# CALCUL NUMERIC ÎN MATLAB

***OBIECTIVE***

- Introducerea şi manipularea matricelor şi vectorilor.
- Operații şi funcții cu matrice.
- Introducerea şi manipularea polinoamelor.

## 2.1. Operații aritmetice

În Matlab, calculele aritmetice asupra tablourilor de date pot fi: operații după regulile calculului matriceal (operații cu matrice) şi operații după regulile calculului scalar (operații cu vectori).

În tabelul 2.1.1 sunt prezentați operatorii aritmetici din Matlab.

Tab. 2.1.1 Operatorii aritmetici Matlab

| Operația | Scalari | Matrice | Vectori |
|---|---|---|---|
| Adunarea | + | + | .+ |
| Scăderea | - | - | .- |
| Înmulțirea | * | * | .* |
| Împărțirea la stânga | \ | \ | .\ |
| Împărțirea la dreapta | / | / | ./ |
| Ridicarea la putere | ^ | ^ | .^ |
| Transpunerea | ‘ | ‘ | .’ |

Ordinea operațiilor aritmetice este aceeaşi cu cea cunoscută în matematica elementară, a operațiilor aritmetice standard (tabelul 2.1.2). Deşi variabilele memorate au un interval foarte mare, cel mai adesea sunt în limitele $10^{-308}$ şi $10^{308}$ (vezi funcțiile Matlab **realmax** şi **realmin**). Dacă rezultatul unui calcul este mai mare de $10^{308}$ (uneori, este posibil ca rezultatul unei expresii să depăşească limitele menționate anterior), Matlab-ul înregistrează $\infty$ (Inf). Dacă rezultatul unui calcul este mai mic decât $10^{-308}$, calculatorul înregistrează valoarea zero. În Matlab, rezultatul împărțirii cu zero este $\infty$. În acest caz este afişat mesajul de atenționare „Warning: Divide by zero”, dar calculele sunt continuate cu operandul $\infty$.

```
>> x=2.5*10^200;          >> x=2.5*10^200;
>> y=10^200;              >> y=0;
>> z=x*y                  >> z=x/y
z =                       Warning: Divide by zero.
Inf                       Z =Inf
```

Tab. 2.1.2 Ordinea operațiilor aritmetice

| Ordinea | Operația |
|---|---|
| 1 | parantezele |
| 2 | ridicarea la putere |
| 3 | înmulțirea şi împărțirea |
| 4 | adunarea şi scăderea |

### 2.1.1. Operații aritmetice cu scalari

În tabelul 2.1.3 sunt prezentate operațiile aritmetice între doi scalari, fiind prezentată atât forma algebrică cât şi forma Matlab.

Tab. 2.1.3 Forma Matlab a operatorilor scalari

| Operația | Forma algebrică | Forma Matlab |
|---|---|---|
| Adunarea | a+b | a+b |
| Scăderea | a-b | a-b |
| Înmulțirea | a x b | a*b |
| Împărțirea la dreapta | a:b | a/b |
| Împărțirea la stânga | b:a | a\b |
| Ridicarea la putere | $a^b$ | a^b |

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

### 2.1.2. Operații aritmetice cu vectori

Operațiile cu vectori sunt operații aritmetice între elementele situate în aceeaşi poziție a vectorilor, 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 aceeaşi dimensiune, se utilizează operatorul corespunzător operației precedat de punct (vezi tabelul 2.1.1). Dacă unul dintre operanzi este un scalar, acesta operează cu fiecare element al vectorului.

### 2.1.3. Operații aritmetice cu matrice

Operațiile uzuale de algebră liniară cu matrice sunt simbolizate cu semnele grafice prezentate în tabelul 2.1.1 şi 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 şi Y, rezultând elementele:

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

Matricele X şi Y trebuie să aibă aceeaşi 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 a două matrice este simbolizată cu operatorul minus. Instrucțiunea:

```
Z=X-Y,
```

reprezintă scăderea matricelor X şi Y, rezultând elementele:

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

Matricele X şi Y trebuie să aibă aceeaşi 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 există două cazuri: împărțirea la dreapta şi 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 şi Y, şi este identică cu:

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

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

```
Z=X\Y,
```

reprezintă împărţirea la stânga a matricelor X şi Y şi este identică cu:

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

Dacă unul dintre operanzi este un scalar, operaţia nu este posibilă. Operaţia de ridicare la putere a unei matrice este simbolizată cu operatorul ^. Următoarea instrucţiune:

`Z = Xp,`

reprezintă ridicarea la puterea p a matricei X. Expresia are sens doar pentru X matrice pătrată şi pentru p scalar. Dacă p este un întreg pozitiv, ridicarea la putere este obţinută prin înmulţiri repetate, iar dacă p este un întreg negativ, matricea X este mai întâi inversată şi apoi se înmulţesc inversele de p ori.

Operaţia de transpunere a unei matrice este simbolizată cu operatorul apostrof. Cu instrucţiunea:

`Z=Y',`

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

$$Z(i,j) = conj(Y(i,j)) = real(Y(i,j)) - i * imag(Y(i,j))$$

## 2.2. Generarea matricelor

Deşi în Matlab nu există instrucţiuni pentru declararea tipurilor de variabile, matricele se autodimensionează în timpul utilizării. Pentru a creşte 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 instrucţiunea:

`X=[],`

care asignează lui X matricea de dimensiuni 0x0. Orice matrice goală trebuie să aibă cel puţin una din dimensiuni zero.

Pentru a afla dacă o matrice este goală, se foloseşte funcţia Matlab **isempty** care se apelează cu sintaxa:

`r=isempty(X).`

Funcţia întoarce rezultatul `r=1` dacă matricea este goală şi `r=0` în caz contrar.

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

`U=ones(m,n),`

unde m şi n sunt scalari ce definesc dimensiunea matricei.

Pentru a genera matricea zero se foloseşte funcţia **zeros** ce se apelează cu sintaxa:

`O=zeros(m,n).`

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

`I=eye(m,n).`

## 2.3. Generarea vectorilor cu pas liniar şi cu pas logaritmic

În Matlab se pot genera vectori cu pas liniar sau vectori cu pas logaritmic. Generarea vectorilor cu pas liniar implică cunoaşterea limitelor intervalului $(a_{\min};a_{\max})$ şi a pasului dintre două elemente sau a numărului de elemente ale vectorului (N). Dacă se cunosc limitele intervalului şi pasul dintre două elemente, vectorul se generează cu instrucţiunea:

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

Numărul de elemente al vectorului x este: $N=\left[\frac{a_{\max}-a_{\min}}{pas}\right]+1$,

unde [] 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 şi numărul de elemente ale vectorului, acesta se generează cu ajutorul funcţiei **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 funcţia **logspace** ce se apelează cu sintaxa:

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

Vectorul `x` conţine `N` elemente distribuite logaritmic între decadele [$10^{\mathtt{amin}}$; $10^{amax}$]. Dacă `N` nu este precizat, este luat implicit 50.

## 2.4. Calcule cu matrice

Formularea matriceală a problemelor a condus la simplificarea metodelor de rezolvare şi a făcut posibilă extinderea unor soluţii deja cunoscute la domenii noi. În continuare se vor prezenta funcţii Matlab pentru manipularea matricelor şi pentru analiza matriceală. Cunoaşterea acestora este necesară pentru rezolvarea cu ajutorul Matlab-ului a diverselor probleme de automatizări, teoria sistemelor şi identificarea sistemelor.

### 2.4.1. Manipularea matricelor

Elementele individuale ale unei matrice pot fi apelate cu numele acesteia urmat de doi indici, cuprinşi între paranteze rotunde şi separaţi prin virgulă. Primul indice semnifică linia, iar al doilea coloana în care se găseşte elementul apelat. Indicii pot fi scalari sau vectori. Indicii vectori permit definirea unor submatrice prin care se pot referi părţi disparate dintr-o matrice. Utilizarea 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 şi 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 =
     1     2                       1     2
     3     4                       3     4
>> a=A(:,2:-1:1)              >> a=A(2:-1:1,:)
a =                           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:

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

În cazul selectării liniilor şi coloanelor matricelor, se pot utiliza opțiunile:

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

Alt caz interesant este extragerea submatricelor prin vectori cu elemente 0 şi 1. Vectorii cu elemente 0 şi 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×n şi L este un vector de lungime m, cu elemente 0 şi 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 eye(4);ones(A) 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) şi 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 vor rezulta mesaje de eroare.

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

```
B=reshape(A,m,n).
```

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

Crearea unei matrice diagonale se face cu funcția **diag**, folosind sintaxa:

```
Y=diag(X,k),
```

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

- `k=0` - diagonala principală;
- `k>0` - indică diagonala k de deasupra celei principale;
- `k<0` - indică diagonala k de sub cea principală.

Dacă `X` este un vector cu n componente, funcția `diag(X,k)` generează o matrice pătrată de ordinul `n + abs(k)`, cu elementele lui `X` pe diagonala `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 şi inferior triunghiulară, se folosesc funcțiile Matlab **tril** şi **triu**, apelate cu sintaxele:

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

unde argumentele au aceeaşi semnificație ca mai sus.

Funcția `triu(X,k)` înlocuieşte cu zero toate elementele matricei `X` de sub diagonala `k`. Funcția `tril(X,k)` înlocuieşte cu zero toate elementele matricei `X` de deasupra diagonalei `k`.

### 2.4.2. Analiza matriceală

În acest paragraf se vor prezenta funcțiile şi modurile de calcul ale determinantului unei matrice, ale inversei unei matrice, ale normei şi 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:

$$AA^{-1} = A^{-1}A = \mathrm{I},$$

unde I este matricea unitate. Se ştie 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 şi 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 a unui 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.

## 2.5. Calcule numerice cu polinoame

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

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

unde $x$ este variabila, iar $a_i$ $(i = 1 \ldots N+1)$ sunt coeficienții polinomului. În sintaxa Matlab, polinoamele sunt reprezentate cu un vector linie care conține coeficienții în ordinea descrescătoare puterilor variabilei. Matlab-ul oferă mai multe posibilități de evaluare a polinoamelor, cea mai simplă metodă fiind 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 vectori. 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 uitilizarea 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 şi scădere ale polinoamelor presupun adunarea şi scăderea coeficienților de acelaşi ordin. Deoarece în Matlab aceste operații necesită vectori de aceeaşi 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 şi se realizează cu funcția Matlab **conv**:

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

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

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

unde:a şi b sunt vectorii polinoamelor deîmpărțit şi 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.

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

Dacă polul $p(j)$ (reprezintă una din rădăcinile numitorului) este de ordinul $m$, descompunerea conține termeni de forma:

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

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

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

unde: `A` şi `B` sunt vectorii coeficienților polinoamelor de la numitor şi numărător, `r` este vectorul coloană al resturilor, `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)` $D(x)=\left(A(x)\cdot B(x)\right)'=A'(x)\cdot B(x)+A(x)\cdot B'(x)$

`[M,N]=polyder(A,B)` $\dfrac{M(x)}{N(x)}=\left(\dfrac{A(x)}{B(x)}\right)'=\dfrac{A'(x)\cdot B(x)-A(x)\cdot B'(x)}{(B(x))^2}$

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 şi se apelează cu sintaxele:

```
r=roots(p).
```

Funcția Matlab **poly** ce determină coeficienții unui polinom al cărui rădăcini sunt cunoscute, se apelează cu sintaxa:

```
p=poly(r).
```

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

## 2.6. Exerciții propuse

*Exercițiul 2.6.1*

Fie următorul sistem de trei ecuații cu trei necunoscute:

$$\begin{cases} 3x+2y-z=10 \\ -x+3y+2z=5 \\ x-y-z=-1 \end{cases}$$

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

$A * X = B$,

unde: $A$ este matricea coeficienților necunoscutelor, de dimensiune 3×3. Coeficienții aceleiaşi necunoscute sunt pe aceeaşi coloană. $X$ este matricea necunoscutelor (3×1), de dimensiune. $B$ este matricea termenilor liberi, de dimensiune 3×1.

*Exercițiul 2.6.2*

Să se rezolve sistemul de la exercițiul 2.6.1 folosind matricea inversă.

*Exercițiul 2.6.3*

Fie polinoamele:

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

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

Să se calculeze:

- $s(x) = f(x) + g(x)$ şi $d(x) = f(x) - g(x)$.
- $a = f(3) - g(7)$; $f(b)$ şi $g(b)$ unde $b = \{1,3,4,9\}$.
- $f(x) * g(x)$ şi $f(x) / g(x)$.
- $(f(x) * g(x))'$ şi $(f(x) / g(x))'$.
- Să se rezolve: $f(x) = 0$ şi $g(x) = 0$.

*Exercițiul 2.6.4*

Să se descompună în fracții simple expresia:

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

*Exercițiul 2.6.5*

Să se determine o matrice $H$ compusă din matricea $G = \begin{bmatrix} 11 & 12 \\ 4 & 6 \end{bmatrix}$ de dimensiune 2x2, matricea identitate 2x2, matricea 2*G şi matricea unitate de dimensiunile matricei G.

*Exercițiul 2.6.6*

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

- Să se determine un vector $B$, care are ca elemente, al doilea, al treilea şi 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 şi a treia linie 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 şteargă a doua coloană a matricei $A$.
- Să se determine un vector $E$, format din toate elementele matricei $A$.