

CALCUL NUMERIC ÎN MATLAB

1.1. Operații aritmetice

În MATLAB, calculele aritmetice asupra tablourilor de date pot fi: operații după regulile calculului matriceal (operații cu matrice) și operații după regulile calculului scalar (operații cu tablouri). În tabelul 1.1 sunt prezentați operatorii aritmetici din MATLAB.

Tab.1.1. Operatorii aritmetici MATLAB

Operația	Scalar	Matrice	Tablouri
Adunarea	+	+	.+
Scăderea	-	-	.-
Înmulțirea	*	*	.*
Împărțirea la stânga	\	\	.\
Împărțirea la dreapta	/	/	./
Ridicarea la putere	^	^	.^
Transpunerea	'	'	.'

Ordinea operațiilor aritmetice este aceeași cu cea cunoscută în matematica elementară, a operațiilor aritmetice standard (tabelul 1.2).

Tab.1.2. Ordinea operațiilor aritmetice

Ordinea	Operația
---------	----------

1	parantezele
2	ridicarea la putere
3	înmulțirea și împărțirea
4	adunarea și scăderea

Deși variabilele memorate au un interval foarte mare, cel mai adesea fiind în limitele 10^{-308} și 10^{308} (vezi funcțiile MATLAB **realmax** și **realmin**), totuși uneori este posibil ca rezultatul unei expresii să depășească aceste limite. Dacă rezultatul unui calcul este mai mare de 10^{308} , MATLAB-ul înregistrează ∞ (Inf). Dacă rezultatul unui calcul este mai mic decât 10^{-308} , calculatorul înregistrează valoarea zero. În MATLAB, rezultatul împărțirii cu zero este ∞ . În acest caz este afișat mesajul de atenționare „Warning: Divide by zero”, dar calculele sunt continuate cu operandul ∞ .

```
>> x=2.5*10^200;
```

```
>> y=10^200;
```

```
>> z=x*y
```

```
z =
```

```
Inf
```

```
>> x=2.5*10^200;
```

```
>> y=0;
```

```
>> z=x/y
```

```
Warning: Divide by  
zero.
```

```
z =
```

```
Inf
```

1.1.1. Operații aritmetice cu scalari

În tabelul 1.3 sunt prezentate operațiile aritmetice între doi scalari, fiind prezentată atât forma algebrică cât și forma MATLAB.

Tab.1.3. Forma MATLAB a operatorilor scalari

Operația	Forma algebrică	Forma MATLAB
Adunare	$a+b$	$a+b$
Scădere	$a-b$	$a-b$
Înmulțire	$a \times b$	$a*b$
Împărțire la dreapta	$a:b$	a/b

Împărțire la stânga	$b:a$	$a \setminus b$
Ridicare la putere	a^b	a^b

Expresiile aritmetice pot fi evaluate și rezultatul memorat în variabila specificată. O valoare introdusă fără nominalizare este asignată variabilei **ans** (**answer**). În variabila ans este memorată în permanență valoarea ultimei variabile căreia nu i s-a atribuit un nume.

1.1.2. Operații aritmetice cu tablouri

Operațiile cu tablouri sunt operații aritmetice între elementele situate în aceeași poziție a tablourilor, cunoscute sub numele de operații element cu element. Pentru a preciza că o operație se efectuează element cu element între componentele a două matrice de aceeași dimensiune, se utilizează operatorul corespunzător operației precedat de punct (vezi tab.1.1). Dacă unul dintre operanzi este un scalar, acesta operează cu fiecare element al tabloului.

1.1.3. Operații aritmetice cu matrice

Operațiile uzuale de algebră liniară cu matrice sunt simbolizate cu semnele grafice prezentate în tab.1.1 și se efectuează după regulile cunoscute din calculul matriceal.

Operația de adunare a două matrice este simbolizată cu operatorul plus. Instrucțiunea:

$$Z = X + Y$$

reprezintă adunarea matricelor X și Y, rezultând elementele:

$$Z(i, j) = X(i, j) + Y(i, j)$$

Matricele X și Y trebuie să aibă aceeași dimensiune, în afara cazului când una dintre matrici, X sau Y, este un scalar. Un scalar poate fi adunat cu orice matrice.

Operația de scădere două matrice este simbolizată cu operatorul minus. Instrucțiunea:

$$Z = X - Y$$

reprezintă scăderea matricelor X și Y, rezultând elementele:

$$Z(i, j) = X(i, j) - Y(i, j)$$

Matricele X și Y trebuie să aibă aceeași dimensiune, în afara cazului când una dintre ele este un scalar.

Operația de înmulțire a două matrice este simbolizată cu operatorul steluță. Instrucțiunea:

$$Z = X * Y$$

reprezintă matricea produs având elementele:

$$Z(i, j) = \sum_k X(i, k)Y(k, j)$$

Produsul matriceal este posibil numai dacă numărul de coloane al matricei X este egal cu numărul de linii al matricei Y. Produsul matrice-vector este un caz particular al cazului general al produsului matrice-matrice. De asemenea, un scalar poate fi înmulțit cu orice matrice.

Pentru operația de împărțire a matricelor, în MATLAB avem două cazuri: împărțirea la dreapta și împărțirea la stânga. Operația de împărțire la dreapta a două matrice este simbolizată cu operatorul slash. Instrucțiunea:

$$Z = X/Y$$

reprezintă împărțirea la dreapta a matricelor X și Y, și este identică cu:

$$Z = X * Y^{-1}$$

Operația de împărțire la stânga a două matrice este simbolizată cu operatorul backslash. Instrucțiunea:

$$Z = X \backslash Y$$

reprezintă împărțirea la stânga a matricelor X și Y și este identică cu:

$$Z = X^{-1} * Y$$

Dacă unul dintre operanzi este un scalar, operația nu este posibilă.

Operația de ridicare la putere a unei matrice este simbolizată cu operatorul ^. Următoarea instrucțiune:

$$Z = X^p$$

Reprezintă ridicarea la puterea p a matricei X. Expresia are sens doar pentru X matrice pătrată și pentru p scalar. Dacă p este un întreg pozitiv, ridicarea la putere este obținută prin înmulțiri repetate, iar dacă p este un întreg negativ, matricea X este mai întâi inversată și apoi se înmulțesc inversele de p ori.

Operația de transpunere a unei matrice este simbolizată cu operatorul apostrof. Cu instrucțiunea:

$$Z = Y'$$

liniile matricei Y devin coloanele matricei transpuse Z. dacă elementele matricei Y sunt numere complexe, operația de transpunere returnează conjugata transpusei, adică:

$$Z(i, j) = \text{conj}(Y(i, j)) = \text{real}(Y(i, j)) - i * \text{imag}(Y(i, j))$$

1.2. Generarea matricilor

Deși în MATLAB nu există instrucțiuni pentru declararea tipurilor de variabile, matricele se autodimensionează în timpul utilizării, pentru a crește viteza de lucru se procedează la crearea unei matrice goale. Acest lucru se face

la începutul sesiunii de lucru sau la apelarea unui program. Declararea unei matrice goale se face cu instrucțiunea:

```
X = [ ]
```

care asignează lui X matricea de dimensiuni 0x0. Orice matrice goală trebuie să aibă cel puțin una din dimensiuni zero. Pentru a afla dacă o matrice este goală se folosește funcția MATLAB **isempty** care se apelează cu sintaxa:

```
r = isempty(X)
```

Funcția întoarce rezultatul $r = 1$ dacă matricea este goală și $r = 0$ în caz contrar.

Matricea unitate este o matrice care are toate elementele egale cu 1 și poate fi generată cu funcția **ones** ce se apelează cu sintaxa:

```
U = ones(m,n)
```

unde m și n sunt scalari ce definesc dimensiunea matricei.

Pentru a genera matricea zero (este o matrice cu toate elementele zero) se folosește funcția **zeros** ce se apelează cu sintaxa:

```
O = zeros(m,n)
```

Matricea identitate (are elementele de pe diagonala principală 1) se generează cu funcția **eye** care este apelată cu sintaxa:

```
I = eye(m,n)
```

1.3. Generarea vectorilor cu pas liniar și cu pas logaritmic

În MATLAB se pot genera vectori cu pas liniar sau vectori cu pas logaritmic. Generarea vectorilor cu pas liniar implică cunoașterea limitelor intervalului (a_{\min} ; a_{\max}) și a pasului dintre două elemente sau a numărului de elemente ale vectorului (N).

Dacă se cunosc limitele intervalului și pasul dintre două elemente, vectorul se generează cu instrucțiunea:

```
x = amin:pas:amax
```

numărul de elemente al vectorului x este: $N = \left\lceil \frac{a_{\max} - a_{\min}}{pas} \right\rceil + 1$.

unde $\lceil \cdot \rceil$ semnifică partea întreagă a numărului. Dacă pasul este mai mare ca zero, este necesar ca $a_{\min} < a_{\max}$, iar dacă pasul este mai mic decât zero este necesar ca $a_{\min} > a_{\max}$.

Dacă se cunosc limitele intervalului și numărul de elemente ale vectorului, vectorul se generează cu ajutorul funcției **linspace**, care este apelată cu următoarea sintaxă:

```
x = linspace(amin,amax,N)
```

rezultând un vector cu N elemente (dacă valoarea N nu este precizată atunci aceasta este considerată implicit egală cu 100), pasul dintre două elemente fiind:

$$pas = \frac{a_{\max} - a_{\min}}{N - 1}.$$

Pentru generarea vectorilor cu pas logaritmic se utilizează funcția **logspace** ce se apelează cu sintaxa:

```
x = logspace(amin,amax,N)
```

Vectorul x conține N elemente distribuite logaritmic între decadele $[10^{a_{\min}}, 10^{a_{\max}}]$. Dacă N nu este precizat el este luat implicit 50.

1.4. Calcule cu matrice

Formularea matricială a problemelor a condus la simplificarea metodelor de rezolvare și a făcut posibilă extinderea unor soluții deja cunoscute la domenii noi. În continuare se vor prezenta funcții MATLAB pentru manipularea matricelor și pentru analiza matriceală. Cunoașterea acestora este necesară pentru rezolvarea cu ajutorul MATLAB-ului a diverselor probleme de automatizări, teoria sistemelor și identificarea sistemelor.

1.4.1. Manipularea matricelor

Elementele individuale ale unei matrice pot fi apelate cu numele acesteia, urmat de doi indici, cuprinși între paranteze rotunde și separați prin virgulă. Primul indice semnifică linia, iar al doilea coloana în care se găsește elementul apelat. Indicii pot fi scalari sau vectori. Indicii vectori permit definirea unor submatrice prin care se pot referi părți disparate dintr-o matrice. Utilizarea numai a semnului două puncte „:” în locul indicilor pentru linii sau pentru coloane, presupune considerarea tuturor elementelor pe linii sau respectiv pe coloane. Crearea matricelor mari, precum și manipularea acestora se face cu multă flexibilitate dacă se utilizează indici vectori.

Inversarea coloanelor unei matrice A se face cu instrucțiunea:

```
A=A(:,n:-1:1),
```

iar inversarea liniilor se face cu instrucțiunea:

```
A=A(n:-1:1,:).
```

```
>> A=[1 2;3 4]
```

```
A =
```

```
1    2
3    4
```

```
>> a=A(:,2:-1:1)
```

```
a =
```

```
>> A=[1 2;3 4]
```

```
A =
```

```
1    2
3    4
```

```
>> a=A(2:-1:1,:)
```

```
a =
```

2	1	3	4
4	3	1	2

De o mare importanță este utilizarea fără indici, care are ca efect transformarea matricei într-un vector coloană, citind coloanele una după alta. Astfel, secvența următoare:

```
A=[1 2;3 4]
b=A(:)
```

are ca rezultat transformarea matricei A într-un vector coloană b.

```
>> A=[1 2;3 4]
A =
     1     2
     3     4
>> b=A(:)
b =
     1
     3
     2
     4
```

Dacă în partea stângă a expresiei este asignată o instrucțiune de forma $A(:)$, matricea A trebuie să existe dintr-o utilizare anterioară. Spre exemplu, secvența:

```
A=[1 2;3 4]
A(:)=11:14
```

returnează matricea:

$$A = \begin{bmatrix} 11 & 13 \\ 12 & 14 \end{bmatrix}$$

Pentru extragerea vectorilor cu elemente decupate din alți vectori, se folosesc formele:

- $j:k$ - selectează elementele $[j, j+1, j+2, \dots, k]$ ale unui vector. Dacă $j > k$ vectorul rezultat este gol;
- $j:i:k$ - selectează elementele $[j, j+i, j+2i, \dots, k]$ ale unui vector. Vectorul rezultat este gol dacă $i > 0$ și $j > k$ sau dacă $i < 0$ și $j < k$;

În cazul selectării liniilor și coloanelor matricelor, se folosesc formele:

- $A(:, j)$ - selectează coloana j a matricei A;

- $A(i,:)$ - selectează linia i a matricei A ;
- $A(:,j)$ - selectează toată matricea A ;
- $A(j:k,:)$ - selectează elementele $A(j), A(j+1), \dots, A(k)$ ale vectorului A ;
- $A(:,j:k)$ - selectează toate liniile și coloanele de al j la k , $A(:,j), A(:,j+1), \dots, A(:,k)$, ale matricei A ;
- $A(:)$ - selectează toate elementele matricei A , privite ca o singură coloană (începând cu prima).

Alt caz interesant este extragera submatricelor prin vectori cu elemente 0 și 1. Vectorii cu elemente 0 și 1, creați de operatorii logici sau relaționali, au proprietăți deosebite ce pot conduce la scrierea unor programe foarte compacte. Dacă A este o matrice de dimensiunea $m \times n$ și L este un vector de lungime m , cu elemente 0 și 1, instrucțiunea:

$$B = A(L, :)$$

Returnează în matricea B toate elementele din liniile matricei A pentru care elementul corespunzător ca poziție din vectorul L este 1.

O matrice poate fi definită pe baza altor matrice de dimensiuni inferioare, ca în exemplul de mai jos:

$$C = [A \quad \text{eye}(4) ; \text{ones}(A) \quad A^2]$$

Matricea C este compusă din matricea A (de dimensiune 4×4), matricea identitate (de dimensiune 4×4), matricea unitate (de dimensiune egală cu matricea A) și matricea A ridicată la pătrat. Matricele folosite în asamblare trebuie să fie consistente în dimensiuni, adică să determine blocuri care se integrează compact în tabloul matricei rezultat, în caz contrar se obțin mesaje de eroare.

Redimensionarea unei matrice se poate face cu ajutorul funcției **reshape**. Această funcție redimensionează o matrice A cu dimensiunea $l \times p$ într-o altă matrice B , dimensiunea $m \times n$. Pentru ca această operație să fie posibilă este necesar ca matricea A să aibă același număr de elemente cu matricea B , deci $l \times p = m \times n$. Sintaxa funcției este:

$$B = \text{reshape}(A, m, n)$$

Elementele matricei B (în ordinea succesivă a coloanelor și pe fiecare coloană de sus în jos) sunt elementele matricei argument A (citite de sus în jos și de la stânga la dreapta).

Crearea unei matrice diagonale se face cu funcția **diag**, care se apelează cu următoarea sintaxă:

$$Y = \text{diag}(X, k)$$

unde X este vectorul sau matricea asupra căreia se operează, iar argumentul opțional k indică diagonală acesteia, cu următoarea semnificație:

- $k = 0$ – diagonală principală;
- $k > 0$ – indică diagonală k de deasupra celei principale;
- $k < 0$ – indică diagonală k de sub cea principală;

Dacă x este un vector cu n componente, funcția `diag(X,k)` generează o matrice pătrată de ordinul $n + \text{abs}(k)$, cu elementele lui x pe diagonală k . Dacă x este o matrice, funcția `diag(X,k)` extrage un vector coloană format din elementele diagonalei k a matricei x .

Pentru a crea o matrice superior și inferior triunghiulară, se folosesc funcțiile MATLAB **tril** și **triu**, apelate cu sintaxele:

```
Y=tril(X,k)
Y=triu(X,k)
```

unde argumentele au aceeași semnificație ca mai sus. Funcția `triu(X,k)` înlocuiește cu zero toate elementele matricei x de sub diagonală k . Funcția `tril(X,k)` înlocuiește cu zero toate elementele matricei x de deasupra diagonalei k .

1.4.2. Analiza matriceală

În acest paragraf se vor prezenta funcțiile și modurile de calcul a determinantului unei matrice, a inversei unei matrice, a normei și rangului unei matrice. Calculul determinantului unei matrice se face cu funcția **det** ce are următoarea sintaxă:

```
D=det(X).
```

Prin definiție, inversa unei matrice pătrate A este matricea A^{-1} , care satisface relația:

$$AA^{-1} = A^{-1}A = I$$

unde I este matricea unitate. Se știe că o matrice este inversabilă numai dacă determinantul acesteia este diferit de zero, adică matricea este nesingulară. Inversa se calculează cu funcția **inv**, ce este apelată cu sintaxa:

```
Y=inv(X)
```

Rangul unei matrice reprezintă numărul de linii sau coloane liniar independente ale acesteia și se determină cu funcția **rank**. Această funcție este apelată cu sintaxa:

```
r =rank(X,tol)
```

Ea returnează numărul de valori singulare ale lui x , mai mari decât parametrul opțional `tol`.

Norma unei matrice (sau vector) este un scalar care dă o măsură a mărimii elementelor matricei (sau vectorului). Normele vectorilor sau matricelor se calculează cu funcția **norm**:

```
n = norm(X)
```

Această funcție calculează mai multe tipuri de norme pentru matrici sau vectori, în funcție de sintaxă. Detalii se pot afla consultând help-ul funcției.

1.5. Calcule numerice cu polinoame

În continuare se prezintă câteva funcții MATLAB pentru evaluarea polinoamelor, pentru calculul derivatei unui polinom, pentru generarea unui polinom când se cunosc rădăcinile acestuia precum și pentru descompunerea în fracții simple. Polinomul este o funcție de o singură variabilă, care are următoarea formă generală:

$$f(x) = a_1 x^N + a_2 x^{N-1} + \dots + a_{N-1} x^2 + a_N x + a_{N+1}$$

unde x este variabila, iar a_i ($i = 1 \dots N+1$) sunt coeficienții polinomului. În sintaxa MATLAB polinoamele sunt reprezentate cu un vector linie care conține coeficienții în ordinea descrescătoare a puterilor variabilei. MATLAB-ul oferă mai multe posibilități de evaluare a polinoamelor, cea mai simplă metodă este evaluarea cu scalari, adică pentru o singură valoare a variabilei. A doua metodă constă în evaluarea polinomului în mai multe puncte. În acest caz se efectuează operații cu tablouri. Dimensiunea matricei în care se returnează valorile polinomului trebuie să fie identică cu cea a matricei care conține punctele în care se face evaluarea. A treia metodă de evaluare constă în utilizarea funcției **polyval**, care se apelează cu sintaxa:

```
f = polyval(p,s)
```

Funcția evaluează polinomul definit de vectorul p în punctul s . Dacă s este un vector sau o matrice, polinomul este evaluat pentru fiecare dintre elementele acestuia.

Operațiile aritmetice de adunare și scădere a polinoamelor presupun adunarea și scăderea coeficienților de același ordin. Deoarece în MATLAB aceste operații necesită vectori de aceeași dimensiune, lungimea vectorilor coeficienților este dată de polinomul cu puterea cea mai mare. Odată stabilită această dimensiune, spre exemplu M , vectorii coeficienților polinoamelor care au puterea maximă N , mai mică decât M , vor fi completați la stânga cu coeficienții zero până la dimensiunea M .

Înmulțirea a două polinoame este echivalentă unei operații de convoluție și se realizează cu funcția MATLAB **conv**:

```
c = conv(a,b)
```

unde: a și b sunt vectorii polinoamelor care se înmulțesc, iar c este vectorul coeficienților polinomului produs. Împărțirea a două polinoame este echivalentă unei operații de deconvoluție și se realizează cu funcția MATLAB **deconv**:

```
[d,r] = deconv(a,b)
```

unde: a și b sunt vectorii polinoamelor deîmpărțit și împărțitor, d este vectorul coeficienților polinomului cât, iar r este vectorul polinomului rest.

Descompunerea în fracții simple reprezintă scrierea raportului a două polinoame ca sumă de fracții cu polinoame de ordinul unu.

$$\frac{B(x)}{A(x)} = \frac{r(1)}{x-p(1)} + \frac{r(2)}{x-p(2)} + \dots + \frac{r(n)}{x-p(n)} + k(x)$$

dacă polul $p(j)$ este de ordinul m , descompunerea conține termeni de forma:

$$\frac{r(j)}{x-p(j)} + \frac{r(j+1)}{(x-p(j))^2} + \dots + \frac{r(j+m-1)}{(x-p(j))^m}$$

Descompunerea în fracții simple se realizează cu funcția MATLAB **residue**, ce se apelează cu sintaxele:

`[r,p,k]=residue(B,A)` și `[B,A]=residue(r,p,k)`

unde: A și B sunt vectorii coeficienților polinoamelor de la numitor și numărător, r este vectorul coloană al rezidurilor, p este vectorul coloană al polilor, iar k este vectorul linie al termenilor liberi.

Pentru a calcula derivata unui polinom în MATLAB se utilizează funcția **polyder**, care se apelează cu sintaxele:

`D=polyder(C)`

$$D(x) = C'(x)$$

`D=polyder(A,B)`

$$D(x) = (A(x) \cdot B(x))' = A'(x) \cdot B(x) + A(x) \cdot B'(x)$$

`[M,N]=polyder(A,B)`

$$\frac{M(x)}{N(x)} = \left(\frac{A(x)}{B(x)} \right)' = \frac{A'(x) \cdot B(x) - A(x) \cdot B'(x)}{(B(x))^2}$$

Dacă funcția $f(x)$ este un polinom de gradul N , atunci $f(x)=0$ are N rădăcini, care pot fi reale sau complexe. În cazul în care coeficienții polinomului sunt numere reale, rădăcinile complexe sunt perechi complex conjugate. Funcția MATLAB **roots** determină rădăcinile polinoamelor și se apelează cu sintaxele:

`r = roots(p)`

Funcția MATLAB **poly** determină coeficienții unui polinom ale cărui rădăcini sunt cunoscute, se apelează cu sintaxa:

`p = poly(r)`

În ambele cazuri peste un vector linie care conține coeficienții polinomului, iar r este un vector coloană care conține rădăcinile polinomului.

1.6. Exerciții propuse

Exercițiul 1.

Fie următorul sistem de trei ecuații cu trei necunoscute:

$$3x + 2y - z = 10$$

$$-x + 3y + 2z = 5$$

$$x - y - z = -1$$

Să se rezolve sistemul prin împărțirea matricelor. Descrierea matriceală a sistemului este următoarea:

$$A \cdot X = B$$

unde: A este matricea coeficienților necunoscutelor, de dimensiune 3×3 . Coeficienții aceleiași necunoscute sunt pe aceeași coloană. X este matricea necunoscutelor, de dimensiune 3×1 . B este matricea termenilor liberi, de dimensiune 3×1 .

Exercițiul 2.

Să se rezolve sistemul de la exercițiul 1 folosind matricea inversă.

Exercițiul 3.

Fie polinoamele:

$$f(x) = 4x^6 - 2x^4 + 3x^2 - x + 4$$

$$g(x) = 2x^2 + 5x - 16$$

Să se calculeze:

a) $s(x) = f(x) + g(x)$ și $d(x) = f(x) - g(x)$

b) $a = f(3) - g(7)$; $f(b)$ și $g(b)$ unde $b = \{1, 3, 4, 9\}$

c) $f(x) \cdot g(x)$ și $f(x) / g(x)$

d) $(f(x) \cdot g(x))'$ și $(f(x) / g(x))'$

e) să se rezolve: $f(x) = 0$ și $g(x) = 0$

Exercițiul 4.

Să se descompună în fracții simple expresia:

$$\frac{B(x)}{A(x)} = \frac{x^3 + 2x^2 - 2}{x^2 + 1}$$

Exercițiul 5.

Să se determine o matrice H compusă din matricea $G = \begin{pmatrix} 11 & 12 \\ 4 & 6 \end{pmatrix}$, de dimensiune 2×2 , din matricea identitate 2×2 , matricea $2 \cdot G$ și matricea unitate de dimensiunile matricei G .

Exercițiul 6.

Se consideră matricea: $A = \begin{pmatrix} 11 & 12 & 13 & 14 \\ 2 & 3 & 4 & 5 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \end{pmatrix}$.

- Să se determine un vector B , care are ca elemente, al doilea, al treilea și al patrulea termen de pe linia a doua a matricei A ;
- Să se determine un vector C , care are ca elemente linia a doua din matricea A ;
- Să se determine o matrice D , care are ca elemente prima linie și a treia din matricea A ;
- Să se determine o matrice F , ce conține elementele matricei A , în care s-a schimbat coloana a treia cu coloana a doua;
- Să se șteargă a doua coloană a matricei A ;
- Să se determine un vector E , format din toate elementele matricei A .