Metode Numerice – Lucrarea nr. 4

METODE NUMERICE DE REZOLVARE A SISTEMELOR DE ECUAȚII

Modelul matematic şi metodele numerice utilizate

Determinarea ecuației matriceale a circuitului și a sistemului de ecuații a curenților

Un sistem de **n** ecuaţii liniare, având coeficienţii reali a_{ij} *i, j*= 1,2,…,n, cu ***n*** neconoscute şi termenii liberi reali b_i , i = 1, 2,…,n, cu ***n*** necunoscute, x_i, este de forma relației:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n = b_3 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n \end{cases} \tag{1}$$

Iar în forma matriceală se poate scrie: $A{\cdot}x = b$, conform relației (2):

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}; \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} \tag{2}$$

unde: *A* este matricea pătrată *n* x *n* dimensională a sistemului de ecuaţii, *b* este vectorul termenilor liberi şi *x* vectorul necunoscutelor.

În unele cazuri, sistemele de ecuaţii algebrice liniare apar în mod natural, din însăşi formularea problemei. În multe alte cazuri, însă, sistemele de ecuaţii liniare rezultă ca urmare a aplicării unor metode numerice de rezolvare a problemei iniţiale. Se poate spune că rezolvarea sistemelor de ecuaţii liniare joacă un rol central în cadrul metodelor numerice.

Rezolvarea ecuației matriceale cu metoda inversării matriceale

Fie ecuația matriceală $A \cdot X = B$. Dacă se înmulțește la stânga cu A^{-1} ecuația devine: $A^{-1} \cdot (A \cdot X) = A^{-1} \cdot (B)$.

Se știe că $M^{-1} \cdot M = I, \ \forall \mathrm{M} \in \mathrm{R}^* \times \mathrm{R}^*$, deci ecuația devine $I \cdot X = A^{-1} \cdot B$.

Se știe că $I \cdot M = M, \ \forall \mathrm{M} \in \mathrm{R}^* \times \mathrm{R}^*$, deci ecuația devine $X = A^{-1} \cdot B$.

Rezolvarea ecuațiilor matriceale și sistemelor de ecuații utilizând funcțiile built-in a software-ului utilizat

Foarte multe softuri de calcul numeric oferă posibilitatea utilizării funcțiilor built-in pentru a executa operații cu matrici, a rezolva ecuații matriceale sau sisteme de ecuații. Dintre funcțiile de acest gen ale sofware-ului *Mathcad* se pot aminti: funcţia *lsolve* şi blocul *Given – Find*.

Metoda aproximațiilor succesive a lui Jacobi (metodă iterativă)

Considerăm un sistem de ***n*** ecuaţii liniare cu ***n*** necunoscute scris matricial **[A]·[x]=[B]**, dacă $\det(A)\neq 0$ şi dacă $a_{ij}\neq 0$ atunci sistemul **[A]·[x]=[B]** îl rescriem sub forma:

$$\begin{cases} a_{11}x_1+a_{12}x_2+a_{13}x_3+\ldots+a_{1n}x_n=b_1 \\ a_{21}x_1+a_{22}x_2+a_{23}x_3+\ldots+a_{2n}x_n=b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1+a_{n2}x_2+a_{n3}x_3+\ldots+a_{nn}x_n=b_n \end{cases} \Rightarrow \begin{cases} x_1=\alpha_{11}x_1+\alpha_{12}x_2+\ldots+\alpha_{1n}x_n+\beta_1 \\ x_2=\alpha_{21}x_1+\alpha_{22}x_2+\ldots+\alpha_{2n}x_n+\beta_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n=\alpha_{n1}x_1+\alpha_{n2}x_2+\ldots+\alpha_{nn}x_n+\beta_n \end{cases}$$

adică matricial: **[x]=[α]·[x]+[β]** unde:

$$\alpha=-\begin{pmatrix} 0 & \frac{a_{12}}{a_{11}} & \ldots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 0 & \ldots & \frac{a_{2n}}{a_{22}} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{a_{n1}}{a_{nn}} & \frac{a_{n2}}{a_{nn}} & \ldots & 0 \end{pmatrix} \qquad \beta=\begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \ldots \\ \frac{b_n}{a_{nn}} \end{pmatrix} \tag{3}$$

Se continuă rezolvarea sistemului în forma $x=\alpha\cdot x+\beta$.

Se alege vectorul aproximaţiilor iniţiale ale soluţiilor $x^{(0)}=\left[x_1^{(0)}\quad x_2^{(0)}\quad \ldots\quad x_n^{(0)}\right]^T$ care se obţine prin măsurători experimentale sau se alege de obicei în aplicaţiile practice ca fiind egal cu vectorul termenilor liberi $x^{(0)}=\beta$.

Metoda lui Jacobi presupune calculul unui şir de aproximaţii succesive $x^{(1)},x^{(2)},\ldots$ cu ajutorul formulei de iterare într-un pas care se demonstrează prin inducţie:

$$x^{(k+1)}=\alpha\cdot x^{(k)}+\beta \tag{4}$$

Astfel vectorul soluţie după prima iteraţie devine:

$$x^{(1)}=\alpha\cdot x^{(0)}+\beta \tag{5}$$

Dacă şirul de soluţii aproximative de la iteraţia *k*, $x^{(k)}$, converge, atunci limita lui $\lim\limits_{k\to\infty}x^{(k)}=\lim\limits_{k\to\infty}\left(x_1^{(k)},x_2^{(k)},\ldots,x_n^{(k)}\right)=x$ este soluţie a sistemului **[A]·[x]=[B]**.

Evaluarea erorii în metoda Jacobi se face în funcţie de normele lui **[α]** şi **[β]**, conform relației:

$$er=\left\|x-x^{(k)}\right\|\leq\frac{\|\alpha\|^{k+1}}{1-\|\alpha\|}\cdot\|\beta\| \tag{6}$$

Comentarii asupra metodei lui Jacobi

- Metoda se poate aplica şi pentru sisteme neliniare;
- Soluţia se obţine printr-o serie (proces) de aproximaţii succesive, fiecare secvenţă de operaţii aritmetice elementare (mai mic decât la metodele directe) este parcursă de mai multe ori (metodă iterativă);
- Pentru calculul lui $x_i^{(k+1)}$ avem nevoie de toate valorile $x_1^{(k)},...,x_n^{(k)}$ ceea ce duce la creşterea timpului de calcul;
- Dacă şirul iteraţiilor este convergent, cu cât se efectuează mai multe iteraţii, cu atât soluţia numerică este mai precis determinată iar erorile atât cele de trunchiere cât şi cele de rotunjire, devin tot mai mici;
- Se obţin aproximaţii din ce în ce mai bune ale soluţiei (prin parcurgerea procesului iterativ) până la atingerea unei precizii fixate dinainte (precizie dorită);
- condiţie suficientă pentru convergenţa procesului este: $\max_{1\le j\le n}\sum_{j=1}^{n}\left|\alpha_{ij}\right|<1$, iar datorită convergenţei lor metodele iterative au proprietatea de a corecta erorile de rotunjire;
- Aceste metode permit obţinerea soluţiei numerice a unui sistem de ecuaţii prin generarea unui şir care tinde la soluţia exactă;
- Deoarece practic se poate efectua numai un număr finit de iteraţii, în mod inevitabil erorile de rotunjire sunt însoţite în cazul metodelor iterative şi de erori de trunchiere;
- Dintre principalele avantaje ale metodelor iterative menţionăm simplitatea şi eficienţa implementării lor în programe de calcul în cazurile în care nu sunt rezolvabile prin metode directe;
- Chiar dacă soluţia obţinută prin metode iterative este afectată de erori de trunchiere (prin reţinerea din şirul convergent către soluţia exactă a unui număr finit de termeni), erori care nu apar la metodele directe, este totuşi posibil ca soluţia să fie mai precisă decât cea obţinută prin metode directe;
- În practică, prin efectuarea unui număr finit de iteraţii, se poate ajunge la o aproximare suficient de bună a soluţiei exacte;
- Eroarea finală depinde de eroarea iniţială, de numărul de iteraţii efectuate şi de norma matricei de iteraţie, care determină viteza de convergenţă.

Instrumente folosite

Operații cu matrici

În continuare, sunt prezentate principalele caracterisitici ale matricilor și operațiile care se pot executa, folosind utilitarul *Mathcad*.

Pasul 1. Pentru a introduce o matrice în *Mathcad*, este necesar toolbar-ul *Matrix*. Dacă toolbar-ul *Matrix* nu este vizibil, din meniul principal se selectează *View – Toolbars – Matrix*, prezentat în Fig. 1. Din toolbar-ul *Matrix*, se selectează icoana *Matrix or Vector*, conform Fig. 2 (shortcut: **Ctrl+M**).

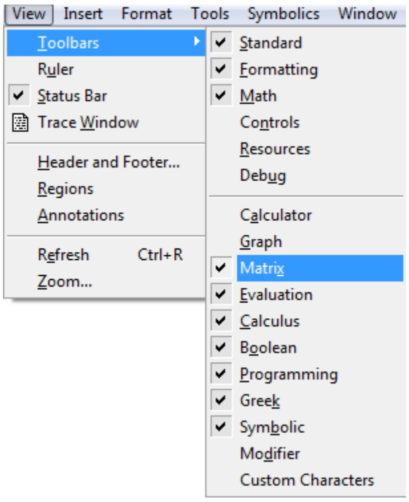


Fig. 1: *Activarea paletei „Matrix”*

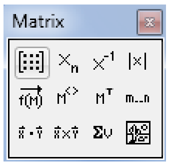


Fig. 2: *Inserarea unei Matrice*

Pasul 2. Apare fereastra *Insert Matrix*, conform Fig. 3. Se poate specifica numărul de coloane a matricii în rubrica *Columns* și numărul de linii în rubrica *Rows*. Dacă se dorește inserarea unui vector, acesta este definit ca o matrice cu o singură coloană.

Pasul 3. După selectarea numărului de linii (*Rows*) și coloane (*Columns*), apare o matrice în fereastra de comandă, unde se pot introduce valorile dorite pe fiecare poziție, conform Fig 4.

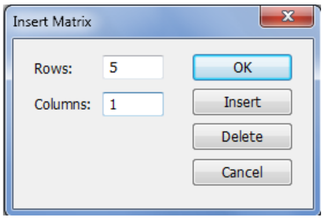


Fig. 3: *Definirea dimensiunilor matricei*

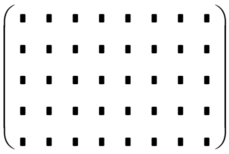


Fig. 4: *Matricea goala introdusa*

Pasul 4. Referirea la elementele matricii se face în felul următor: $M_{a,b}$ unde M este matricea, a este linia, iar b este coloana. Numerotarea coloanelor și liniilor începe de la 0. Indicii a, b se introduc cu tasta „**[**” .

Pasul 5. Matricilor li se pot aplica comenzi, care oferă direct determinantul, inversa, transpusa matricii,etc. Comanda se poate aplica unei variabile de tip matrice, sau direct unei matrici. Când comanda se aplică direct unei matrici, cursorul de selecție trebuie să fie lângă paranteza stângă sau dreaptă a matricii.

$$\boxed{\begin{pmatrix}1 & 0 & 1\\ 1 & 1 & 1\\ 1 & 0 & 0\end{pmatrix}}$$

Pasul 6. Selectând *Inverse* din toolbar-ul *Matrix* se poate determina inversa unei matrici, conform Fig. 5.

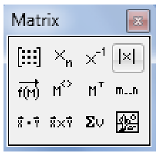
$$M := \begin{pmatrix}1 & 0 & 2\\ 4 & 3 & 6\\ 7 & 8 & 9\end{pmatrix} \qquad M^{-1} = \begin{pmatrix}-21 & 16 & -6\\ 6 & -5 & 2\\ 11 & -8 & 3\end{pmatrix}$$

sau

$$\begin{pmatrix}1 & 0 & 2\\ 4 & 3 & 6\\ 7 & 8 & 9\end{pmatrix}^{-1} = \begin{pmatrix}-21 & 16 & -6\\ 6 & -5 & 2\\ 11 & -8 & 3\end{pmatrix}$$

Fig. 5: *Apelarea operatorului de inversare*

Pasul 7. Selectând *Determinant* din toolbar-ul *Matrix* se poate calcula determinantul unei matrici, conform Fig. 6.



$$M := \begin{pmatrix} 1 & 0 & 2 \\ 4 & 3 & 6 \\ 7 & 8 & 9 \end{pmatrix} \qquad |M| = 1$$

sau $$\left| \begin{pmatrix} 1 & 0 & 2 \\ 4 & 3 & 6 \\ 7 & 8 & 9 \end{pmatrix} \right| = 1$$

Fig. 6: *Calcularea determinantului*

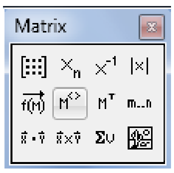
Pasul 8. Selectând *Matrix Transpose* din toolbar-ul *Matrix* se poate reprezenta transpusa unei matrici, conform Fig. 7.

$$M := \begin{pmatrix} 1 & 0 & 2 \\ 4 & 3 & 6 \\ 7 & 8 & 9 \end{pmatrix} \qquad M^T = \begin{pmatrix} 1 & 4 & 7 \\ 0 & 3 & 8 \\ 2 & 6 & 9 \end{pmatrix}$$

sau $$\begin{pmatrix} 1 & 0 & 2 \\ 4 & 3 & 6 \\ 7 & 8 & 9 \end{pmatrix}^T = \begin{pmatrix} 1 & 4 & 7 \\ 0 & 3 & 8 \\ 2 & 6 & 9 \end{pmatrix}$$

Fig. 7: *Apelarea operatorului de Transpunere*

Pasul 9. Selectând *Matrix Column* din toolbar-ul *Matrix* se poate afișa coloana unei matrici, conform Fig. 8.



$$M := \begin{pmatrix} 1 & 0 & 2 \\ 4 & 3 & 6 \\ 7 & 8 & 9 \end{pmatrix} \qquad M^{\langle 2 \rangle} = \begin{pmatrix} 2 \\ 6 \\ 9 \end{pmatrix}$$

sau $$\begin{pmatrix} 1 & 0 & 2 \\ 4 & 3 & 6 \\ 7 & 8 & 9 \end{pmatrix}^{\langle 0 \rangle} = \begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix}$$

Fig. 8: *Extragerea unei coloane dintr-o matrice*

Funcția lsolve

Funcția *lsolve* este o funcție built-in în Mathcad, utilizată pentru rezolvarea sistemelor de ecuații liniare (**l**inear **solve**). Se consideră sistemul de ecuații:

$$\begin{aligned} a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 &= b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 &= b_2 \\ a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 &= b_3 \end{aligned} \qquad (2)$$

Indicii formali se introduc cu tasta „**.**”

Pasul 1. Pentru apelarea funcției *lsolve*, sunt necesare două argumente: *lsolve(M,v)*, unde *M* este matricea coefcienților și v este vectorul termenilor liberi. Funcția *lsolve* returnează ca rezultat un vector, care conține valorile necunoscutelelor (x_1, x_2, x_3). În cazul de față:

$$M := \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \qquad v := \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

$$\text{rezultat} := \text{lsolve}(M, v) \qquad \text{sau} \qquad \text{rezultat} := \text{lsolve}\left[\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}\right]$$

Pasul 2. Există cazuri când funcția *lsolve* nu ne oferă rezultatele cu precizia impusă apriori. În acest caz se poate utiliza varianta optimizată a funcției: făcând click dreapta pe numele funcției, apare un meniu, din care se selectează *Optimize*, conform Fig. 10. Apare o *steluță* (*) în dreapta parantezei, notificând utilizatorul că se folosește varianta optimizată a funcției *lsolve*.

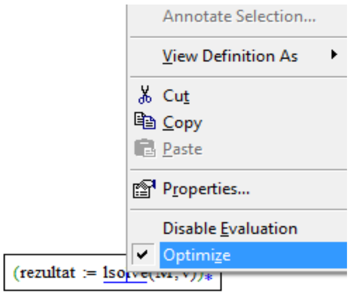


Fig. 9: *Folosirea variantei optimizate a funcției* lsolve

$$\text{rezultat} := \text{lsolve}(M, v)_{*} \qquad \text{sau} \qquad \text{rezultat} := \text{lsolve}\left[\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}\right]_{*}$$

Blocul Given – Find

Blocul *Given – Find* se poate utiliza la calculul direct al valorii uneia sau mai multor variabile. Este necesară introducerea ecuațiilor și egalului boolean (**Ctrl** =) pentru aplicarea blocului de rezolvare.

Pasul 1. Se introduce comanda *Given* înaintea ecuației sau a ecuațiilor sistemului.

Pasul 2. Se introduce ecuația sau ecuațiile sistemului

$$\begin{array}{lcl} \text{Given} & & \text{Given} \\ 3x + 4 = 5 & \text{sau} & 3x_1 + 8x_2 = 2 \\ & & x_1 \cdot x_2 = 45 \end{array}$$

Pasul 3. Se introduce comanda *Find* pentru determinarea necunoscutelor, urmată de un operator de evaluare simbolică.

Given

$$3x + 4 = 5$$

$$\mathrm{Find}(x) \rightarrow \frac{1}{3}$$

sau

Given

$$3x_1 + 8x_2 = 19$$

$$x_1 \cdot x_2 = 2$$

$$\mathrm{Find}(x_1, x_2) \rightarrow \begin{pmatrix} \frac{16}{3} & 1 \\ \frac{3}{8} & 2 \end{pmatrix}$$

Operatorul de evaluare simbolică se introduce din toolbar-ul *Symbolic*. Dacă acest toolbar nu este vizibil, din meniul principal se selectează *View – Toolbars – Symbolic*, conform Fig. 10. Din toolbar-ul *Symbolic* se selectează *Symbolic Evaluation*, Fig. 11.

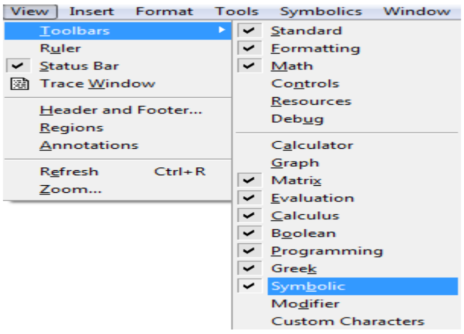


Fig. 10: *Activarea paletei „Symbolic”*

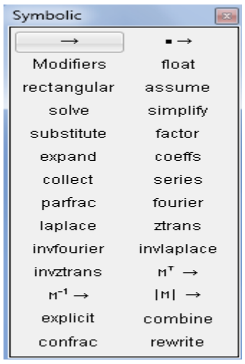


Fig. 11: *Apelarea operatorului de evaluare simbolică*