Cap.2. Sisteme de ecuaţii algebrice liniare - metode directe (I)

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" Bucureşti, Facultatea de Inginerie Electrică, Departamentul de Electrotehnică

Suport didactic pentru disciplina *Metode numerice*, 2017-2018

Cuprins

Formularea problemei

Sistem de n ecuaţii algebrice liniare cu n necunoscute:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n. \end{cases} \tag{1}$$

Formularea problemei

Se dă matricea coeficienţilor

$$\mathbf{A} = \left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array} \right] \in \mathbb{R}^{n \times n} \tag{2}$$

şi vectorul termenilor liberi

$$\mathbf{b} = \left[\begin{array}{cccc} b_1 & b_2 & \cdots & b_n \end{array} \right]^T \in \mathbb{R}^n, \tag{3}$$

se cere să se rezolve sistemul

$$\mathbf{Ax} = \mathbf{b}, \tag{4}$$

unde **x** este soluţia

$$\mathbf{x} = \left[\begin{array}{cccc} x_1 & x_2 & \cdots & x_n \end{array} \right]^T \in \mathbb{R}^n. \tag{5}$$

Buna formulare matematică

Problema este bine formulată din punct de vedere matematic (soluţia există şi este unică)

$\Leftrightarrow$

matricea **A** este nesingulară (are determinantul nenul).
Se scrie formal:

$$"\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}"$$

trebuie citită ca:
"$\mathbf{x}$ *este soluţia sistemului algebric liniar* $\mathbf{Ax} = \mathbf{b}$"
şi **NU** "*se calculează inversa matricei* **A** *care se înmulţeşte cu vectorul* **b**".

Condiţionarea problemei

Condiţionarea

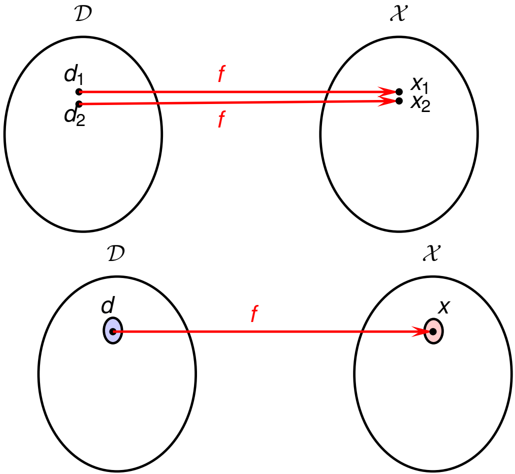
se referă la comportarea problemei matematice la perturbaţii ale datelor.

Problemă matematică f formulată explicit:

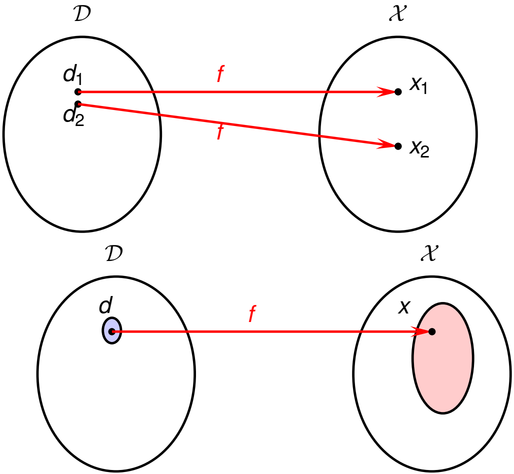
$$\text{Fie} \quad f : \mathcal{D} \to \mathcal{X} \quad \text{şi} \quad \mathbf{d} \in \mathcal{D}.$$

$$\text{Să se găsească} \quad \mathbf{x} \in \mathcal{X} \quad \text{astfel încât} \quad f(\mathbf{d}) = \mathbf{x}. \qquad (6)$$

Reprezentări intuitive - problemă bine condiţionată

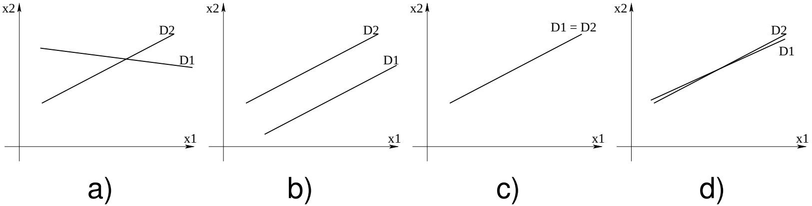


Reprezentări intuitive - problemă prost condiţionată



Condiţionarea - intuitiv ($n = 2$)

Nu orice problemă de rezolvare a unui sistem de ecuaţii algebrice liniare care este bine formulată matematic este şi bine condiţionată.



a) Problemă matematică bine formulată şi bine condiţionată. b) Problemă matematică prost formulată (nu există soluţie). c) Problemă matematică prost formulată (are o infinitate de soluţii). d) Problemă matematică bine formulată şi slab condiţionată.

Numărul de condiţionare

Fie

$$\mathbf{Ax} = \mathbf{b}, \tag{7}$$

unde $\mathbf{x}$ este soluţia exactă şi presupunem o perturbaţie a soluţiei $\mathbf{x} + \mathbf{e_x}$, corespunzătoare unei perturbaţii a datelor $\mathbf{b} + \mathbf{e_b}$:

$$\mathbf{A}(\mathbf{x} + \mathbf{e_x}) = \mathbf{b} + \mathbf{e_b}, \tag{8}$$

$\Rightarrow$

$$\mathbf{Ae_x} = \mathbf{e_b}. \tag{9}$$

Notăm erorile relativă a soluţiei şi a datelor:

$$\varepsilon_x = \frac{\|\mathbf{e_x}\|}{\|\mathbf{x}\|}, \quad \varepsilon_b = \frac{\|\mathbf{e_b}\|}{\|\mathbf{b}\|}. \tag{10}$$

$\Rightarrow$

$$\mathbf{e_x} = \mathbf{A}^{-1}\mathbf{e_b} \Rightarrow \|\mathbf{e_x}\| = \|\mathbf{A}^{-1}\mathbf{e_b}\| \leq \|\mathbf{A}^{-1}\|\|\mathbf{e_b}\|. \tag{11}$$

Numărul de condiţionare

$$\|\mathbf{b}\| = \|\mathbf{Ax}\| \leq \|\mathbf{A}\|\|\mathbf{x}\| \quad \Rightarrow \quad \|\mathbf{x}\| \geq \frac{\|\mathbf{b}\|}{\|\mathbf{A}\|}. \tag{12}$$

Un majorant pentru eroarea asupra soluţiei

$$\varepsilon_x = \frac{\|\mathbf{e_x}\|}{\|\mathbf{x}\|} \leq \frac{\|\mathbf{A}^{-1}\|\|\mathbf{e_b}\|}{\frac{\|\mathbf{b}\|}{\|\mathbf{A}\|}} = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|\varepsilon_b. \tag{13}$$

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\| \tag{14}$$

număr de condiţionare la inversare al matricei **A.**

$$\varepsilon_x \leq \kappa(\mathbf{A})\varepsilon_b, \tag{15}$$

Numărul de condiţionare - proprietăţi

- Numărul de condiţionare este întotdeauna supraunitar $\kappa(\mathbf{A}) \geq 1$:

$$1 = \|\mathbf{I}\| = \|\mathbf{A}\mathbf{A}^{-1}\| \leq \|\mathbf{A}\|\|\mathbf{A}^{-1}\| = \kappa(\mathbf{A}). \qquad (16)$$

  Cazul cel mai favorabil: $n_A = 1$ şi $\varepsilon_x = \varepsilon_b$. (matrice ortogonală)

- Numărul de condiţionare este o proprietate a matricei şi nu are legătură nici cu metoda de rezolvare propriu-zisă, nici cu erorile de rotunjire care apar în mediul de calcul.
  În practică:
  Dacă $\kappa(\mathbf{A}) > 1/\mathtt{eps}$ problema se consideră slab condiţionată.

Clasificarea metodelor

1. **Metode directe** - găsesc soluţia teoretică a problemei într-un număr finit de paşi. (Gauss, factorizare LU)
2. **Metode iterative** - generează un şir de aproximaţii ale soluţiei care se doreşte a fi convergent către soluţia exactă.
    - **staţionare:** Jacobi, Gauss-Seidel, SOR, SSOR
    - **nestaţionare (semiiterative):** gradienţi conjugaţi (GC), reziduu minim (MINRES), reziduu minim generalizat (GMRES), gradienţi biconjugaţi (BiGC), etc.

Ideea metodei Gauss

$$\mathbf{Ax} = \mathbf{b} \quad \underbrace{\Leftrightarrow}_{\text{eliminare}} \quad \mathbf{Ux} = \mathbf{b}' \quad \underbrace{\Rightarrow}_{\text{subst.regresivă}} \quad \mathbf{x} = \mathbf{U}^{-1}\mathbf{b}'. \tag{17}$$

Un exemplu simplu

$$\left\{ \begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ -2x_1 & + & 3x_2 & + & x_3 & = & 0, \\ 4x_1 & - & x_2 & - & 3x_3 & = & -2. \end{array} \right. \tag{18}$$

$$\left\{ \begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ & & 7x_2 & - & x_3 & = & -2, \\ & & -9x_2 & + & x_3 & = & 2. \end{array} \right. \tag{19}$$

$$\left\{ \begin{array}{rcrcrcr} x_1 & + & 2x_2 & - & x_3 & = & -1, \\ & & 7x_2 & - & x_3 & = & -2, \\ & & & - & 2/7x_3 & = & -4/7. \end{array} \right. \tag{20}$$

$$\begin{array}{l} x_3 = (-4/7)/(-2/7) = 2, \\ x_2 = (-2 + x_3)/7 = 0, \\ x_1 = -1 - 2x_2 + x_3 = 1. \end{array} \tag{21}$$

Algoritmul metodei Gauss - etapa de eliminare

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \rightarrow \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix} \rightarrow \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{bmatrix} \rightarrow \cdots$$

$$\mathbf{A}_0 \qquad \mathbf{A}_1 \qquad \mathbf{A}_2 \qquad \cdots$$

Eliminare în metoda Gauss: pentru un sistem de dimensiune n există $n-1$ sub-etape de eliminare. La final matricea este superior triunghiulară. Matricea iniţială este notată $\mathbf{A}_0$ iar matricea superior triunghiulară obţinută este notată $\mathbf{A}_{n-1}$. În realitate, transformările sunt memorate "în loc", în acelaşi tablou bidimensional.

Algoritmul metodei Gauss - etapa de eliminare

Modificarea ecuaţiei a doua în prima sub-etapă de eliminare poate fi descrisă astfel:

; anularea elementului a_{21}
$p = -a_{21}/a_{11}$; element de multiplicare
pentru $j = 1, n$; parcurge coloanele
    $a_{2j} = a_{2j} + pa_{1j}$
•
$b_2 = b_2 + pb_1$

$2 \rightarrow i$ inserată într-un ciclu cu contor

Algoritmul metodei Gauss - etapa de eliminare

Prima sub-etapă de eliminare:

```
; prima sub-etapă de eliminare
pentru i = 2, n                    ; parcurge liniile
      p = -a_i1 / a_11             ; element de multiplicare
      pentru j = 2, n              ; parcurge coloanele
            a_ij = a_ij + p a_1j
      •
      b_i = b_i + p b_1
•
```

$$\underline{\texttt{pentru}}\; i = 2, n \qquad \text{; parcurge liniile}$$
$$p = -a_{i1}/a_{11} \qquad \text{; element de multiplicare}$$
$$\underline{\texttt{pentru}}\; j = 2, n \qquad \text{; parcurge coloanele}$$
$$a_{ij} = a_{ij} + pa_{1j}$$
$$b_i = b_i + pb_1$$

OBS: În ciclul în j contorul începe cu valoarea 2.
$1 \to k, 2 \to k + 1$ inserate într-un ciclu cu contor.

Algoritmul metodei Gauss - etapa de eliminare

Secvenţa de cod corespunzătoare etapei de eliminare

```
; etapa de eliminare din metoda Gauss
pentru k = 1, n − 1
    pentru i = k + 1, n              ; parcurge liniile
        p = −a_ik / a_kk             ; element de multiplicare
        pentru j = k + 1, n          ; parcurge coloanele
            a_ij = a_ij + p a_kj
        •
        b_i = b_i + p b_k
    •
•
```

Algoritmul metodei Gauss - substituţie regresivă

$$\left\{ \begin{array}{rrrl} a_{11}x_1 + & a_{12}x_2 + \cdots + & a_{1n}x_n & = b_1, \\ & a_{22}x_2 + \cdots + & a_{2n}x_n & = b_2, \\ & \cdots & & \\ & & a_{nn}x_n & = b_n. \end{array} \right. \tag{22}$$

$$x_n = b_n / a_{nn}, \tag{23}$$

$$a_{ii}x_i + a_{i,i+1}x_{i+1} + \cdots + a_{in}x_n = b_i, \tag{24}$$

$\Rightarrow$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}. \tag{25}$$

Algoritmul metodei Gauss - substituţie regresivă

$$x_n = b_n/a_{nn}, \tag{26}$$

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}. \tag{27}$$

; etapa de retrosubstituţie

$x_n = b_n/a_{nn}$

<u>`pentru`</u> $i = n-1, 1, -1$

  $s = 0$

  <u>`pentru`</u> $j = i+1, n$

    $s = s + a_{ij}x_j$

  •

  $x_i = (b_i - s)/a_{ii}$

•

Algoritmul metodei Gauss

```
procedură Gauss(n, a, b, x)
; rezolvă sistemul algebric liniar ax = b prin metoda Gauss
întreg n                                ; dimensiunea sistemului
tablou real a[n][n]                     ; matricea coeficienţilor - indici de la 1
tablou real b[n]                        ; vectorul termenilor liberi
tablou real x[n]                        ; vectorul soluţie
întreg i, j, k
real p, s
; etapa de eliminare
pentru k = 1, n − 1                     ;
        ; aici se poate introduce pivotarea
        pentru i = k + 1, n             ; parcurge liniile
                p = −a_ik / a_kk        ; element de multiplicare
                pentru j = k + 1, n     ; parcurge coloanele
                        a_ij = a_ij + p a_kj
                •
                b_i = b_i + p b_k
        •
•
```

Algoritmul metodei Gauss

```
; etapa de retrosubstituţie
x_n = b_n/a_nn
pentru i = n − 1, 1, −1
        s = 0
        pentru j = i + 1, n
                s = s + a_ij x_j
        •
        x_i = (b_i − s)/a_ii
•
retur
```

Algoritmul poate fi îmbunătăţit prin folosirea la fiecare etapă de eliminare a unei strategii de pivotare.

Evaluarea algoritmului

Din punct de vedere al timpului de calcul:

$$\begin{aligned} T_e &= \sum_{k=1}^{n-1}[2(n-k)+3](n-k) \approx \sum_{k=1}^{n-1} 2(n-k)^2 = \\ &= 2\frac{(n-1)n(2n-1)}{6} \approx \frac{2n^3}{3}. \end{aligned} \tag{28}$$

$$T_s = \sum_{i=1}^{n-1}[2(n-i)+2] \approx \sum_{i=1}^{n-1}[2(n-i)] = 2\frac{n(n-1)}{2} \approx n^2. \tag{29}$$

$T_{\mathrm{Gauss}} = \mathrm{O}(2n^3/3 + n^2) = \mathrm{O}(2n^3/3)$ - costisitor

Din punct de vedere al necesarului de memorie:

$M = n^2 + 2n + 2 \Rightarrow M = \mathrm{O}(n^2)$

Evaluarea algoritmului

Din punct de vedere al erorilor: erorile inerente, erori de rotunjire.

- Cu cât sistemul este de dimensiune mai mare, cu atât erorile acumulate datorită rotunjirii cresc.
- O diminuare a erorilor de rotunjire se poate obţine dacă se includ în algoritm strategii de pivotare.

Din punct de vedere al stabilităţii: algoritmul Gauss poate să nu fie stabil chiar dacă problema matematică este bine formulată şi bine condiţionată (numărul de condiţionare al matricei **A** este mic). Acest lucru se întâmplă atunci când numărul de condiţionare al matricei **U** este mare. Remediul îl constituie în acest caz pivotarea.

Strategii de pivotare

Pivoţi

Elementele diagonale a_{kk} obţinute în urma etapei de eliminare.

Determinantul sistemului = produsul pivoţilor.

$\Rightarrow$

Problema este bine formulată matematic $\Leftrightarrow$ toţi pivoţii sunt nenuli.

Elementele de multiplicare: $p = -a_{ik}/a_{kk}$. a_{kk} = pivot,

Pivotare

Operaţie de permutare care urmăreşte obţinerea valorilor nenule pentru pivoţi.
Trebuie făcută înainte de calculul multiplicatorului.

Strategii de pivotare

Strategii de pivotare:

- Pivotarea pe linii (parţială)
- Pivotarea pe coloane
- Pivotarea totală (completă sau maximală)
- Pivotarea diagonală

$$
\begin{bmatrix}
X & X & X & X & X & X \\
0 & X & X & X & X & X \\
0 & 0 & a_{kk} & * & * & * \\
0 & 0 & {\color{red}*} & * & * & * \\
0 & 0 & {\color{red}*} & * & * & * \\
0 & 0 & {\color{red}*} & * & * & *
\end{bmatrix}
\quad
\begin{bmatrix}
X & X & X & X & X & X \\
0 & X & X & X & X & X \\
0 & 0 & a_{kk} & {\color{red}*} & {\color{red}*} & {\color{red}*} \\
0 & 0 & * & * & * & * \\
0 & 0 & * & * & * & * \\
0 & 0 & * & * & * & *
\end{bmatrix}
$$

$$
\begin{bmatrix}
X & X & X & X & X & X \\
0 & X & X & X & X & X \\
0 & 0 & a_{kk} & {\color{red}*} & {\color{red}*} & {\color{red}*} \\
0 & 0 & {\color{red}*} & {\color{red}*} & {\color{red}*} & {\color{red}*} \\
0 & 0 & {\color{red}*} & {\color{red}*} & {\color{red}*} & {\color{red}*} \\
0 & 0 & {\color{red}*} & {\color{red}*} & {\color{red}*} & {\color{red}*}
\end{bmatrix}
\quad
\begin{bmatrix}
X & X & X & X & X & X \\
0 & X & X & X & X & X \\
0 & 0 & a_{kk} & * & * & * \\
0 & 0 & * & {\color{red}*} & * & * \\
0 & 0 & * & * & {\color{red}*} & * \\
0 & 0 & * & * & * & {\color{red}*}
\end{bmatrix}
$$

Zona de căutare a pivotului. Cu X sunt marcate elementele nenule ale căror valori nu se vor mai modifica.

Algoritmul pivotării pe linii

```
p = 0
pentru i = k, n                         ; parcurge coloana k
      dacă |a_ik| > p atunci
            l = i                       ; memorează poziţia potenţialului pivot
            p = |a_ik|
      •
•
dacă p = 0 atunci
      scrie "problema este prost formulată matematic"
altfel
      pentru j = k, n                   ; permută linia l cu linia k
            p = a_kj
            a_kj = a_lj
            a_lj = p
      •
      p = b_k                           ; permută termenii liberi
      b_k = b_l
      b_l = p
•
```

Avantajele pivotării

Pivotarea

1. necesară dacă pe parcursul algoritmului se întâlneşte un pivot nul.
2. efect benefic asupra stabilităţii şi acurateţii.

Avantajele pivotării

Exemplu

$$\begin{cases} 10^{-20}x + y & = & 1, \\ x + y & = & 0, \end{cases} \tag{30}$$

soluţia corectă $(x, y) \approx (-1, 1)$.
Gauss şi presupunem că eps = 10^{-16}:

$$\begin{cases} 10^{-20}x + y & = & 1, \\ (1 - 10^{20})y & = & -10^{20}. \end{cases} \tag{31}$$

$$\begin{cases} 10^{-20}x + y & = & 1, \\ -10^{20}y & = & -10^{20}. \end{cases} \tag{32}$$

Rezultatul final: $(x, y) = (0, 1)$ extrem de eronat.
Explicaţie: $\kappa(\mathbf{A}) \approx 2.6$, dar $\kappa(\mathbf{U}) = 10^{40}$!

Metoda Gauss - Concluzii

- Este o metodă directă - găseşte soluţia teoretică a problemei într-un număr finit de paşi.
- Calculele sunt afectate de erori de rotunjire $\Rightarrow$ nu se obţine soluţia exactă, ci o aproximare a ei.
- Se transformă sistemului de ecuaţii într-unul echivalent din punct de vedere al soluţiei (Δ sup.), mult mai uşor de rezolvat (subs. regr.).
- Pivotarea: esenţială pentru a asigura pivoţi nenuli; utilă pentru a creşte stabilitatea algoritmului şi acurateţea soluţiei.

Metoda Gauss - Concluzii

- Pivotarea parţială are un efort de implementare nesemnificativ.
- Pivotarea totală este rareori aplicată deoarece duce la o creştere semnificativă a timpului de calcul, nerealizând decât o îmbunătăţire nesemnificativă a acurateţii soluţiei.
- Dezavantajul metodei Gauss: în anumite situaţii, efortul de generare a problemei echivalente (eliminarea) este mare sau, necesarul de memorie poate deveni extrem de mare.

Lectura obligatorie pentru această săptămână

- Cap.3 din

[1] Gabriela Ciuprina, Mihai Rebican, Daniel Ioan - Metode numerice in ingineria electrica - Indrumar de laborator pentru studentii facultatii de Inginerie electrica, Editura Printech, 2013, disponibil la http://mn.lmn.pub.ro/indrumar/IndrumarMN_Printech2013.pdf