{continuitatea derivatei secunde}, \\ P''_0(x_0) = a_0 = f''(a), \ P''_{n-1}(x_n) = a_n = f''(b). \end{cases}$$

Vom nota:

$$S''(x_i) = a_i \ , i = 0,\ldots n.$$

Ținând seama de faptul că funcția $S'' \in C[a,b]$ este o funcție liniară pe fiecare din intervalele $[x_i, x_{i+1}]$ rezultă că:

$$S''(x)=\frac{x-x_i}{h_i}a_{i+1}+\frac{x_{i+1}-x}{h_i}a_i\ ,\ x\in[x_i,x_{i+1}],\quad \forall i=0,\ldots,n-1$$

$$h_i=x_{i+1}-x_i\quad ,\ i=0,\ldots,n-1$$

iar din

$$S'(x)=\int S''(x)dx\ ,\qquad S(x)=\int S'(x)dx$$

rezultă:

$$S(x)=\frac{\left(x-x_i\right)^3}{6h_i}a_{i+1}+\frac{\left(x_{i+1}-x\right)^3}{6h_i}a_i+b_ix+c_i\ ,$$

$$x\in[x_i,x_{i+1}],b_i,c_i\in\mathbb{R}\ ,i=0,\ldots n-1,$$

$$P_i(x) = \frac{(x - x_i)^3}{6h_i} a_{i+1} + \frac{(x_{i+1} - x)^3}{6h_i} a_i + b_i x + c_i \ ,$$

$$b_i, c_i \in \mathbb{R}, i = 0, \ldots n-1,$$

Vom calcula funcția spline pentru cazul:

$$S''(a) = a_0 = P_0''(x_0),$$

$$S''(b) = a_n = P_{n-1}''(x_n).$$

(a_0 și a_n sunt două constante cunoscute)

Impunând condițiile de interpolare și de continuitate vom obține:

$$P_i(x_i) = \frac{h_i^2}{6} a_i + b_i x_i + c_i = y_i \ ,$$

$$P_i(x_{i+1}) = \frac{h_i^2}{6} a_{i+1} + b_i x_{i+1} + c_i = y_{i+1} \ , \ i = 0,\ldots n-1.$$

Din aceste relații calculăm b_i şi c_i în funcţie de $a_i \ , a_{i+1}, y_i \ , y_{i+1}$:

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}\left(a_{i+1} - a_i\right),$$

$$i = 0,\ldots n-1$$

$$c_i = \frac{x_{i+1} y_i - x_i y_{i+1}}{h_i} - \frac{h_i}{6}\left(x_{i+1} a_i - x_i a_{i+1}\right).$$

Din condiția de continuitate a primei derivate a funcției spline cubice $P'_{i-1}(x_i) = P'_i\left(x_i\right) \ , \ i = 1,...,n-1$, ținând seama de:

$$P'_{i-1}(x)=\frac{\left(x-x_{i-1}\right)^2}{2h_{i-1}}a_i-\frac{\left(x_i-x\right)^2}{2h_{i-1}}a_{i-1}+b_{i-1},$$

$$P'_i(x)=\frac{\left(x-x_i\right)^2}{2h_i}a_{i+1}-\frac{\left(x_{i+1}-x\right)^2}{2h_i}a_i+b_i,$$

rezultă, utilizând formulele pentru b_{i-1} și b_i deduse mai sus:

$$P'_{i-1}(x_i)=\frac{h_{i-1}}{2}a_i+\frac{y_i-y_{i-1}}{h_{i-1}}-\frac{h_{i-1}}{6}\left(a_i-a_{i-1}\right)=$$

$$P'_i(x_i)=-\frac{h_i}{2}a_i+\frac{y_{i+1}-y_i}{h_i}-\frac{h_i}{6}\left(a_{i+1}-a_i\right)$$

sau

$$(h_{i-1}+h_i)\,a_i + h_i\,a_{i+1} = 6\left(\frac{y_{i+1}-y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right),\ i = 1,\ldots,n-1.$$

(1)

Pentru $i=1$ și $i=n$ din (1) avem:

$$2(h_0+h_1)a_1 + h_1 a_2 = 6\left(\frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0}\right) - h_0 a_0$$

$$h_{n-2}a_{n-2} + 2(h_{n-2}+h_{n-11})a_{n-1} = 6\left(\frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}}\right) - h_{n-1}a_n$$

Sistemul liniar format din ecuațiile (1) cu necunoscutele $\{a_1, a_2, \cdots, a_{n-1}\}$ are forma:

$$Ha = f \text{, cu } H \in \mathbb{R}^{n \times n}, f \in \mathbb{R}^n$$

$$2(h_0 + h_1)a_1 + h_1 a_2 = 6\left(\frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0}\right) - h_0 f''(a)$$

$$(h_{i-1} + h_i)a_i + h_i a_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) \quad , \; i = 2, \ldots, n-1$$

$$h_{n-2} a_{n-2} + 2(h_{n-2} + h_{n-11})a_{n-1} = 6\left(\frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}}\right) - h_{n-1} f''(b)$$

$$H = \begin{bmatrix} 2(h_0+h_1) & h_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ h_1 & 2(h_1+h_2) & h_2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & h_2 & 2(h_2+h_3) & h_3 & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} \\ 0 & 0 & 0 & 0 & \cdots & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{bmatrix}$$

$$f = \begin{bmatrix} 6\left(\frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0}\right) - h_0 f''(a) \\ 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) \quad i = 1, \ldots, n-1 \\ 6\left(\frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}}\right) - h_{n-1} f''(b) \end{bmatrix}$$

Matricea $\boldsymbol{H}$ are diagonala dominantă atât pe linii cât și pe coloane, este simetrică și pozitiv definită prin urmare putem utiliza metoda Gauss-Seidel sau o metodă de relaxare pentru rezolvarea sistemului $\boldsymbol{Ha=f}$.

Interpolare în sensul celor mai mici pătrate

x	x_0	x_1	x_2	...	x_{n-1}	x_n
f	y_0	y_1	y_2	...	y_{n-1}	y_n

$$f(x_i) = y_i \ , \ i=0,...,n$$

$$f(x) \approx S_f(x; a_0, a_1, ..., a_m)$$

$$S_f(x; a_0, a_1, ..., a_m) = a_m x^m + a_{m-1} x^{m-1} + \ldots + a_1 x + a_0$$

Coeficienții $a_0, a_1, ..., a_m$ se găsesc rezolvând problema de minimizare în sensul celor mai mici pătrate:

$$\min\{ \sum_{r=0}^{n} \left(S_f(x_r; a_0, a_1, ..., a_m) - y_r \right)^2 ; a_0, a_1, ..., a_m \in \mathbb{R} \} \quad \textbf{(LSP)}$$

$$g : \mathbb{R}^{m+1} \to \mathbb{R}_+ ,$$

$$g(a_0, a_1, ..., a_m) = \sum_{r=0}^{n} \left(S_f(x_r; a_0, a_1, ..., a_m) - y_r \right)^2$$

$$g(a_0, a_1, ..., a_m) = \sum_{r=0}^{n} \left(a_m x_r^m + \cdots + a_k x_r^k + \cdots + a_1 x_r + a_0 - y_r \right)^2$$

$$\frac{\partial g}{\partial a_k}(a_0, a_1, ..., a_m) = 2 \sum_{r=0}^{n} \left(a_m x_r^m + \cdots + a_k x_r^k + \cdots + a_1 x_r + a_0 - y_r \right) x_r^k$$

Soluția problemei de minimizare a problemei **(LSP)** este obținută rezolvând sistemul de ecuații liniare, de dimensiune ***(m+1)***:

$$\frac{\partial g}{\partial a_k}(a_0, a_1, ..., a_m) = 0\ ,\ \ k = 0, 1, ..., m$$

$$\sum_{r=0}^{n} \left(a_m x_r^m + \cdots + a_k x_r^k + \cdots + a_1 x_r + a_0 \right) x_r^k = \sum_{r=0}^{n} y_r x_r^k\ ,\ k = 0, ..., m$$

$$a_0 \sum_{r=0}^{n} x_r^k + a_1 \sum_{r=0}^{n} x_r^{k+1} + \cdots + a_{m-1} \sum_{r=0}^{n} x_r^{k+m-1} + a_m \sum_{r=0}^{n} x_r^{k+m} = \sum_{r=0}^{n} y_r x_r^k\ ,$$

$$k = 0, ..., m$$

Constantele $\{ a_0, a_1, ..., a_m \}$ sunt soluția sistemului liniar:

$$Ba = z \ ,$$

$$B \in \mathbb{R}^{(m+1)\times(m+1)} \ , \ B = (b_{kj})_{k,j=0}^{m} \ , \ z \in \mathbb{R}^{(m+1)} \ z = (z_k)_{k=0}^{m}$$

$$b_{kj} = \sum_{r=0}^{n} x_r^{k+j} \quad , \quad z_k = \sum_{r=0}^{n} y_r x_r^k \ , \qquad k, j = 0, ..., m$$

Calcul Numeric

Cursul 10

2022

Anca Ignat

Rezolvarea ecuaţiilor neliniare

Fie $f:[a,b]\to\mathbb{R}$ o **funcţie continuă** pe intervalul $[a,b]$ astfel ca $f(a)f(b)<0$. În aceste condiţii, există $x^*\in(a,b)$ astfel ca $f(x^*)=0$. În cele ce urmează ne propunem să aproximăm soluţia x^* a ecuaţiei neliniare $f(x)=0$.

Metoda bisecţiei (a înjumătăţirii intervalului)

Pp. $f(a)f(b)<0!!!$ Pentru a aproxima soluţia x^* căutată, vom construi un şir de intervale $\{[a_k,b_k];k\geq 0\}$ ce satsifac:

$$x^* \in \left[a_k, b_k\right]$$

$$\left[a_{k+1}, b_{k+1}\right] \subset \left[a_k, b_k\right]$$

$$b_{k+1} - a_{k+1} = \frac{b_k - a_k}{2}$$

Pentru primul interval vom considera:

$$a_0 = a \ , \ b_0 = b \ \ , \ \ k = 0.$$

Considerăm punctul c de mijloc al intervalului $[a_k, b_k]$:

$$c = \frac{a_k + b_k}{2}$$

Avem următoarele 3 variante:

1. $f(c)=0$ - soluţia căutată este $x^*=c$, algoritmul se opreşte;
2. $f(a_k)f(c) < 0 \rightarrow$ soluţia se găseşte în intervalul (a_k,c), continuăm procedeul cu intervalul $[a_{k+1} = a_k \ , \ b_{k+1} = c]$;
3. $f(b_k)f(c) < 0 \rightarrow$ soluţia se găseşte în intervalul $x^* \in (c,b_k)$ procedeul continuă cu intervalul $[a_{k+1} = c \ , \ b_{k+1} = b_k]$.

Dat $\varepsilon > 0$ există un interval $\left[a_{\bar{k}}, b_{\bar{k}}\right]$ astfel ca $x^* \in \left(a_{\bar{k}}, b_{\bar{k}}\right)$ şi $b_{\bar{k}} - a_{\bar{k}} < \varepsilon$ ($\bar{k} > \log_2 \frac{b-a}{\varepsilon}$). Pentru ε suficient de mic atât $a_{\bar{k}}$ cât şi $b_{\bar{k}}$ pot fi considerate aproximări ale soluţiei x^* ($a_{\bar{k}} \approx x^*$ prin lipsă iar $b_{\bar{k}} \approx x^*$ prin adaos).

Metoda tangentei (Newton-Raphson)

Vom presupune că funcţia $f \in C^1[a,b]$ este derivabilă pe $[a,b]$ cu derivata continuă în acest interval şi satisface relaţia $f(a)f(b) < 0$. Pentru a aproxima soluţia x^* a ecuaţiei $f(x)=0$ vom construi un şir $\{x_k\} \subseteq \mathbb{R}$ care să conveargă la x^*, $x_k \to x^*$, pentru $k \to \infty$. Primul element din şir, x_0, considerăm că este dat. Următorul element din şir se construieşte ca fiind punctul de intersecţie al tangentei la graficul funcţiei f în punctul $(x_0 , f(x_0))$ cu axa absciselor. Procedeul se repetă cu x_1 pentru a-l obţine pe x_2, ş.a.m.d.

$x_1 = Ox \cap$ **tangenta la graficul funcţiei** f **în punctul** $\left(x_0, f(x_0)\right)$

$x_2 = Ox \cap$ **tangenta la graficul funcţiei** f **în punctul** $\left(x_1, f(x_1)\right)$

$\vdots$

$x_{k+1} = Ox \cap$ **tangenta la graficul fcţ.** f **în pt.** $\left(x_k, f(x_k)\right), k = 0,1,2,$

Ecuaţia tangentei la graficul funcţiei f într-un punct $(a, f(a))$ este următoarea (pentru o funcţie derivabilă):

$$y = f(a) + f'(a)(x - a)$$

Pentru a calcula x_{k+1} din x_k vom considera ecuaţia tangentei:

$$y = f(x_k) + f'(x_k)(x - x_k)$$

unde luăm $y = 0$. Avem:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}\ ,\ k = 0,1,2,\ldots\ ,\ x_0 \text{ dat}$$

Formula de mai sus poate fi folosită doar dacă la fiecare pas $f'(x_k) \neq 0$. Dacă la un pas avem $f'(x_k) = 0$ putem calcula câteva iteraţii x_k $(k \geq \overline{k})$ folosind $f'(x_{\overline{k}-1})$.

Teoremă de convergenţă

Fie $f \in C^2[a,b]$, cu $f(a)f(b)<0$, $f'(x)\neq 0$ şi $f''(x)\neq 0$ $\forall x \in [a,b]$. Dacă alegem

$$x_0 = \begin{cases} a & \text{pentru } f(a)f''(a)>0 \\ b & \text{pentru } f(b)f''(b)>0 \end{cases}$$

atunci şirul $\{x_k ; k \geq 0\}$ construit cu metoda tangentei este monoton, mărginit şi convergent la unica soluţie x^* a ecuaţiei $f(x)=0$. Ordinul de convergenţă este mai mare decât 2.

Metoda falsei poziţii (a coardei)

Presupunem că funcţia f este continuă, $f \in C[a,b]$ şi satisface relaţia $f(a)f(b) < 0$. Vom construi un şir $\{x_k\} \subseteq \mathbb{R}$ care să conveargă la soluţia căutată x^*, $x_k \to x^*$, pentru $k \to \infty$. Considerăm date primul element din şir, x_0 şi un alt punct $\tilde{x}$. Procedeul de construire a şirului este următorul:

$x_1 = Ox \cap$ **dreapta ce uneşte punctele** $\left(\tilde{x}, f(\tilde{x})\right), \left(x_0, f(x_0)\right)$

$x_2 = Ox \cap$ **dreapta ce uneşte punctele** $\left(\tilde{x}, f(\tilde{x})\right), \left(x_1, f(x_1)\right)$

$\vdots$

$x_{k+1} = Ox \cap$ **dreapta ce uneşte pt.** $\left(\tilde{x}, f(\tilde{x})\right), \left(x_k, f(x_k)\right), k = 0,1,2,..$

Ecuaţia dreptei ce trece prin punctele ***(a, f(a))*** cu ***(b, f(b))*** este:

$$\frac{y - f(a)}{f(a) - f(b)} = \frac{x - a}{a - b}.$$

Pentru a-l obţine pe x_{k+1} din x_k avem:

$$\frac{y - f(x_k)}{f(x_k) - f(\tilde{x})} = \frac{x - x_k}{x_k - \tilde{x}} \quad \text{cu} \quad y = 0$$

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - \tilde{x})}{f(x_k) - f(\tilde{x})} = \frac{\tilde{x} f(x_k) - x_k f(\tilde{x})}{f(x_k) - f(\tilde{x})} \ ,$$

$$k = 0, 1, 2, \ldots, \quad x_0, \tilde{x} - \text{date}$$

Teoremă de convergenţă

Fie $f \in C^2[a,b]$, cu $f(a)f(b)<0$, $f'(x)\neq 0$ şi $f''(x)\neq 0$ $\forall x \in [a,b]$. Dacă alegem:

$$\begin{cases} \tilde{x} = a \text{ şi } x_0 = b & \text{pentru } f(a)f''(a)>0 \\ \tilde{x} = b \text{ şi } x_0 = a & \text{pentru } f(b)f''(b)>0 \end{cases}$$

atunci şirul $\{x_k; k \geq 0\}$ construit cu metoda falsei poziţii este monoton, mărginit deci convergent la unica soluţie x^* a ecuaţiei $f(x)=0$.

Metoda secantei

Presupunem că funcţia f este continuă, $f \in C[a,b]$ şi satisface relaţia $f(a)f(b) < 0$. Vom construi un şir $\{x_k\} \subseteq \mathbb{R}$ care să conveargă la soluţia căutată x^*, $x_k \to x^*$, pentru $k \to \infty$. Considerăm date primele două elemente din şir, x_0 şi x_1.

Procedeul de construire a şirului este următorul:

$x_2 = \mathbf{O}x \cap$ **dreapta ce uneşte punctele** $\big(x_0, f(x_0)\big), \big(x_1, f(x_1)\big)$,

$x_3 = \mathbf{O}x \cap$ **dreapta ce uneşte punctele** $\big(x_1, f(x_1)\big), \big(x_2, f(x_2)\big)$,

$\vdots$

$x_{k+1} = \mathbf{O}x \cap$ **dreapta ce uneşte pct.** $\big(x_{k-1}, f(x_{k-1})\big), \big(x_k, f(x_k)\big)$,

$k = 1, 2, \ldots$

Obţinem elementul x_{k+1} din x_k şi x_{k-1} astfel:

$$\frac{y - f(x_k)}{f(x_k) - f(x_{k-1})} = \frac{x - x_k}{x_k - x_{k-1}} \quad \textbf{cu} \quad y = 0$$

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} =$$

$$= \frac{x_{k-1} f(x_k) - x_k f(x_{k-1})}{f(x_k) - f(x_{k-1})} \quad ,$$

$k = 1, 2, \ldots$, x_0, x_1 dați

Teoremă de convergenţă

Fie x^* o soluţie a ecuţiei $f(x)=0$. Pp. că $f \in C^2\left[x^* - r, x^* + r\right]$, $f'(x) \neq 0$ şi $f''(x) \neq 0$ $\forall x \in \left[x^* - r, x^* + r\right]$. Atunci există $0 < r_0 \leq r$ pentru care, dacă $x_0, x_1 \in [x^* - r_0, x^* + r_0]$ atunci $x_k \in [x^* - r_0, x^* + r_0], \forall k \geq 2$ şi $x_k \to x^*$, pentru $k \to \infty$.

Ordinul de convergenţă este $q = \dfrac{1+\sqrt{5}}{2} \approx 1.61803$.

Metoda lui Laguerre

Fie polinomul:

$$p(x) = a_0(x - x_1)(x - x_2)\cdots(x - x_n)\ ,\ a_0 \neq 0$$

Metoda lui Laguerre propune construirea unui şir de numere care să conveargă la una din rădăcinile polinomului p.
Considerăm derivata polinomului p:

$$p'(x) = p(x)\left[\frac{1}{x-x_1} + \frac{1}{x-x_2} + \cdots + \frac{1}{x-x_n}\right]$$

Avem:

$$\ln|p(x)| = \ln|a_0| + \ln|x-x_1| + \ln|x-x_2| + \cdots + \ln|x-x_n|$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\ln|p(x)| = \frac{1}{x-x_1} + \frac{1}{x-x_2} + \cdots + \frac{1}{x-x_n} = \frac{p'(x)}{p(x)} = G(x)$$

$$-\frac{\mathrm{d}^2}{\mathrm{d}x^2}\ln|p(x)| = \frac{1}{(x-x_1)^2} + \frac{1}{(x-x_2)^2} + \cdots + \frac{1}{(x-x_n)^2} =$$

$$= \frac{\left[p'(x)\right]^2 - p(x)p''(x)}{\left[p(x)\right]^2} = H(x)$$

Fie x_1 rădăcina pe care vrem s-o aproximăm şi y_k valoarea aproximativă curentă. Notăm cu $a = y_k - x_1$ şi facem presupunerea că y_k se află la acceeaşi distanţă de toate celelalte rădăcini, adică:

$$y_k - x_i = b \quad \forall i = 2,\ldots,n.$$

Prin urmare avem:

$$G(y_k)=\frac{p'(y_k)}{p(y_k)}=\frac{1}{a}+\frac{n-1}{b}$$

$$H(y_k)=\frac{\left[p'(y_k)\right]^2-p(y_k)p''(y_k)}{\left[p(y_k)\right]^2}=\frac{1}{a^2}+\frac{n-1}{b^2}$$

Rezolvăm acest sistem în raport cu a şi obţinem:

$$a=\frac{n}{\max\left[G(y_k)\pm\sqrt{(n-1)\left(nH(y_k)-G^2(y_k)\right)}\right]}$$

Semnul la numitor este ales astfel ca expresia să aibă magnitudine maximă. Dacă ţinem cont de expresiile pentru G şi H obţinem pentru a următoarea formulă:

$$a = \frac{n\, p(y_k)}{\max\left[p'(y_k) \pm \sqrt{(n-1)^2 \left[p'(y_k) \right]^2 - n(n-1) p(y_k) p''(y_k)} \right]}$$

Următorul element din şir va fi:

$$y_{k+1} = y_k - a = y_k - \frac{n}{\max\left[G(y_k) \pm \sqrt{(n-1)\left(nH(y_k) - G^2(y_k)\right)}\right]}$$

$$y_{k+1} = y_k - \frac{n\, p(y_k)}{\max\left[p'(y_k) \pm \sqrt{(n-1)^2\left[p'(y_k)\right]^2 - n(n-1)p(y_k)p''(y_k)}\right]}$$

Procedeul se opreşte când ***a*** devine suficient de mic. Metoda lui Laguerre se poate apl. şi ptr. aprox. răd. complexe şi de asemenea pentru polinoame cu coeficienţi complecşi. Pentru rădăcini simple metoda lui Laguerre are ordinul de convergenţă ***3***.

Sisteme de ecuaţii neliniare

Consideră sistemul neliniar:

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases} \Leftrightarrow \quad F(X) = 0,\ F = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix},\ X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Fie matricea jacobiană asociată funcţiei F (presupunem că funcţiile f_i sunt diferenţiabile):

$$\nabla F(X)=\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}(x_1, x_2, \dots, x_n)$$

Pentru a găsi soluţia X^* a sistemului de ecuaţii neliniare $F(X)= 0$ se construieşte un şir de vectori $X^{(k)} \in \mathbb{R}^n$ astfel:

$$X^{(0)} - \text{dat}$$

$$X^{(k+1)} = X^{(k)} - \left[\nabla F(X^{(k)})\right]^{-1} F(X^{(k)}) = X^{(k)} + \Delta^{(k)},$$

$$\Delta^{(k)} = -\left[\nabla F(X^{(k)})\right]^{-1} F(X^{(k)}) \in \mathbb{R}^n \quad , \quad k = 0,1,\ldots$$

Vectorul de corecţie $\Delta^{(k)}$ poate fi calculat şi ca soluţie a sistemului liniar:

$$\nabla F(X^{(k)})\Delta = -F(X^{(k)})$$

unde matricea sistemului este matricea jacobiană calculată în punctul $X^{(k)}$ iar vectorul termenilor liberi este $\left(-F(X^{(k)})\right)$.

Metoda descrisă mai sus poartă numele de metoda Newton. Pentru ***n=1*** metoda Newton este chiar metoda tangentei descrisă anterior.

Optimizare numerică

$$\min \{ f(x);\ x \in \mathbb{R}^n \},\ f : \mathbb{R}^n \to \mathbb{R}$$

Un punct $x^* \in \mathbb{R}^n$ se numeşte **punct de minim global** pentru funcţia f dacă $f(x^*) \le f(x)\ \ \forall x \in \mathbb{R}^n$.

Un punct $x^* \in \mathbb{R}^n$ se numeşte **punct de minim local** pentru funcţia f dacă există o vecinătate V a punctului x^* pentru care $f(x^*) \le f(x)\ \ \forall x \in V$.

Un punct $x^* \in \mathbb{R}^n$ se numeşte **punct de minim strict local** pentru funcţia f dacă există o vecinătate V a punctului x^* pentru care $f(x^*) < f(x)\ \ \forall x \in V,\ x \neq x^*$.

$$V = S(x^*, r) = \left\{ x \in \mathbb{R}^n ; \left\| x - x^* \right\| \le r \right\}.$$

Dacă funcţia $f \in C^1(\mathbb{R}^n)$ se numeşte **gradient** al funcţiei f în punctul x vectorul:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n} \right)^T .$$

Dacă funcţia $f \in C^2(\mathbb{R}^n)$ se numeşte **matrice hessiană** a funcţiei f în punctul x matricea:

$$\nabla^2 f(x) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right)_{i,j=1,\ldots,n}$$

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ & \vdots & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{pmatrix}$$

Teorema lui Taylor

Dacă funcţia $f \in C^1(\mathbb{R}^n)$ este continuu diferenţiabilă atunci există $t \in (0,1)$ astfel ca:

$$f(x+p) = f(x) + \nabla f(x+tp)^T p. \quad (1)$$

Dacă $f \in C^2(\mathbb{R}^n)$ este de două ori continuu diferenţiabilă atunci există $t \in (0,1)$ astfel ca:

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp) p. \quad (2)$$

Teorema 1 (condiţii necesare de optim de ordinul întâi)

Fie $f \in C^1(\mathbb{R}^n)$ şi $x^* \in \mathbb{R}^n$ un punct de minim local pentru f. Atunci $\nabla f(x^*) = 0$.

Un punct x^* pentru care $\nabla f(x^*) = 0$ se numeşte **punct staţionar**. Teorema de mai sus spune că orice punct de minim local trebuie să fie punct staţionar.

Teorema 2 (condiţii necesare de optim de ordinul al doilea)

Fie $f \in C^2(\mathbb{R}^n)$ şi $x^* \in \mathbb{R}^n$ un punct de minim local pentru f. Atunci $\nabla f(x^*) = 0$ şi matricea hessiană $\nabla^2 f(x^*)$ este pozitiv semidefinită.

O matrice $A \in \mathbb{R}^{n \times n}$ este **pozitiv semidefinită** dacă

$$\left(Ax, x\right)_{\mathbb{R}^n} \geq 0 \quad \forall x \in \mathbb{R}^n$$

şi este **pozitiv definită** dacă

$$\left(Ax, x\right)_{\mathbb{R}^n} > 0 \quad \forall x \in \mathbb{R}^n, \; x \neq 0.$$

Teorema 3 (condiţii suficiente de ordinul al doilea)

Fie $f \in C^2(\mathbb{R}^n)$ şi $x^* \in \mathbb{R}^n$ un punct pentru care $\nabla f(x^*) = 0$ şi matricea hessiană $\nabla^2 f(x^*)$ este pozitiv definită. Atunci x^* este punct de minim strict local pentru f.

O funcţie f se numeşte **convexă** dacă:

$$f\left(a\,x_1 + (1-a)x_2\right) \le af(x_1) + (1-a)\,f(x_2)\ ,\forall x_1, x_2 \in \mathbb{R}^n, \forall a \in [0,1].$$

Teorema

Dacă f este funcţie convexă orice punct de minim local este punct de minim global. Dacă în plus $f \in C^1(\mathbb{R}^n)$ atunci orice punct staţionar este punct de minim global.

Metode de descreştere

Se numeşte **direcţie de descreştere** a funcţiei f în punctul x un vector $d \in \mathbb{R}^n$ pentru care

$$f(x+\alpha d) \leq f(x) \ , \ \forall \alpha \in [0, \bar{\alpha}).$$

Teoremă

Fie $f \in C^1(\mathbb{R}^n)$. Un vector d este direcţie de descreştere pentru f în x dacă şi numai dacă:

$$\nabla f(x)^T d = \left(d, \nabla f(x)\right)_{\mathbb{R}^n} < 0.$$

Algoritm de descreştere

1. alege $\boldsymbol{x \in \mathbb{R}^n}$
2. do
   - găseşte o direcţie de descreştere $\boldsymbol{d}$ a lui $\boldsymbol{f}$ în $\boldsymbol{x}$
   - găseşte $\boldsymbol{\tilde{\alpha} > 0}$ astfel ca

     $$\boldsymbol{f(x+\tilde{\alpha}d) = \min\{ f(x+\alpha d);\ \alpha \in (0,\bar{\alpha})\}}$$

     (ajustarea pasului – line search)
   - $\boldsymbol{x = x + \tilde{\alpha} d}$

   while (nu am găsit soluţia)

Ajustarea pasului (line search)

Trebuie rezolvată o problemă de minimizare unidimensională

$$\min\{g(a)\ ;\ a \in [b,c]\}$$

Metoda Newton

Fie a_k aproximarea curentă a soluţiei problemei de minimizare de mai sus. Vom folosi dezvoltarea în serie Taylor a funcţiei g

$$g(a) = g(a_k) + g'(a_k)(a - a_k) + \frac{1}{2} g''(a_k)(a - a_k)^2 + \frac{1}{3!} g'''(b_k)(a - a_k)^3$$

$$q(a) = g(a_k) + g'(a_k)(a - a_k) + \frac{1}{2} g''(a_k)(a - a_k)^2$$

Pentru $a \approx a_k$ putem considera că funcţia q aproximează funcţia g , $q(a) \approx g(a)$ şi

$$\min\{g(a) ; a \in [c,b]\} \simeq \min\{q(a) ; a \in [c,b]\}.$$

Construim elementul a_{k+1} ca fiind soluţia problemei:

$$q(a_{k+1}) = \min\{q(a) ; a \in [c,b]\}$$

a_{k+1} este soluţia ecuaţiei

$$q'(a) = 0 \Leftrightarrow q'(a) = g'(a_k) + g''(a_k)(a - a_k) \Rightarrow \boxed{a_{k+1} = a_k - \frac{g'(a_k)}{g''(a_k)}}$$

Metoda secantei

Dacă în relaţiile de mai sus se face următoarea aproximare:

$$g''(a_k) \approx \frac{g'(a_k) - g'(a_{k-1})}{a_k - a_{k-1}}$$

obţinem următoarea metodă de aproximare, numită şi metoda secantei:

$$\boxed{a_{k+1} = a_k - g'(a_k)\frac{a_k - a_{k-1}}{g'(a_k) - g'(a_{k-1})}}$$

Aproximare spline cubică

Vom folosi pentru a construi a_{k+1} pe a_{k-1} , a_k , $g(a_{k-1})$, $g(a_k)$, $g'(a_{k-1})$, $g'(a_k)$. Aproximăm funcţia g cu un polinom de grad 3

$$g(a) \approx q(a) = c_0 a^3 + c_1 a^2 + c_2 a + c_3$$

Funcţia q (respectiv constantele c_i) se calculează a.î.

$$q(a_{k-1}) = g(a_{k-1}) \quad , \quad q(a_k) = g(a_k)$$
$$q'(a_{k-1}) = g'(a_{k-1}) \quad , \quad q'(a_k) = g'(a_k)$$

a_{k+1} este punctul de minim al funcţiei q

$$a_{k+1} = a_k - (a_k - a_{k-1})\left[\frac{g'(a_k) + u_2 - u_1}{g'(a_k) - g'(a_{k-1}) + 2u_2}\right]$$

$$u_1 = g'(a_{k-1}) + g'(a_k) - 3\frac{g(a_k) - g(a_{k-1})}{a_k - a_{k-1}}$$

$$u_2 = \sqrt{u_1^2 - g'(a_{k-1})g'(a_k)}$$

Ajustarea inexactă a pasului

$$f(x+\tilde{\alpha}d) \cong \min\{ f(x+\alpha d);\ \alpha \in (0,\bar{\alpha})\}$$

- nu se obţine $\tilde{\alpha}$ optimal
- pentru reducerea timpului de calcul optimizarea se opreşte înainte de a ajunge la soluţie, în funcţie de anumite criterii/teste de oprire

Regula lui Armijo

$$g(a) = f(x_k + a\,d_k) \quad , \quad \bar{g}(a) = g(0) + \varepsilon a\, g'(0) \, , \; \varepsilon \in (0,1)$$

$\bar{a}$ este acceptabil după regula lui Armijo dacă

$$\textbf{(1)} \qquad g(\bar{a}) \leq \bar{g}(\bar{a})$$

$$\textbf{(2)} \qquad g(\sigma\bar{a}) \geq \bar{g}(\sigma\bar{a})$$

$k = 0;$

se alege a_0

while $(g(a_k) > \bar{g}(a_k))$

- $a_{k+1} = \frac{1}{\sigma} a_k$
- $k = k + 1$

$\sigma = 2$ **sau 10** , $\varepsilon = 0.2$

Testul Goldstein

Dat $\varepsilon \in (0,2), \bar{\alpha}$ este considerat acceptabil dacă:

$$g(\bar{\alpha}) \leq \bar{g}(\bar{\alpha}) \quad \text{si} \quad g(\bar{\alpha}) > g(0) + (1-\varepsilon) g'(0)$$

$$x_{k+1} = x_k + \bar{\alpha} d_k$$

$\bar{\alpha}$ este acceptat dacă

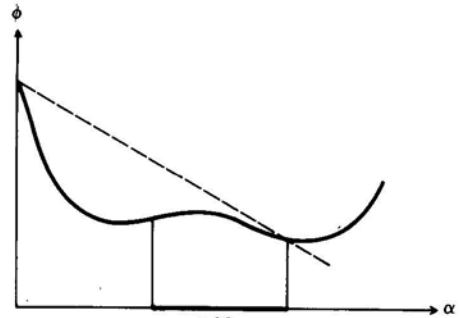
$$\varepsilon \leq \frac{f(x_{k+1}) - f(x_k)}{\alpha \nabla f(x_k) d_k} \leq 1 - \varepsilon$$

Testul Wolfe

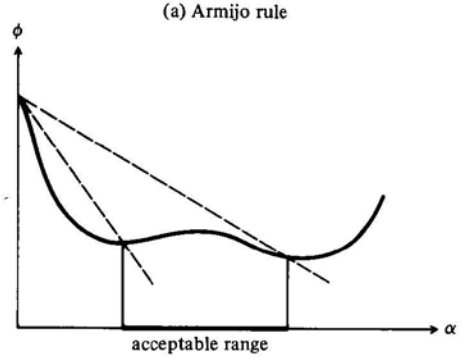
$\varepsilon \in (0,2)$

$\bar{\alpha}$: $g(\bar{\alpha}) \leq \bar{g}(\bar{\alpha}) = g(0) + \varepsilon \bar{\alpha} g'(0)$

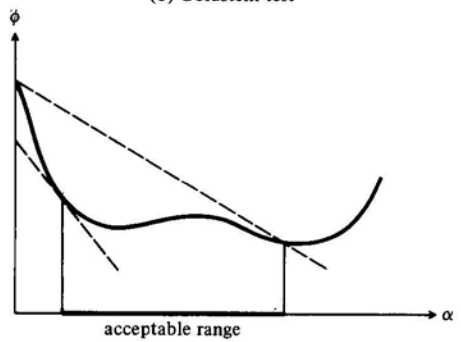
$g'(\bar{\alpha}) \geq (1-\varepsilon) g'(0)$



(a) Armijo rule



(b) Goldstein test



(c) Wolfe test

Calcul Numeric

Cursul 11

2022

Anca Ignat

Metoda pantei maxime

$\min\{ f(x) ; x \in \mathbb{R}^n \}$, $f : \mathbb{R}^n \to \mathbb{R}$

x_0 dat , $x_{k+1} = x_k + \alpha_k d_k$, $k = 0,1,...$

d_k direcţie de descreştere a lui f în x_k

$\alpha_k > 0$ $f(x_k + \alpha_k d_k) = \min\{ f(x_k + \alpha d_k) ; \alpha \in [0, \bar{\alpha}] \}$

d_k direcţie de descreştere dacă :

$$\nabla f(x_k)^T d_k < 0.$$

Metoda pantei maxime:

$$\boldsymbol{d_k = -\nabla f(x_k)}$$

Cazul pătratic:

$$\boldsymbol{f(x) = \frac{1}{2} x^T A x - x^T b = \frac{1}{2}\left(Ax, x\right)_{\mathbb{R}^n} - \left(b, x\right)_{\mathbb{R}^n}}$$

$\boldsymbol{b \in \mathbb{R}^n}$, $\boldsymbol{A \in \mathbb{R}^{n \times n}}$ simetrică şi pozitiv definită

$\boldsymbol{A}$ pozitiv definită $\rightarrow$ $\boldsymbol{\det A \neq 0}$, $\boldsymbol{f}$ este strict convexă

$$\boldsymbol{\nabla f(x) = Ax - b} \quad , \quad \boldsymbol{\nabla^2 f(x) = A}$$

$f(x^*) = \min\{ f(x);\ x \in \mathbb{R}^n \}$ x^* punct unic de minim $\Leftrightarrow$

x^* soluţia sistemului liniar $Ax = b$, $x^* = A^{-1}b$

$$g(x) = Ax - b$$

$$x_{k+1} = x_k - \alpha_k g_k \ , \quad g_k = Ax_k - b$$

$$\alpha_k = \arg\min\{ f(x_k - \alpha\ g_k)\ ;\ \alpha \in [0, \bar{\alpha}] \}$$

$$f(x_k - \alpha g_k) = \frac{1}{2}(x_k - \alpha g_k)^T A(x_k - \alpha g_k) - (x_k - \alpha g_k)^T b =$$

$$= \frac{1}{2}\left(g_k^T A g_k\right)\alpha^2 - \left(g_k^T A x_k - g_k^T b\right)\alpha + f(x_k) =$$

$$= \frac{1}{2}\left(g_k^T A g_k\right)\alpha^2 - \left(g_k^T g_k\right)\alpha + f(x_k)$$

$f(x_k - \alpha g_k)$ ecuaţie de gr. 2 în α, coef. lui $\alpha^2, g_k^T A g_k > 0$

$$\alpha_{\min} = \alpha_k = \frac{g_k^T g_k}{g_k^T A g_k}$$

Metoda pantei maxime pentru funcționale pătratice:

$$x_0 - \text{dat}$$

$$x_{k+1} = x_k - \left(\frac{g_k^T g_k}{g_k^T A g_k} \right) g_k \quad , \quad k = 0,1,\ldots$$

$$g_k = A x_k - b$$

$$E(x)=\frac{1}{2}(x-x^*)^T A(x-x^*)= f(x)+\frac{1}{2}x^{*T} Ax^*$$

$$\nabla E(x)=\nabla f(x)= g(x)$$

Şirul construit cu metoda pantei maxime satisface:

$$E(x_{k+1})=\left[1-\frac{\left(g_k^T g_k\right)^2}{\left(g_k^T A g_k\right)\left(g_k^T A^{-1} g_k\right)}\right]E(x_k)$$

Inegalitatea lui Kantorovich

$\boldsymbol{A \in \mathbb{R}^{n\times n},\ A = A^T,\ A > 0}$ pozitiv definită

$$\frac{\left(x^T x\right)^2}{\left(x^T A x\right)\left(x^T A^{-1} x\right)} \geq \frac{4cC}{(c+C)^2}$$

$\boldsymbol{c, C}$ - cea mai mica şi cea mai mare valoare proprie a lui $\boldsymbol{A}$.

Teoremă

Cazul pătratic: Pentru orice iteraţie iniţială x_0, şirul construit cu metoda pantei maxime converge la x^* unicul punct de minim al funcţiei f. Avem:

$$E(x_{k+1}) \leq \left(\frac{C-c}{C+c}\right)^2 E(x_k)$$

Cazul general: $f : \mathbb{R}^n \to \mathbb{R}$, $f \in C^2(\mathbb{R}^n)$, f are un punct de minim local x^*. Presupunem că $F(x^*) = \nabla^2 f(x^*)$ are $c > 0$ cea mai mică valoare proprie şi $C > 0$ cea mai mare valoare proprie. Dacă şirul $\{x_k\}$ construit cu metoda pantei maxime converge la x^* (la fiecare pas $A = \nabla^2 f(x_k)$, $b = \nabla f(x_k)$), $x_k \to x^*$

atunci $f(x_k) \to f(x^*)$ converge liniar cu o rată de convergenţă $\leq \left(\frac{C-c}{C+c}\right)^2$.

Metoda Newton

x_k - punctul curent de aproximare

$$f(x) \simeq f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k) = g(y)$$

g – funcţională pătratică, $y = x - x_k$,

$$g(y)=\frac{1}{2}y^T Ay - y^T b + c$$

$$A=\nabla^2 f(x_k) \quad , \quad b=-\nabla f(x_k) \quad , \quad c=f(x_k)$$

$y^*=\arg\min\{g(y);\ y\in\mathbb{R}^n\}$ este unica soluție a sistemului liniar $Ay=b$, $y^*=A^{-1}b=-\left[\nabla^2 f(x_k)\right]^{-1}\nabla f(x_k)$

Metoda Newton

$$x_{k+1}=x_k-\left[\nabla^2 f(x_k)\right]^{-1}\nabla f(x_k)\,,\ k=0,1,\ldots\ ,\ x_0 \text{ - dat}$$

Teoremă

Fie $f \in C^3(\mathbb{R}^n)$ care are un punct de minim local x^* astfel ca matricea $\nabla^2 f(x^*) > 0$ este pozitiv definită. Dacă punctul de început x_0 este suficient de aproape de x^*, $\| x - x^* \| \leq r$, atunci şirul $\{x_k\}$ generat cu metoda lui Newton converge la x^* şi ordinul de convergenţă este cel puţin 2.

Metoda gradienţilor conjugaţi (a direcţiilor conjugate)

$A \in \mathbb{R}^{n\times n}$, $A = A^T$, $A > 0$ pozitiv definită

$$\min\{ f(x) = \frac{1}{2} x^T A x - x^T b;\ x \in \mathbb{R}^n \}$$

Definiţie

Pentru o matrice $A \in \mathbb{R}^{n\times n}$, $A = A^T$ simetrică, doi vectori $d_1, d_2 \in \mathbb{R}^n$ se numesc ***A*-ortogonali** sau **conjugaţi în raport cu *A*** dacă:

$$d_1^T A d_2 = \left(A d_2, d_1\right)_{\mathbb{R}^n} = \left(d_2, A d_1\right)_{\mathbb{R}^n} = \left(A d_1, d_2\right)_{\mathbb{R}^n} = 0$$

$A = 0_{n\times n} \Rightarrow \forall d_1, d_2$ sunt $A-$ ortogonali

$A = I_n \Rightarrow$ ortogonalitate clasică $\left(d_1, d_2\right)_{\mathbb{R}^n} = 0$

Vectorii $\left\{d_0, d_1, \ldots, d_k\right\}$ se numesc A-ortogonali sau A-conjugaţi dacă:

$$d_i^T A d_j = \left(A d_j, d_i\right)_{\mathbb{R}^n} = 0 \ , \ \forall i \neq j \ , \ i, j = 0,1,\ldots,k$$

Propoziţie

Fie $A \in \mathbb{R}^{n \times n}$, $A = A^T$, $A > 0$, şi $\{d_0, d_1, \ldots, d_k\}$ direcţii A-conjugate, $d_i \neq 0$, $\forall i = 0, \ldots, k$. Atunci vectorii $\{d_0, d_1, \ldots, d_k\}$ sunt liniar independenţi.

$$\left.\begin{array}{l} A \in \mathbb{R}^{n \times n} \ , \ A = A^T \ , \ A > 0 \\ \\ \{d_0, d_1, \ldots, d_{n-1}\} \ - \text{ direcţii } A - \text{conjugate} \end{array}\right\} \Rightarrow \begin{array}{c} \{d_0, d_1, \ldots, d_{n-1}\} \\ \text{bază în } \mathbb{R}^n \end{array}$$

$$x^* = \arg\min\{ f(x);\ x \in \mathbb{R}^n \} \Leftrightarrow x^* \text{ soluţia sist. } Ax = b$$

$$x^* = \alpha_0 d_0 + \alpha_1 d_1 + \cdots + \alpha_{n-1} d_{n-1}$$

$$\left(\underbrace{Ax^*}_{=b}, d_i \right)_{\mathbb{R}^n} = d_i^T A x^* = \alpha_i d_i^T A d_i$$

$$\alpha_i = \frac{d_i^T b}{d_i^T A d_i} = \frac{\left(b, d_i\right)_{\mathbb{R}^n}}{\left(Ad_i, d_i\right)_{\mathbb{R}^n}}$$

$$x^* = \sum_{i=0}^{n-1} \frac{\left(b, d_i\right)_{\mathbb{R}^n}}{\left(Ad_i, d_i\right)_{\mathbb{R}^n}} d_i$$

Considerăm procesul iterativ

$$x_0 \in \mathbb{R}^n - \text{dat}$$

$$x_{k+1} = x_k + \alpha_k d_k \ , k = 0,1,2,...,n-1$$

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T A d_k} \ , \quad g_k = Ax_k - b$$

are proprietatea că $x_n = x^*$.

$$x_k = x_0 + \alpha_0 d_0 + \alpha_1 d_1 + \cdots + \alpha_{k-1} d_{k-1}$$

$$x^* = x_0 + \beta_0 d_0 + \beta_1 d_1 + \cdots + \beta_{n-1} d_{n-1}$$

$$\alpha_i = \frac{d_i^T A(x_k - x_0)}{d_i^T A d_i} \quad , \quad \beta_i = \frac{d_i^T A(x^* - x_0)}{d_i^T A d_i}$$

$$d_k^T A(x_k - x_0) = 0$$

$$x^* - x_k = (\beta_0 - \alpha_0)d_0 + \cdots + (\beta_{k-1} - \alpha_{k-1})d_{k-1} + \beta_k d_k + \cdots + \beta_{n-1} d_{n-1}$$

Corolar

$$g_k^T d_i = 0 \ \forall i < k$$

Algoritmul gradienţilor conjugaţi

$$x_0 \in \mathbb{R}^n \ , \ g_0 = Ax_0 - b$$

$$d_0 = -g_0 = b - Ax_0$$

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T A d_k}$$

$$x_{k+1} = x_k + \alpha_k d_k$$

$$g_{k+1} = Ax_{k+1} - b \ \text{ sau } \ g_{k+1} = g_k + \alpha_k A d_k$$

$$\beta_k = \frac{g_{k+1}^T A d_k}{d_k^T A d_k}$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k$$

$y_1, y_2, ..., y_p \in \mathbb{R}^n$

$\text{span}\{y_1, y_2, ..., y_p\} = \{y = a_1 y_1 + \cdots + a_p y_p \in \mathbb{R}^n;\ a_i \in \mathbb{R}, i = 1, ..., p\}$

= subspaţiul generat de vectorii $y_1, y_2, ..., y_p$

Teoremă

Presupunem că $x_k \neq x^*$. Avem următoarele relaţii:

(1) $$g_k^T g_i = 0 \quad \forall i = 0, 1, ..., k-1$$

(2) $$\text{span}\{g_0, g_1, ..., g_k\} = \text{span}\{g_0, Ag_0, ..., A^k g_0\}$$

(3) $$\text{span}\{d_0, d_1, ..., d_k\} = \text{span}\{g_0, Ag_0, ..., A^k g_0\}$$

(4) $$d_k^T A d_i = 0 \quad \forall i = 0, 1, ..., k-1$$

Şirul $x_k \to x^*$ în cel mult n paşi.

$\mathcal{K}(\boldsymbol{g_0}, \boldsymbol{k}) = \text{span}\{\boldsymbol{g_0}, \boldsymbol{A g_0}, ..., \boldsymbol{A^k g_0}\}$ - se numeşte **subspaţiu Krylov de grad** $\boldsymbol{k}$ pentru $\boldsymbol{g_0}$.

Forma practică a metodei gradienţilor conjugaţi

$$x_0 \in \mathbb{R}^n - \text{dat}, \quad k = 0$$

$$g_0 = Ax_0 - b, \quad d_0 = -g_0$$

$$\textit{while}\,(g_k \neq 0)$$

$$\begin{cases} \alpha_k = \dfrac{g_k^T g_k}{d_k^T A d_k} \\ x_{k+1} = x_k + \alpha_k d_k \\ g_{k+1} = g_k + \alpha_k A d_k \\ \beta_k = \dfrac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \\ d_{k+1} = -g_{k+1} + \beta_k d_k \\ k = k+1; \end{cases}$$

Teoremă

Dacă matricea A are doar r valori proprii distincte, algoritmul gradienţilor conjugaţi calculează soluţia x^* în cel mult r iteraţii.

Teoremă

Dacă $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ sunt valorile proprii ale matricei A atunci:

$$\| x_{k+1} - x^* \|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \| x_0 - x^* \|_A^2$$

$$\| x_{k+1} - x^* \|_A^2 = 2E(x_{k+1})$$

$$E(x_{k+1}) \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 E(x_0).$$

Metodele gradienţilor conjugaţi neliniare

$$f(x) \simeq f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k)$$

$$g_k \leftrightarrow -\nabla f(x_k)$$

$$A \leftrightarrow \nabla^2 f(x_k)$$

Varianta pentru funcţii oarecare a metodei gradienţilor conjugaţi:

$$x_0 \in \mathbb{R}^n - \text{dat}, \; k = 0$$

$$g_0 = \nabla f(x_0), \quad d_0 = -g_0$$

$$while (g_k \neq 0)$$

$$\begin{cases} A = \left[\nabla^2 f(x_k) \right] \\ \alpha_k = -\dfrac{g_k^T d_k}{d_k^T A d_k} \\ x_{k+1} = x_k + \alpha_k d_k \\ g_{k+1} = \nabla f(x_{k+1}) \\ \beta_k = \dfrac{g_{k+1}^T A d_k}{d_k^T A d_k} \\ d_{k+1} = -g_{k+1} + \beta_k d_k \\ k = k + 1; \end{cases}$$

Metoda Fletcher-Reeves

α_k se calculează folosind metoda ajustării pasului

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$$

$$x_0 \in \mathbb{R}^n - \text{dat},$$

$$g_0 = \nabla f(x_0), \quad d_0 = -g_0 \ , \ k = 0$$

$$\textbf{while}\,(g_k \neq 0)$$

$$\begin{cases} \\ \alpha_k = \min\{ f(x_k + \alpha d_k);\ \alpha \in [0, \bar{\alpha})\} \\ \text{(exact sau inexact cu testul lui Wolfe)} \\ x_{k+1} = x_k + \alpha_k d_k \\ g_{k+1} = \nabla f(x_{k+1}) \\ \beta_k^{FR} = \dfrac{g_{k+1}^T g_k}{g_k^T g_k} \\ d_{k+1} = -g_{k+1} + \beta_k^{FR} d_k \\ k = k + 1; \end{cases}$$

Se pune problema dacă $\boldsymbol{d_k}$ sunt direcţii de descreştere?

$$\boldsymbol{d_k = -g_k + \beta_{k-1} d_{k-1}}$$

$$\boldsymbol{g_k^T d_k = -g_k^T g_k + \beta_{k-1} g_k^T d_{k-1}}$$

Dacă se foloseşte ajustarea pasului exactă:
$\boldsymbol{\alpha_{k-1}}$ este punct de minim local pentru $\boldsymbol{f}$ pe direcţia $\boldsymbol{d_{k-1}}$ prin urmare $\boldsymbol{g_k^T d_{k-1} = 0}$ $\boldsymbol{(g_k = \nabla f(x_k))}$.
$\boldsymbol{\Rightarrow \; g_k^T d_k = -g_k^T g_k = -\| g_k \|_2^2 < 0 \Rightarrow d_k}$ direcţie de descreştere

Dacă se foloseşte ajustarea pasului inexactă am putea avea $g_k^T d_k > 0$ (d_k direcţie de creştere!!) dar folosind testul lui Wolfe deducem:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \varepsilon \alpha_k g_k^T d_k$$

$$|\nabla f(x_k + \alpha_k d_k)^T d_k| \leq (1 - \varepsilon) |g_k^T d_k|$$

$$\varepsilon \in \left(0, \frac{1}{2}\right) \Rightarrow g_k^T d_k < 0$$

Metoda Polak-Ribière

- variantă a metodei Fletcher-Reeves

$$\beta_k^{PR} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k}$$

Dacă se face ajustarea pasului inexactă cu testul lui Wolfe nu putem deduce că d_k sunt direcţii de descreştere.

Se foloseşte $\beta_k^+ = \max\{\beta_k^{PR}, 0\}$ şi un test Wolfe adaptat pentru a obţine d_k direcţii de descreştere.

Varianta Hestenes-Stiefel

$$\beta_k^{HS} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{(g_{k+1} - g_k)^T d_k}$$

Precondiţionare

Se consideră norma:

$$\| x \|_A = \sqrt{(Ax, x)_{\mathbb{R}^n}}$$

Evaluarea erorii în metoda pantei maxime:

$$\| x^{(k)} - x^* \|_A \leq \left(\frac{k(A)-1}{k(A)+1} \right)^k \| x^{(0)} - x^* \|_A$$

$k(A) = \frac{\lambda_n}{\lambda_1}$ – **numărul de condiţionare spectral**

$0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ **valorile proprii ale matricii** A

Avem convergenţă rapidă dacă numărul de condiţionare spectrală al matricei $\boldsymbol{A}$ este apropiat de **1** ($\boldsymbol{k(A)\geq 1}$ întotdeauna).

Ideea precondiţionării este de a transforma sistemul $\boldsymbol{Ax=b}$ astfel încât să îmbunătăţim proprietăţile spectrale.

$$\boldsymbol{Ax=b} \quad \rightleftarrows \quad \boldsymbol{\tilde{A}x=\tilde{b}} \quad \textbf{, cu} \quad \boldsymbol{k(\tilde{A}) << k(A)}$$

Precondiţionare

$$Ax = b \rightarrow M^{-1}Ax = M^{-1}b \text{ (la stânga)}$$

$$\rightarrow AM^{-1}y = b \ , \ x = M^{-1}y \text{ (la dreapta)}$$

$$\rightarrow M_1^{-1}AM_2^{-1}y = M_1^{-1}b \ , \ x = M_2^{-1}\ y \text{ (split)}, \ M = M_1M_2$$

cu M matrice nesingulară , M **"≈"** A. Matricea M sau M^{-1} poartă numele de ***matrice de precondiţionare***.

Cum trebuie să alegem matricea $\boldsymbol{M}$?

- sistemul precondiţionat ($\tilde{\boldsymbol{A}}\boldsymbol{x} = \tilde{\boldsymbol{b}}$) să fie uşor de rezolvat (convergenţă rapidă)

- matricea de precondiţionare să fie economic de construit şi aplicat – ietraţiile să nu fie costisitor de construit

Matricea de precondiţionare Jacobi

$$M = \mathbf{diag}[a_{11}, a_{22}, \ldots, a_{nn}]$$

Matrice de precondiţionare SSOR

$$M = (D + L)\, D^{-1}\, (D + L)^T \quad (A = L + D + L^T)$$

$$M(\omega) = \frac{1}{2-\omega}\left(\frac{1}{\omega}D + L\right)\left(\frac{1}{\omega}D\right)^{-1}\left(\frac{1}{\omega}D + L\right)^T, \ \omega \in (0,2)$$

Pentru ω - optimal, în anumite cazuri:

$$k(M(\omega_{opt})^{-1} A) = O(\sqrt{k(A)})$$

(ω_{opt} - foarte costisitor de calculat)