ERORI DE TRUNCHIERE

Erorile produse ca urmare a calculului în virgulă mobilă sunt erori de trunchiere. Acestea apar ca urmare a faptului că numerele sunt reprezentate cu un **număr finit de zecimale**.

Notaţii

x - valoarea exactă a unei mărimi;

$\bar{x}$ - valoarea aproximativă a aceleiaşi mărimi;

e_x - eroarea de trunchiere.

Cu aceste notaţii, este valabilă relaţia: $x = \bar{x} + e_x$

Eroarea produsă la operaţia de **adunare**: $x + y = \bar{x} + \bar{y} + e_x + e_y \quad \Rightarrow \quad e_{x+y} = e_x + e_y$

Eroarea produsă la operaţia de **scădere**: $x - y = \bar{x} - \bar{y} + e_x - e_y \quad \Rightarrow \quad e_{x-y} = e_x - e_y$

Eroarea produsă la operaţia de **înmulţire**

$$x \cdot y = \left(\overline{x} + e_x\right) \cdot \left(\overline{y} + e_y\right) = \overline{x} \cdot \overline{y} + \overline{x} \cdot e_y + \overline{y} \cdot e_x + e_x \cdot e_y$$

neglijând termenul $e_x \cdot e_y$ obţinem $e_{x \cdot y} = \overline{x} \cdot e_y + \overline{y} \cdot e_x$

Observaţie

La operaţia de înmulţire eroarea produsă depinde nu numai de erorile termenilor care intervin în operaţia respectivă dar şi de valoarea acestora.

Eroarea produsă la operaţia de **împărţire**

$$\frac{x}{y} = \frac{\overline{x} + e_x}{\overline{y} + e_y} = \frac{\overline{x} + e_x}{\overline{y} \cdot \left(1 + \frac{e_y}{\overline{y}}\right)} = \cdot \frac{1}{1 + \frac{e_y}{\overline{y}}} \cdot \frac{\overline{x} + e_x}{\overline{y}} = \frac{1 - \frac{e_y}{\overline{y}}}{1 - \left(\frac{e_y}{\overline{y}}\right)^2} \cdot \frac{\overline{x} + e_x}{\overline{y}} \cong \frac{\overline{x} + e_x}{\overline{y}} \cdot \left(1 - \frac{e_y}{\overline{y}}\right)$$

În şirul precedent de egalităţi s-a neglijat termenul: $\left(\frac{e_y}{\overline{y}}\right)^2 \approx 0$

$$\frac{x}{y} = \frac{\bar{x}}{\bar{y}} + \frac{e_x}{\bar{y}} - \frac{\bar{x}}{\bar{y}^2} \cdot e_y - \frac{e_x \cdot e_y}{\bar{y}^2} \qquad \text{neglijând termenul:} \qquad \frac{e_x \cdot e_y}{\bar{y}^2}$$

se obţine următoarea expresie pentru eroarea de trunchiere produsă la operaţia de împărţire:

$$e_{\frac{x}{y}} = \frac{e_x}{\bar{y}} - \frac{\bar{x}}{\bar{y}^2} \cdot e_y$$

Observaţie

La operaţia de împărţire eroarea produsă depinde nu numai de erorile termenilor care intervin în operaţia respectivă dar şi de valoarea acestora, **cu cât împărţitorul este mai mic cu atât valoarea erorii este mai mare**.

Ca urmare rezultă că în procesul de *inversare a unei matrici* este de preferat ca la fiecare etapă să se rearanjeze matricea, prin permutări de linii şi coloane, astfel încât termenul de pe diagonală să fie cel cu valoarea absolută maximă la etapa respectivă. Acest proces poartă denumirea de pivotare şi poate fi efectuat în două moduri:

→ *Pivotare parţială*, dacă la fiecare etapă se determină valoarea maximă de pe coloana corespunzătore etapei respective

→ *Pivotare totală*, dacă la fiecare etapă se determină valoarea maximă din întreaga matrice.

Există o categorie de **matrici** numite **slab condiţionate** care se caracterizează prin aceea că o mică modificare asupra valorii unui termen al matricii determină o modificare importantă asupra termenilor matricii inverse. Aceste matrici se caracterizează prin aceea că **valoarea determinantului matricii este mult mai mic decât termenii matricii**.

Exemplu:

$$A=\begin{bmatrix} 100 & 10 \\ 9{,}5 & 1 \end{bmatrix} \quad ; \quad |A|=5 \quad ; \quad A^{-1}=\begin{bmatrix} 0{,}2 & -2 \\ -1{,}9 & 20 \end{bmatrix}$$

presupunem că se produce o mică modificare a unuia dintre termenii matricii:

$$A=\begin{bmatrix} 100 & 10 \\ 9{,}9 & 1 \end{bmatrix} \quad ; \quad |A|=1 \quad ; \quad A^{-1}=\begin{bmatrix} 1 & -10 \\ -9{,}9 & 100 \end{bmatrix}$$

se poate constata o modificare semnificativă a termenilor matricei inverse.

Erorile de trunchiere pot determina obţinerea unor rezultate complet diferite faţă de valoarea reală a matricii inverse. Ca urmare pentru calculul matricii inverse se va utiliza un ***algoritm iterativ*** care permite efectuarea **corectării matricii inverse până când se obţine rezultatul cu precizia dorită**.

ALGORITMUL LUI HOTTELING PENTRU CALCULUL INVERSEI UNEI MATRICI

Algoritmul se bazează pe efectuarea unei operaţii de **împărţire prin operaţia de înmulţire**. Vom exemplifica algoritmul pentru cazul numerelor reale:

Presupunem că se cunoaşte o primă aproximare $(1/a)$ a inversului numărului real a pe care o notăm d_1. Condiţia de convergenţa a algoritmului este ca:

$$e_1 = \left|1 - a \cdot d_1\right| < 1$$

Algoritmul constă în determinarea unui şir de aproximări d_2, d_3, d_4, … a inversului numărului a astfel încât erorile să tindă spre zero.

Pentru a realiza condiţia cerută este necesar să se determine o relaţie de recurenţă astfel încât eroarea la un moment dat să poată fi exprimată sub forma unei puteri a lui e_1. Aceasta deoarece:

$$\lim_{k \to \infty} e_1^k = 0$$

Se pune condiţia: $e_2 = 1 - a \cdot d_2 = e_1^2$

din care rezultă succesiv:

$$e_1^2 = e_1 \cdot e_1 = (1 - a \cdot d_1) \cdot (1 - a \cdot d_1) = 1 - 2 \cdot a \cdot d_1 + a^2 \cdot d_1^2$$

adică:

$$e_1^2 = 1 - a \cdot d_1 \cdot [1 + (1 - a \cdot d_1)] = 1 - a \cdot d_1 \cdot (1 + e_1)$$

şi

$$d_2 = d_1 \cdot (1 + e_1)$$

Procesul de determinare al inversului unui număr va fi următorul:

$$d_2 = d_1 \cdot (1 + e_1) \quad \Rightarrow \quad e_2 = 1 - a \cdot d_2$$

$$d_3 = d_2 \cdot (1 + e_2) \quad \Rightarrow \quad e_3 = 1 - a \cdot d_3$$

……..

$$d_k = d_{k-1} \cdot (1 + e_{k-1}) \quad \Rightarrow \quad e_k = 1 - a \cdot d_k$$

care continuă până când e_k devine mai mic decât o valoare impusă.

Pentru aplicarea algoritmului în cazul matricilor se va utiliza noţiunea de ***normă*** a unei matrici, definită astfel:

$$N\left[A\right]=\underbrace{\max}_{i}\left(\sum_{j=1}^{n}\left|a_{i,j}\right|\right) \qquad \text{Dacă} \qquad N\left[A\right]<1 \qquad \text{atunci} \qquad \lim_{n\to\infty} N\left[A\right]^{n}=0$$

Pentru a calcula inversa unei matrici utilizând algoritmul lui Hotteling se va proceda după cum urmează:

Presupunem:

A - Matricea pentru care dorim să calculăm inversa

D_1 - Prima aproximare a inversei matricii, determinată cu algoritmul descris anterior

Eroarea care se produce în acest caz se va calcula astfel:

$$E_1 = U - A \cdot D_1$$

în care s-a notat cu U matricea unitate.

Următoarele aproximări ale matricii inverse se vor obţine astfel:

$$D_2 = D_1 \cdot (U + E_1) \Rightarrow \quad E_2 = U - A \cdot D_2$$

$$D_3 = D_2 \cdot (U + E_2) \Rightarrow \quad E_3 = U - A \cdot D_3$$

……

$$D_k = D_{k-1} \cdot (U + E_{k-1}) \Rightarrow \quad E_k = U - A \cdot D_k$$

Procesul se întrerupe în momentul în care este îndeplinită relaţia $N[E_k] < \varepsilon$

în care ε este o valoare impusă iniţial.

Concluzie

Calculul inversei unei matrici se va face parcurgând următoarele etape:

- determinarea unei prime aproximări a inversei prin algoritmul de inversare direct, de preferat utilizând pivotarea parţială sau totală;
- corectarea valorii inversei prin parcurgerea algoritmului lui Hotteling.

REZOLVAREA SISTEMELOR DE ECUAŢII LINIARE

Există două metode de rezolvare a sistemelor de ecuaţii liniare

Metode directe

Constau în obţinerea soluţiei printr-un **algoritm cu un număr finit de paşi** (de regulă egal cu numărul de ecuaţii ale sistemului) prin aducerea matricii de coeficienţi ai sistemului la o formă particulară (triunghiulară sau matrice unitate).

În cazul sistemelor slab condiţionate metoda se aplică repetat până la obţinerea soluţiei cu o precizie impusă iniţial.

Metode iterative (indirecte)

Constau în determinarea soluţiei într-un **număr de paşi care depinde de precizia cu care se efectuează calculul**.

METODE DIRECTE. *Metoda lui Gauss*

Metoda are drept scop rezolvarea sistemului: $A \cdot X = B$

Algoritmul constă în **aducerea matricii de coeficienţi *A* la forma de matrice unitate modificând simultan matricea *B*, a termenilor liberi**.

Se poate determina o secvenţa de matrici elementare cu ajutorul căreia, prin **preamplificare**, matricea *A* să fie adusă la forma de matrice unitate:

$$E_n \cdot E_{n-1} \cdots E_2 \cdot E_1 \cdot A \cdot X = E_n \cdot E_{n-1} \cdots E_2 \cdot E_1 \cdot B$$

rezultă că:

$$X = E_n \cdot E_{n-1} \cdots E_2 \cdot E_1 \cdot B$$

Algoritmul de rezolvare a sistemului va produce următoarea transformare:

$$\begin{bmatrix} a_{1,1} & \cdot & \cdot & \cdot & \cdot & a_{1,n} & b_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n,1} & \cdot & \cdot & \cdot & \cdot & a_{n,n} & b_n \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot & 0 & b_1^{(n)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 1 & b_n^{(n)} \end{bmatrix}$$

Ca urmare **soluţia sistemului** va fi: $x_i = b_i^{(n)} \quad , \; i = \overline{1,n}$

Se poate reduce numărul de operaţii dacă aducem matricea A la forma triunghiulară.

Acest algoritm nu este suficient pentru **sistemele slab condiţionate**. Este necesară aplicarea repetată a metodei directe. Succesiunea calculelor se va desfăşura în etape, după cum se prezintă în continuare:

Etapa 0

Se rezolvă, cu metoda Gauss, sistemul $A \cdot X = B$ obţinându-se soluţia: X_1

Etapa 1

Se corectează soluţia obţinută la etapa precedentă, astfel:

Se înlocuieşte soluţia în sistem: $A \cdot X_1 - B = E_1$

în care: E_1 este matricea de eroare.

Se rescrie sistemul sub forma: $A \cdot X_1 - E_1 = B$

Prin rezolvarea ecuaţiei (numită *ecuaţie de corecţie*): $A \cdot \delta X_1 = -E_1$

se obţine matricea δX_1 care conţine corecţiile care trebuiesc aplicate soluţiilor. Ca urmare ecuaţia iniţială se va putea scrie sub forma:

$$A \cdot X_1 + A \cdot \delta X_1 = B \quad \Rightarrow \quad A \cdot \left(X_1 + \delta X_1\right) = B$$

După ***Etapa 1***, soluţiile vor fi formate din soluţiile obţinute la ***Etapa 0*** la care se adaugă corecţiile obţinute la ***Etapa 1***, prin rezolvarea ***ecuaţiei de corecţie***, adică:

$$X = X_1 + \delta X_1$$

Dacă **erorile sunt mai mici** decât o valoare impusă iniţial **ε**, atunci procedeul de corectare se încheie, în caz contrar se efectuează o noua etapă de corecţie.

Procedeul se reia până când erorile obţinute devin mai mici decât valoarea **ε** impusă iniţial. În final soluţia sistemului se va obţine sub forma:

$$X = X_1 + \delta X_1 + \cdots + \delta X_k$$

în care ***k*** reprezintă numărul de etape de corectare efectuate.

METODE ITERATIVE

Aceste metode sunt **convergente** numai pentru sistemele de ecuaţii a căror matrice de coeficienţi este ***diagonal dominantă***.

Prin matrice diagonal dominantă se înţelege matricea la care **termenii de pe diagonală au valorile absolute mai mari sau cel mult egale cu suma valorilor absolute a termenilor aflaţi pe aceeaşi linie cu ei**. Adică este îndeplinită condiţia:

$$\left| a_{i,i} \right| \geq \sum_{\substack{j=1 \\ j \neq i}}^{j=n} \left| a_{i,j} \right| \quad , \quad i = \overline{1,n}$$

În cazul tuturor metodelor iterative algoritmul porneşte de la o **soluţie a sistemului aleasă în mod arbitrar**, de obicei soluţia nulă:

$$x_i^{(0)} = 0 \quad , \quad i = \overline{1,n}$$

Scopul metodelor este acela de a **corecta succesiv soluţia iniţială până la obţinerea soluţiei reale** a sistemului.

Metoda Jacobi

Aplicarea metodei presupune scrierea sistemului sub o formă în care **din fiecare ecuaţie se explicitează succesiv câte o necunoscută**. Ca urmare, dacă sistemul iniţial este:

$$\sum_{j=1}^{n} a_{i,j} \cdot x_j = b_i \quad , \quad i = \overline{1,n}$$

acesta se rescrie sub forma: $$x_i = \frac{1}{a_{i,i}} \cdot \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{j=n} a_{i,j} \cdot x_j \right) \quad , \quad i = \overline{1,n}$$

O iteraţie, ***k***, a metodei Jacobi constă în **determinarea soluţiei sistemului, corespunzătoare acelei iteraţii, prin înlocuirea în sistem a soluţiilor determinate la iteraţia precedentă**.
La prima iteraţie se utilizează soluţia de start. Ca urmare pentru o iteraţie oarecare ***k***, se poate scrie:

$$x_i^{(k)} = \frac{1}{a_{i,i}} \cdot \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{j=n} a_{i,j} \cdot x_j^{(k-1)} \right), \quad i = \overline{1,n}$$

Algoritmul se încheie atunci când **diferenţa dintre două soluţii determinate succesiv este mai mică decât o valoare impusă iniţial**, adică este îndeplinită condiţia:

$$\left| x_i^{(k)} - x_i^{(k-1)} \right| \leq \varepsilon , \quad i = \overline{1,n}$$

Metoda Gauss-Seidel

Metoda reprezintă o îmbunătăţire a metodei Jacobi în sensul că la iteraţia ***k***, valorile necunoscutelor sunt calculate nu numai funcţie de valorile determinate la **iteraţia precedentă** dar şi de cele calculate la **iteraţia în curs**.

Ca urmare, relaţia de calcul a valorilor necunoscutelor devine:

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

Metoda Southwell

Aplicarea metodei presupune scrierea sistemului de ecuaţii $A \cdot X = B$ sub o formă în care **din fiecare ecuaţie a fost separată o necunoscută**. Ca urmare forma iniţială a sistemului:

$$\sum_{j=1}^{n} a_{i,j} \cdot x_j = b_i \ , \qquad i = \overline{1,n}$$

va suferi următoarele transformări:

- se împarte fiecare ecuaţie la coeficientul de pe **diagonala principală:**

$$x_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{a_{i,j}}{a_{i,i}} \cdot x_j - \frac{b_j}{a_{i,i}} = 0$$

- după care, sistemul se va scrie sub forma: $$-x_i + \sum_{\substack{j=1 \\ j \neq 1}}^{n} b_{i,j} \cdot x_j + c_i = 0$$

în care s-a notat:

$$b_{i,j} = -\frac{a_{i,j}}{a_{i,i}} \qquad c_i = \frac{b_i}{a_{i,i}}$$

Algoritmul constă în parcurgerea mai multor etape plecând de la o **soluţie iniţială aleasă în mod arbitrar**. De regulă se aleg ca **valori iniţiale valorile nule**: $x_i^{(0)} = 0 \quad , \quad i = \overline{1,n}$

***Etapa 1*:**

Se înlocuiesc valorile iniţiale în sistem, ca urmare din fiecare ecuaţie va rezulta o valoare numită ***rest***. Valorile acestora se vor obţine cu relaţiile:

$$r_i^{(1)} = -x_i^{(0)} + \sum_{\substack{j=1 \\ j \neq i}}^{n} b_{i,j} \cdot x_j^{(0)} + c_i$$

Se determină ***valoarea absolută maximă dintre resturile*** calculate anterior:

$$\left| r_m^{(1)} \right| = \max_i \left| r_i^{(1)} \right| \quad , \quad i = \overline{1,n}$$

valoarea astfel determinată ***va corecta valoarea necunoscutei având indicele corespunzător***

$$x_m^{(1)} = x_m^{(0)} + r_m^{(1)}$$

După ce a fost obţinută valoarea $x_m^{(1)}$ ***se recalculează resturile utilizând această valoare a necunoscutei***, se obţin astfel **valori noi pentru celelalte *n*-1 resturi**. ***Din nou se alege valoarea maximă dintre cele n-1 resturi corectându-se necunoscuta corespunzătoare***. Procedeul se repetă, în acelaşi mod, **până la corectarea tuturor necunoscutelor**, după care se trece la etapa următoare.

***Etapa k*:** Se înlocuiesc valorile necunoscutelor obţinute la etapa precedentă în sistem, se obţin **resturile**:

$$r_i^{(k)} = -x_i^{(k-1)} + \sum_{\substack{j=1 \\ j \neq i}}^{n} b_{i,j} \cdot x_j^{(k-1)} + c_i$$

Se determină **restul cu valoarea absolută cea mai mare**:

$$\left| r_m^{(k)} \right| = \max_i \left| r_i^{(k)} \right| \quad , \quad i = \overline{1,n}$$

valoarea astfel determinată va **corecta valoarea necunoscutei având indicele corespunzător**

$$x_m^{(k)} = x_m^{(0)} + r_m^{(1)} + r_m^{(2)} + \cdots + r_m^{(k)}$$

După ce a fost obţinută valoarea $x_m^{(k)}$ **se recalculează resturile utilizând această valoare a necunoscutei, se obţin astfel valori noi pentru celelalte *n*-1 resturi şi se alege valoarea maximă corectându-se necunoscuta corespunzătoare**. Procedeul se repetă, în acelaşi mod, până la corectarea tuturor necunoscutelor. Algoritmul se încheie atunci când diferenţa dintre două soluţii determinate succesiv este mai mică decât o valoare impusă iniţial, adică este îndeplinită condiţia:

$$\left| x_i^{(k)} - x_i^{(k-1)} \right| \leq \varepsilon \quad , \quad i = \overline{1,n}$$

Exemplu

Se consideră sistemul: $\begin{cases} 10 \cdot x_1 - 2 \cdot x_2 = 8 \\ -x_1 + 10 \cdot x_2 = 9 \end{cases}$

Sistemul are, evident, soluţiile: $x_1 = 1$, $x_2 = 1$.

Rezolvarea sistemului prin metoda Southwell presupune scrierea acestuia, succesiv, în următoarele forme:

$$\begin{cases} x_1 - 0{,}2 \cdot x_2 = 0{,}8 \\ -0{,}1 \cdot x_1 + x_2 = 0{,}9 \end{cases}$$

Respectiv:

$$\begin{cases} -x_1 + 0{,}2 \cdot x_2 + 0{,}8 = 0 \\ -x_2 + 0{,}1 \cdot x_1 + 0{,}9 = 0 \end{cases}$$

Se aleg ca valori iniţiale ale soluţiilor valorile: $x_1 = 0$, $x_2 = 0$.

$$\begin{cases} -x_1 + 0{,}2 \cdot x_2 + 0{,}8 = 0 \\ -x_2 + 0{,}1 \cdot x_1 + 0{,}9 = 0 \end{cases}$$

Etapa 1.

Se înlocuiesc **valorile iniţiale** în sistem, $\begin{cases} r_1^{(1)} = 0{,}8 \\ r_2^{(1)} = 0{,}9 \end{cases}$

se obţin următoarele valori pentru resturi:

Se determină **valoarea absolută maximă** dintre *resturile* calculate anterior, care va fi:

$$r_2^{(1)} = 0{,}9$$

Valoarea astfel determinată va corecta **valoarea necunoscutei având indicele corespunzător**:

$$x_2^{(1)} = x_2^{(0)} + r_2^{(1)} = 0 + 0{,}9 = 0{,}9$$

Se **recalculează** restul $r_1^{(1)}$ utilizând această valoare a necunoscutei x_2, şi se obţine:

$$r_1^{(1)} = 0 + 0{,}2 \cdot 0{,}9 + 0{,}8 = 0{,}98$$

Deoarece este singurul rest rămas, nu mai este necesară determinarea unei **valori maxime**. Ca urmare, va fi corectată valoarea necunoscutei corespunzătoare, adică x_1, se obţine:

$$x_1^{(1)} = 0 + 0{,}98$$

Ca urmare la **încheierea *Etapei 1*** se obţin $\begin{cases} x_1^{(1)} = 0{,}98 \\ x_2^{(1)} = 0{,}9 \end{cases}$

următoarele valori ale soluţiilor sistemului:

Se constată că valorile obţinute s-au **apropiat semnificativ de soluţiile sistemului**. Se pot parcurge etape suplimentare până la obţinerea soluţiilor cu precizia dorită.