CURS #2

CONȚINUTUL CURSULUI #2:

I. Metode de aproximare a soluțiilor ecuațiilor neliniare.
- I.3. Metoda secantei.
- I.4. Metoda poziției false.

II. Metode numerice de rezolvare a sistemelor liniare.
- II.1. Metode directe de rezolvare a sistemelor de ecuații liniare.
- II.1.1. Sisteme liniare superior triunghiulare.
- II.1.2. Metoda Gauss fără pivotare.
- II.1.3. Metoda Gauss cu pivotare parțială.
- II.1.4. Metoda Gauss cu pivotare totală.

I.3. Metoda secantei.

La pasul k, aproximarea x_k a soluției exacte x^* a ecuației $f(x) = 0, x \in [a, b]$ se obține prin intersecția cu axa Ox a secantei AB la graficul lui f, prin punctele $A\big(x_{k-1}, f(x_{k-1})\big)$ și $B\big(x_{k-2}, f(x_{k-2})\big)$. Prin urmare, nu se mai folosește tangenta la graficul lui f, deci nu mai este necesar caculul derivatei lui f.

$$AB : \frac{x - x_{k-1}}{x_{k-2} - x_{k-1}} = \frac{y - f(x_{k-1})}{f(x_{k-2}) - f(x_{k-1})} \tag{1}$$

$$\{x_k\} = AB \cap Ox \Rightarrow \frac{x_k - x_{k-1}}{x_{k-2} - x_{k-1}} = -\frac{f(x_{k-1})}{f(x_{k-2}) - f(x_{k-1})} \Rightarrow$$

$$x_k = x_{k-1} - f(x_{k-1}) \frac{x_{k-1} - x_{k-2}}{f(x_{k-1}) - f(x_{k-2})}$$

sau

$$x_k = \frac{x_{k-2} f(x_{k-1}) - x_{k-1} f(x_{k-2})}{f(x_{k-1}) - f(x_{k-2})}, k \geq 2 \tag{2}$$

unde $x_0, x_1 \in [a, b]$

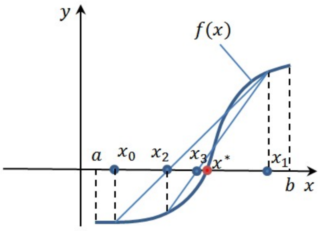

Figure: Metoda secantei

Teorema ((I.3.) Convergența metodei secantei)

Presupunem că $f \in C^1([a, b]), f(a)f(b) < 0, f'(x) \neq 0, \forall x \in [a, b]$. Atunci $\exists! x^*$ astfel încât $f(x^*) = 0$. Mai mult, $\exists \delta > 0$, astfel încât, dacă $x_0, x_1 \in [x^* - \delta, x^* + \delta] \subseteq [a, b]$, atunci șirul $(x_k)_{k \geq 0}$ construit prin metoda secantei rămâne în intervalul $[x^* - \delta, x^* + \delta]$ și converge către x^*.

Demonstrație: Existența și unicitatea este asigurată de faptul că $f(a)f(b) < 0$ și $f'(x) \neq 0, \forall x \in [a, b]$.
Deoarece $f'(x^*) \neq 0$, putem considera $f'(x^*) = \mu > 0$.
Din continuitatea derivatei f' rezultă că, pentru $\forall \varepsilon > 0, \exists \delta > 0$ astfel încât

$$|f'(x) - f'(x^*)| < \varepsilon, \quad \forall x \in [x^* - \delta, x^* + \delta] \subseteq [a, b] \tag{3}$$

sau

$$-\varepsilon + \mu < f'(x) < \varepsilon + \mu$$

Fie $\varepsilon = \frac{\mu}{4}$, atunci

$$\frac{3}{4}\mu < f'(x) < \frac{5}{4}\mu, \quad \forall x \in [x^* - \delta, x^* + \delta] \tag{4}$$

Conform metodei secantei

$$x_{k+1} = x_k - f(x_k)\,\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \tag{5}$$

Din dezvoltărea în serie Taylor a funcției f în vecinătatea punctului x_k și evalută în x^* rezultă:

$$f(x^*) = f(x_k) + (x^* - x_k)f'(\xi_k), \quad \xi_k \in [x^*, x_k]$$

sau

$$f(x_k) = -(x^* - x_k)f'(\xi_k) \tag{6}$$

Mai mult, aplicând teorema Lagrange pe intervalul $[x_{k-1}, x_k]$ rezultă că $\exists \eta_k \in (x_{k-1}, x_k)$ astfel încât:

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = f'(\eta_k) \tag{7}$$

Din (7) în (5) rezultă:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(\eta_k)}$$

sau

$$x^* - x_{k+1} = x^* - x_k + \frac{f(x_k)}{f'(\eta_k)}, \tag{8}$$

iar conform cu (6) avem:

$$x^* - x_{k+1} = x^* - x_k - \frac{(x^* - x_k)f'(\xi_k)}{f'(\eta_k)}$$

sau

$$x^* - x_{k+1} = (x^* - x_k)\Big(1 - \frac{f'(\xi_k)}{f'(\eta_k)}\Big) \tag{9}$$

Din (4) rezultă următoarea estimare:

$$-\frac{2}{3} < 1 - \frac{f'(x)}{f'(y)} < \frac{2}{3} \tag{10}$$

Fie $x_0, x_1 \in [x^* - \delta, x^* + \delta]$. Presupunem că $x_k \in [x^* - \delta, x^* + \delta]$ și vom demonstra că și $x_{k+1} \in [x^* - \delta, x^* + \delta]$. Se observă că $\eta_k, \xi_k \in [x^* - \delta, x^* + \delta]$, iar conform relației (10), din (9) rezultă

$$|x^* - x_{k+1}| \le \frac{2}{3}|x^* - x_k| \le \frac{2}{3}\delta \tag{11}$$

Astfel că, $x_{k+1} \in [x^* - \delta, x^* + \delta]$, deci șirul $(x_k)_{k\ge0}$ rămâne în intervalul $[x^* - \delta, x^* + \delta]$. Mai mult,

$$|x^* - x_{k+1}| \le \frac{2}{3}|x^* - x_k| \le \ldots \le \Big(\frac{2}{3}\Big)^{k+1}|x^* - x_0| \tag{12}$$

rezultă că șirul $(x_k)_{k\ge0}$ este convergent la x^*. □

Obs.: Se poate arăta că

$$lim_{k\to\infty}\frac{|x^* - x_{k+1}|}{|x^* - x_k|^r} = \alpha, \alpha > 0 \tag{13}$$

unde $r = \frac{1}{2}(1 + \sqrt{5}) \approx 1,62$, astfel că metoda secantei este mai rapidă decât metoda liniară dar mai lentă decât cea pătratică.

Din punct de vedere computațional valorile inițiale x_0, x_1 se aleg din vecinătatea soluției x^*, astfel încât la fiecare iterație se testează ca termenul x_k să rămână în intervalul $[a, b]$. Pentru optimizarea metodei se va alege intervalul maxim $[a, b]$ pe care funcția f este definită, nu-și schimbă monotonia (i.e. $f'(x) \neq 0, \forall x \in [a, b]$) și $f(a)f(b) < 0$.

ALGORITM (Metoda secantei)

Date de intrare: $f, a, b, x_0, x_1, \varepsilon$;

Date de ieșire: x_{aprox};

STEP1 : Se aleg $x_0, x_1 \in [a, b]$; $k = 1$;

STEP2 : while $\dfrac{|x_k - x_{k-1}|}{|x_{k-1}|} \ge \varepsilon$ do

$k = k + 1$;

$x_k = \dfrac{x_{k-2}f(x_{k-1}) - x_{k-1}f(x_{k-2})}{f(x_{k-1}) - f(x_{k-2})}$;

if $x_k < a$ or $x_k > b$ then

OUTPUT('Introduceţi alte valori pentru x_0, x_1');

break;

endif

endwhile;

$x_{aprox} = x_k$;

I.4. Metoda poziţiei false

Metoda poziţiei false construieşte şirurile $(a_k)_{k\geq 0}, (b_k)_{k\geq 0}, (x_k)_{k\geq 0}$ conform următoarei scheme grafice: la pasul k, aproximarea x_k a soluţiei exacte x^* a ecuaţiei $f(x) = 0$ se obţine prin intersecţia dreptei AB cu axa Ox, unde $A(a_k, f(a_k)), B(b_k, f(b_k))$. Intervalul $[a_k, b_k]$ se construieşte conform metodei bisecţiei.

$$AB : \frac{x - a_k}{b_k - a_k} = \frac{y - f(a_k)}{f(b_k) - f(a_k)} \tag{14}$$

$$\{x_k\} = AB \cap Ox \Rightarrow \frac{x_k - a_k}{b_k - a_k} = \frac{-f(a_k)}{f(b_k) - f(a_k)} \Rightarrow \tag{15}$$

$$x_k = a_k - f(a_k)\frac{b_k - a_k}{f(b_k) - f(a_k)} \tag{16}$$

sau

$$x_k = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)} \tag{17}$$

Avem astfel următoarea schemă generală:

$$(a_k, b_k, x_k) \tag{18}$$

$$= \begin{cases} a_k = a_{k-1}, b_k = b_{k-1}, x_k = x_{k-1}, & \text{dacă } f(x_{k-1}) = 0 \\ a_k = a_{k-1}, b_k = x_{k-1}, x_k = \dfrac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}, & \text{dacă } f(a_{k-1})f(x_{k-1}) < 0 \\ a_k = x_{k-1}, b_k = b_{k-1}, x_k = \dfrac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}, & \text{dacă } f(a_{k-1})f(x_{k-1}) > 0, \end{cases} \tag{19}$$

unde $a_0 = a, b_0 = b, x_0 = \dfrac{a_0 f(b_0) - b_0 f(a_0)}{f(b_0) - f(a_0)}$.

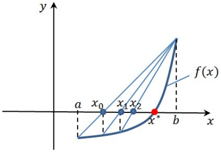

Figure: Metoda poziţiei false

Teorema (I.4. Teorema de convergenţă a metodei poziţiei false)

Presupunem că $f \in C^2([a, b]), f(a)f(b) < 0$ şi f', f'' nu se anulează pe $[a, b]$. Atunci ecuaţia $f(x) = 0$ are o soluţie unică $x^* \in (a, b)$, iar şirul $(x_k)_{k\geq 0}$ construit prin metoda poziţiei false converge la x^*.

ALGORITM (Metoda poziţiei false)

Date de intrare: f, a, b, ε; **Date de ieşire:** x_{aprox};

STEP1 : $k = 0$; $a_0 = a$; $b_0 = b$; $x_0 = \dfrac{a_0 f(b_0) - b_0 f(a_0)}{f(b_0) - f(a_0)}$;

STEP2 : do

$k = k + 1$;

if $f(x_{k-1}) = 0$ then

$x_k = x_{k-1}$;

break;

elseif $f(a_{k-1})f(x_{k-1}) < 0$ then

$a_k = a_{k-1}$; $b_k = x_{k-1}$; $x_k = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}$;

elseif $f(a_{k-1})f(x_{k-1}) > 0$ then

$a_k = x_{k-1}$; $b_k = b_{k-1}$; $x_k = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}$;

endif

while $\dfrac{|x_k - x_{k-1}|}{|x_{k-1}|} \geq \varepsilon$; $x_{aprox} = x_k$.

II. Metode numerice de rezolvare a sistemelor liniare.

II.1.Metode directe de rezolvare a sistemelor de ecuații liniare.

II.1.1. Sisteme liniare superior triunghiulare

Definiția (II.1.)

a) *Matricea* $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$ *se numește matrice superior triunghiulară dacă și numai dacă elementele sub diagonala principală sunt nule, i.e.* $a_{ij} = 0, \forall i > j$;

b) *Un sistem liniar a cărui matrice asociată este superior triunghiulară se numește sistem superior triunghiular.*

Fie sistemul liniar $Ax = b, A \in \mathcal{M}_n(\mathbb{R})$ superior triunghiulară cu $a_{kk} \neq 0, k = \overline{1,n}$ și $b \in \mathbb{R}^n$. Sistemul superior triunghiular $Ax = b$ se scrie sub forma

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k + \ldots + a_{1n}x_n = b_1 & (E_1) \\ \qquad a_{22}x_2 + \ldots + a_{2k}x_k + \ldots + a_{2n}x_n = b_2 & (E_2) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ \qquad\qquad a_{kk}x_k + \ldots + a_{kn}x_n = b_k & (E_k) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ \qquad\qquad\qquad a_{nn}x_n = b_n & (E_n) \end{cases} \tag{20}$$

Din (E_n) rezultă

$$x_n = \frac{b_n}{a_{nn}}. \tag{21}$$

Fie ecuația (E_k): $a_{kk}x_k + \sum_{j=k+1}^{n} a_{kj}x_j = b_k$. Dacă din ultimele $n-k$ ecuații sunt calculate componentele $x_j, j = \overline{k+1,n}$, atunci din (E_k) rezultă

$$x_k = \frac{1}{a_{kk}}\left(b_k - \sum_{j=k+1}^{n} a_{kj}x_j\right) \tag{22}$$

ALGORITM (Metoda substituției descendente)

Date de intrare: $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$; $b \in \mathbb{R}^n$;
Date de ieșire: $x \in \mathbb{R}^n$;

STEP1 : $x_n = \frac{1}{a_{nn}} b_n$; $k = n-1$;

STEP2 : `while` $k > 0$

$$x_k = \frac{1}{a_{kk}}\left(b_k - \sum_{j=k+1}^{n} a_{kj}x_j\right);$$

$k = k - 1$;

`endwhile`

Definim în continuare conform Algoritmului (Metoda substituției descendente) procedura **SubsDesc** având sintaxa $x =$**SubsDesc**(A, b), unde x este soluția sistemului $Ax = b$.

II.1.2. Metoda Gauss fără pivotare.

Fie sistemul liniar $Ax = b, A \in \mathcal{M}_n(\mathbb{R})$, $b \in \mathbb{R}^n$. Sistemul $Ax = b$ se scrie sub forma

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k + \ldots + a_{1n}x_n = b_1 & (E_1) \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k + \ldots + a_{2n}x_n = b_2 & (E_2) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k + \ldots + a_{kn}x_n = b_k & (E_k) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nk}x_k + \ldots + a_{nn}x_n = b_n & (E_n) \end{cases} \tag{23}$$

Se numește **pivot** al matricei $A = (a_{ij})_{i,j=\overline{1,n}}$ orice element de pe diagonala principală a matricei A, i.e. a_{kk}, $k \in \overline{1,n}$.
Metoda Gauss fără pivotare transformă matricea extinsă $\overline{A}$ folosind transformările elementare într-o matrice superior triunghiulară, obținându-se astfel un sistem compatibil cu sistemul inițial.
La fiecare pas $k = \overline{1,n-1}$ al metodei lui Gauss fără pivotare se alege drept pivot corespunzător coloanei k primul element nenul $a_{pk} \neq 0, p \geq k$ de pe coloana k a matricei transformate. Apoi se elimină toate elementele de pe coloana k situate sub pivot (folosind transformările elementare).

ALGORITM (Metoda Gauss fără pivotare)

Date: $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$; $b \in \mathbb{R}^n$;

STEP 1: $A = (A \mid b) = (a_{ij})_{i=\overline{1,n}; j=\overline{1,n+1}}$ `(matricea extinsă);`

STEP 2: `for` $k = 1 : n-1$ `do`

`Se caută primul` p `cu` $k \le p \le n$ `a.î.` $a_{pk} \neq 0$;

`if (nu a fost gasit` p `) then`

`OUTPUT('Sistem incomp. sau sist. comp. nedet.')`

`break;`

`endif`

`if` $p \neq k$ `then`

$L_p \leftrightarrow L_k$ `(schimbă linia` p `cu linia` k`)`

`endif`

`for` $\ell = k+1 : n$ `do`

$$m_{\ell k} = \frac{a_{\ell k}}{a_{kk}};$$

$$L_\ell \leftarrow L_\ell - m_{\ell k} L_k;$$

`endfor`

`endfor`

STEP 3: `if` $a_{nn} = 0$

`OUTPUT('Sistem incomp. sau sist. comp. nedet.')`

STEP 4: $x =$ **SubsDesc**$\left((a_{ij})_{i,j=\overline{1,n}}, (a_{i,n+1})_{i=\overline{1,n}}\right)$

Exemplu 1: Să se rezolve, folosind metoda lui Gauss fără pivotare, sistemul liniar:

$$\begin{cases} x_2 + 2x_3 = 8 \\ x_1 + x_3 = 4 \\ 3x_1 + 2x_2 + x_3 = 10 \end{cases} \tag{24}$$

Matricea extinsă $\overline{A}$ asociată sistemului este:

$$\bar{A} = [A|b] = \left[\begin{array}{ccc|c} 0 & 1 & 2 & 8 \\ 1 & 0 & 1 & 4 \\ 3 & 2 & 1 & 10 \end{array}\right]$$

$k = 1 : a_{21} \neq 0 \Rightarrow p = 2$. Deoarece $p \neq k$ interschimbăm $L_p \leftrightarrow L_k$. Se obține o matrice echivalentă cu matricea $\overline{A}$.

$$\bar{A} \sim \left[\begin{array}{ccc|c} 1 & 0 & 1 & 4 \\ 0 & 1 & 2 & 8 \\ 3 & 2 & 1 & 10 \end{array}\right]$$

Eliminăm toate elementele de pe prima coloană situate sub elementul $a_{11} = 1$ al matricei echivalente. Aplicăm următoarea transformare elementară $L_3 \leftarrow L_3 - \frac{3}{1} L_1$:

$$\bar{A} \sim \left[\begin{array}{ccc|c} 1 & 0 & 1 & 4 \\ 0 & 1 & 2 & 8 \\ 0 & 2 & -2 & -2 \end{array}\right]$$

$k = 2 : a_{22} = 1 \neq 0$. Eliminăm elementul situat sub pivotul curent a_{22} aplicând transformarea $L_3 \leftarrow L_3 - 2L_2$

$$\bar{A} \sim \left[\begin{array}{ccc|c} 1 & 0 & 1 & 4 \\ 0 & 1 & 2 & 8 \\ 0 & 0 & -6 & -18 \end{array}\right]$$

Matricea finală este o matrice superior triunghiulare și reprezintă matricea asociată unui sistem compatibil cu sistemul inițial. Soluția sistemului este: $x_1 = 1, x_2 = 2, x_3 = 3$.

II.1.3. Metoda Gauss cu pivotare parțială.

La fiecare pas $k = \overline{1, n-1}$ al metodei lui Gauss fără pivotare se alege ca pivot corespunzător coloanei k elementul a_{pk} cu valoarea absolută cea mai mare de pe coloana k, aflat sub sau pe diagonala principală a matricei curente A, i.e.

$$|a_{pk}| = \max_{j=\overline{k,n}} |a_{jk}|, \quad p \in \overline{k,n} \tag{25}$$

Se interschimbă linia L_p cu L_k.

ALGORITM (Metoda Gauss cu pivotare parțială)

Date: $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$; $b \in \mathbb{R}^n$;

STEP 1: $A = (A \mid b) = (a_{i,j})_{i=\overline{1,n}, j=\overline{1,n+1}}$ (matricea extinsă)

STEP 2: for $k = 1 : n-1$ do

Determină primul indice p, $(k \le p \le n)$

a.î. $|a_{pk}| = \max_{j=\overline{k,n}} |a_{jk}|$

if $a_{pk} = 0$ then

OUTPUT('Sist. incomp. sau comp. nedet.')

break;

endif

if $p \neq k$ then

$L_p \leftrightarrow L_k$ (schimbă linia p cu linia k)

endif

for $\ell = k+1, n$ do

$m_{\ell k} = \dfrac{a_{\ell k}}{a_{kk}}$;

$L_\ell \leftarrow L_\ell - m_{\ell k} L_k$;

endfor

endfor

STEP 3: if $a_{nn} = 0$

OUTPUT('Sist. incomp. sau comp. nedet.')

STEP 4: $x =$ **SubsDesc**$\left((a_{ij})_{i,j=\overline{1,n}}, (a_{i,n+1})_{i=\overline{1,n}}\right)$

Exemplu 2: Să se rezolve, folosind metoda lui Gauss cu pivotare parțială, sistemul liniar:

$$\begin{cases} x_2 + 2x_3 = 8 \\ x_1 + x_3 = 4 \\ 3x_1 + 2x_2 + x_3 = 10 \end{cases} \tag{26}$$

Matricea extinsă $\bar{A}$ asociată sistemului este:

$$\bar{A} = [A|b] = \left[\begin{array}{ccc|c} 0 & 1 & 2 & 8 \\ 1 & 0 & 1 & 4 \\ 3 & 2 & 1 & 10 \end{array}\right]$$

$k = 1 : |a_{p1}| = \max_{j=\overline{1,3}} |a_{j1}| = |a_{31}| \Rightarrow p = 3$. Interschimbăm $L_3 \leftrightarrow L_1$. Se obține matricea echivalentă cu $\bar{A}$

$$\bar{A} \sim \left[\begin{array}{ccc|c} 3 & 2 & 1 & 10 \\ 0 & 1 & 2 & 8 \\ 1 & 0 & 1 & 4 \end{array}\right]$$

În urma transformării elementare $L_3 \leftarrow L_3 - \dfrac{1}{3} L_1$ se obține:

$$\bar{A} \sim \left[\begin{array}{ccc|c} 3 & 2 & 1 & 10 \\ 0 & 1 & 2 & 8 \\ 0 & -\frac{2}{3} & \frac{2}{3} & \frac{2}{3} \end{array}\right]$$

$k = 2 : |a_{p2}| = \max_{j=\overline{2,3}} |a_{j2}| = |a_{22}| \Rightarrow p = 2$. Alpicăm transformarea elementară $L_3 \leftarrow L_3 - \dfrac{-2/3}{1} = L_3 + \dfrac{2}{3} L_2$

$$\bar{A} \sim \left[\begin{array}{ccc|c} 3 & 2 & 1 & 10 \\ 0 & 1 & 2 & 8 \\ 0 & 0 & 2 & 6 \end{array}\right]$$

Soluția sistemului este: $x_1 = 1, x_2 = 2, x_3 = 3$.

II.1.4. Metoda Gauss cu pivotare totalaă.

La fiecare pas $k = \overline{1, n-1}$ alegem ca pivot elementul curent a_{pm} cu valoarea absolută cea mai mare din submatricea $(a_{ij})_{i,j=\overline{k,n}}$, i.e.

$$\boxed{|a_{pm}| = \max_{i,j=\overline{k,n}} |a_{ij}|, \quad p, m \in \overline{k,n}} \tag{27}$$

Dacă $m \neq k$, atunci interschimbăm coloanele k și m. Dacă $p \neq k$, atunci interschimbăm liniile k și ℓ.

ALGORITM (Metoda Gauss cu pivotare totală)

Date: $A = (a_{ij})_{i,j=\overline{1,n}} \in \mathcal{M}_n(\mathbb{R})$; $b \in \mathbb{R}^n$;

STEP 1: $A = (A \mid b) = (a_{i,j})_{i=\overline{1,n}, j=\overline{1,n+1}}$ (matricea extinsă)

STEP 2: for $k = 1 : n-1$ do

```
        Determină primii indici p, m (k ≤ p, m ≤ n)
        a.î. |a_pm| = max_{i,j=k,n} |a_ij|,
        if a_pm = 0 then
            OUTPUT('Sist. incomp. sau comp.
    nedet.')
            break;
        endif
        if p ≠ k then
            L_ℓ ↔ L_k (schimbă linia p cu linia k)
        endif
```



```
        if m ≠ k then
            C_m ↔ C_k (schimbă coloana m cu coloana k)
        endif
        for ℓ = k+1, n do
            m_ℓk = a_ℓk / a_kk;
            L_ℓ ← L_ℓ − m_ℓk L_k;
        endfor
    endfor
```

STEP 3: if $a_{nn} = 0$

OUTPUT('Sist. incomp. sau comp. nedet.')

STEP 4: $x =$ **SubsDesc**$((a_{ij})_{i,j=\overline{1,n}}, (a_{i,n+1})_{i=\overline{1,n}})$

Obs.: Atunci când se schimbă două coloane în matricea A se va schimba și ordinea necunoscutelor în sistem.

Exercițiu: (II.1.)

a) *Să se construiască în Matlab conform Algoritmilor (metoda substituției descendente) și (metoda Gauss cu pivotare totală), procedurile* **SubsDesc**, *respectiv* **GaussPivTot** *cu următoarele sintaxe:* $[x]=$**SubsDesc**(A, b) *și* $[x]=$**GaussPivTot**(A, b).

b) *Într-un fișier script să se apeleze procedura* **GaussPivTot** *pentru datele* $A = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 3 & 2 & 1 \end{pmatrix}$ *și* $b = \begin{pmatrix} 8 \\ 4 \\ 10 \end{pmatrix}$.

Soluție: Vezi Program II.1.