Lucrarea de laborator 1
Introducere în Matlab

Obiectivele lucrării

- familiarizarea cu mediul de programare Matlab,
- recapitularea unor elemente de calcul vectorial şi de calcul matriceal,
- fixarea cunoştinţelor privitoare la rezolvarea problemelor de calcul vectorial şi calcul matriceal folosind mediul de programare Matlab,

prin studierea unor exemple şi prin rezolvarea unor probleme.

1. Exemple de studiat

Exemplul 1.1: Utilizând help-ul în linia de comandă, să se găsească funcţia Matlab pentru calculul arctangentei şi să se specifice sintaxa de apel a acesteia.

Soluţie.: Rezolvarea problemei constă din următorii paşi:

i) Căutarea unui director cu funcţii trigonometrice. Pentru aceasta se foloseşte comanda:

```
>> help
```

Nu există un director destinat doar funcţiilor trigonometrice. Funcţia arctangentă este însă o funcţie elementară, deci se găseşte probabil în directorul:

```
matlab\elfun         -  Elementary math functions.
```

ii) Vizualizarea conţinutului directorului *elfun*:

```
>> help elfun
```

În lista funcţiilor Matlab din acest director se află şi funcţia căutată:

```
atan        - Inverse tangent.
```

iii) Aflarea sintaxei de apel a funcţiei:

```
>> help atan
```

Din help-ul funcţiei *atan* rezultă că sintaxa de apel este:

```
atan(X)
```

care returnează arctangentele, exprimate în radiani, ale elementelor vectorului *X*.

Observaţii: 1. Denumirile funcţiilor Matlab se scriu cu litere mici. Ele apar scrise în help cu litere mari doar pentru a fi scoase în evidenţă.

2. Dacă s-ar fi presupus că denumirea funcției arctangentă din Matlab este aceeaşi cu cea folosită în matematică, arctg, şi s-ar fi apelat help-ul funcției cu această denumire:

```
>> help arctg
```

Matlab ar fi afişat un mesaj de eroare, anunțând utilizatorul că nu găseşte fişierul-M *`arctg`* (*`arctg.m not found.`*), ceea ce arată că denumirea funcției Matlab şi denumirea fişierului în care este implementată funcția trebuie să fie identice.

Exemplul 1.2: Să se extragă din matricea M= $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ -1 & -2 & -3 & -4 \\ 0 & 5 & 0 & 7 \end{bmatrix}$ elementul de pe linia 1 şi coloana 3, prima linie, coloana a 2-a, submatricea determinată de liniile 1,2 şi 4 şi de coloanele 2, 3 şi 4.

Soluție.: Rezolvarea problemei constă din următorii paşi:

i) Introducerea matricei M:

```
>> M=[1 2 3 4; 2 4 6 8; -1 -2 -3 -4; 0 5 0 7];
```

ii) Extragerea elementului de pe linia 1 şi coloana 3:

```
>> M(1,3)
ans =     3
```

iii) Extragerea primei linii:

```
>> M(1,:)
ans =
     1     2     3     4
```

iv) Extragerea coloanei a 2-a:

```
>> M(:,2)
ans =
     2
     4
    -2
     5
```

v) Extragerea submatricei determinată de liniile 1,2 şi 4 şi de coloanele 2, 3 şi 4:

```
>> M([1,2,4],2:4)
ans =
     2     3     4
     4     6     8
     5     0     7
```

Observații: 1. Rezultatele extragerilor nu au fost atribuite niciunei variabile. În acest caz, Matlab a făcut atribuirea implicită a rezultatului fiecărei operații de extragere variabilei *ans*.

2. Dacă se doreşte extragerea tuturor elementelor unei linii, în locul vectorului ce indică coloanele de pe care se face extragerea se foloseşte forma prescurtată „:". O observație asemănătoare este valabilă şi în caziul extragerii unei coloane.

3. Matricea M nefiind reatribuită în mod explicit, nu a fost afectată de nici una din operațiile de mai sus. Acest lucru se poate vedea afişând la sfârşit conținutul matricei:

```
>> M
M =
     1     2     3     4
     2     4     6     8
    -1    -2    -3    -4
     0     5     0     7
```

Exemplul 1.3: Să se genereze matricele pătratice A şi B, de ordinul 4, definite prin relațiile de mai jos. Să se afişeze suma lor, produsul lor, cubul matricei A, rezultatul împărțirii la stânga a matricei A prin B şi rangul matricei B. Rezolvarea problemei să se facă prin utilizarea unui fişier script.

$$A_{i,j} = \frac{1}{i+j}, \quad i,j = \overline{1,4}$$

$$B_{i,j} = \begin{cases} 1, & i = j \\ i+j, & i > j, \quad i,j = \overline{1,4} \\ i-j, & i < j \end{cases}$$

<u>*Soluție.*</u>: Etapele rezolvării problemei sunt următoarele:

i) Crearea unui fişier-M, numit, de exemplu, `op_matr.m`:

```
>> edit op_matr
```

ii) Rezolvarea cerințelor problemei prin scrierea comenzilor în fişierul script:

```
% generarea matricei A
for i=1:4
    for j=1:4
        A(i,j)=1/(i+j);
    end
end
% generarea matricei B
for i=1:4
```

```
    for j=1:4
        if i==j B(i,j)=1;
        elseif i>j B(i,j)=i+j;
        else B(i,j)=i-j;
        end
    end
end
% afisarea matricei A
A
% afisarea matricei B
disp('matricea B')
disp(B)
% calculul si afisarea sumei
Suma=A+B
% calculul si afisarea produsului
Produs=A*B
% calculul si afisarea cubului matricei A
cub_A=A^3
% calculul si afisarea rezultatului impartirii la stanga
disp('Rezultatul impartirii A\B este')
Rez=A\B
% rangul matricei B
rang_B=rank(B)
```

Se recomandă salvarea constantă a fişierului.

iii) Rularea fişierului script se face din linia de comandă:

```
>> op_matr
```

Rularea unui fişier-M din linia de comandă are ca efect rularea ultimelei versiuni salvate a fişierului.

iv) Vizualizarea rezultatelor afişate în fereastra de comenzi:

```
A =
    0.5000    0.3333    0.2500    0.2000
    0.3333    0.2500    0.2000    0.1667
    0.2500    0.2000    0.1667    0.1429
    0.2000    0.1667    0.1429    0.1250
matricea B
     1    -1    -2    -3
     3     1    -1    -2
     4     5     1    -1
     5     6     7     1
Suma =
    1.5000   -0.6667   -1.7500   -2.8000
    3.3333    1.2500   -0.8000   -1.8333
    4.2500    5.2000    1.1667   -0.8571
    5.2000    6.1667    7.1429    1.1250
Produs =
    3.5000    2.2833    0.3167   -2.2167
    2.7167    1.9167    0.4500   -1.5333
    2.2310    1.6405    0.4667   -1.1738
```

```
    1.8964    1.4310    0.4512   -0.9512
cub_A =
    0.4516    0.3263    0.2571    0.2127
    0.3263    0.2358    0.1859    0.1538
    0.2571    0.1859    0.1465    0.1213
    0.2127    0.1538    0.1213    0.1004
Rezultatul impartirii A\B este
Rez =
  1.0e+004 *
   -0.0600    0.2380   -0.4940   -0.1420
    0.4620   -1.5900    3.7980    1.0920
   -0.9660    2.8980   -7.7280   -2.2260
    0.5880   -1.5680    4.5640    1.3160
rang_B =     4
```

Exemplul 1.4: Să se scrie o funcție Matlab care primeşte ca argumente trei valori a,b,p şi care generează vectorul cu pas liniar v=a:p:b. Funcția returnează vectorul generat, lungimea sa, precum şi suma elementelor sale. Să se testeze funția creată pentru următoarele triplete de valori: (0,25,5), (2,19,3), (5,-3,-2), (5,-2,0), (2,19,0).

Soluție.: Etapele rezolvării problemei sunt următoarele:

i) Crearea unui fişier-M, numit, de exemplu, `vect_lin.m`:

```
>> edit vect_lin
```

ii) Rezolvarea cerințelor problemei prin scrierea comenzilor în fişierul funcție:

```
function [v,lung,suma]=vect_lin(a,b,p)
v=a:p:b;
lung=length(v);
suma=sum(v);
```

iii) Rularea fişierului funcție pentru tripletele de valori cerute şi vizualizarea rezultatelor:

```
>> [v,lung,suma]=vect_lin(0,25,5)
v =
     0     5    10    15    20    25
lung =     6
suma =    75
>> amin=2; amax=19; pas=3;
>> [v,lung,suma]=vect_lin(amin,amax,pas)
v =
     2     5     8    11    14    17
lung =     6
suma =    57
>> a=5; b=-3; p=-2;
>> [v,lung,suma]=vect_lin(a,b,p)
v =
     5     3     1    -1    -3
```

```
lung =     5
suma =     5
>> [v,lung,suma]=vect_lin(5,-2,0)
v =
   Empty matrix: 1-by-0
lung =     0
suma =     0
>> [v,lung,suma]=vect_lin(2,19,0)
v =
   Empty matrix: 1-by-0
lung =     0
suma =     0
```

Exemplul 1.5: Să se scrie o funcție Matlab care primeşte ca argumente doi vectori de lungimi egale cu 3, v şi w, şi returnează normele euclidiene ale vectorilor, produsul lor scalar, produsul lor vectorial şi unghiul dintre cei doi vectori.

Soluție.: Etapele rezolvării problemei sunt următoarele:

i) Crearea fişierului-M:

```
>> edit vectori
```

ii) Scrierea comenzilor în fişierul-M creat:

```
function [n_v,n_w,ps,pv,unghi]=vectori(v,w)
if length(v)~=3 | length(w)~=3
    disp('Vectorii nu satisfac conditia de lungime 3!')
    n_v=[]; n_w=[]; ps=[]; pv=[]; unghi=[];
    return;
end
n_v=norm(v); n_w=norm(w);
ps=dot(v,w); pv=cross(v,w);
if n_v==0 | n_w==0
    disp('Unghiul nu poate fi calculat, unul din vectori fiind 
zero.')
    unghi=[];
else
    unghi=ps/(n_v*n_w);
end
```

iii) Testarea programului pentru diferite perechi de vectori:

```
>> v=[1 -1 3]; w=[0 3 -2];
>> [n_v,n_w,ps,pv,unghi]=vectori(v,w)
n_v =    3.3166
n_w =    3.6056
ps =    -9
pv =    -7     2     3
unghi =   -0.7526
>> a=[1 2 3]; b=[0 0 0]; c=[-1 -2];
>> [n_v,n_w,ps,pv,unghi]=vectori(a,b)
Unghiul nu poate fi calculat, unul din vectori fiind zero.
```

```
n_v =    3.7417
n_w =     0
ps =     0
pv =     0     0     0
unghi =     []
>> [n_v,n_w,ps,pv,unghi]=vectori(a,c)
Vectorii nu satisfac conditia de lungime 3!
n_v =     []
n_w =     []
ps =     []
pv =     []
unghi =     []
```

Observații: 1. Este necesară verificarea respectării condițiilor impuse argumentelor transmise funcției (în acest caz, faptul că cei doi vectori transmişi ca parametric să aibă fiecare lungimea egală cu 3).

2. La testare trebuie considerate atât cazul general, cât şi cazurile particulare care generează probleme.

Exemplul 1.6: Pentru o matrice pătratică transmisă ca parametru unei funcții Matlab se cere să se scrie o secvență de instrucțiuni prin intermediul căreia: să se specifice dacă matricea este inversabilă; în cazul în care matricea este inversabilă, să se afişeze inversa ei, iar în cazul în care matricea nu este inversabilă, să se afişeze rangul ei.

Soluție.: Rezolvarea constă din următoarele etape:

i) Crearea fişierului-M:

```
>> edit calcul_matr
```

ii) Scrierea codului sursă în fişier:

```
function calcul_matr(M)
[lin,col]=size(M);
if lin~=col
    disp('Matricea nu este patratica!')
    return
end
if det(M)~=0
    disp('Matricea este inversabila.')
    inversa=inv(M)
else
    disp('Matricea nu este inversabila.')
    rang=rank(M)
end
```

iii) Testarea programului:

```
>> A=[1 0; -1 1]; B=[1 1; 2 2]; C=[1 1 2; 2 2 3];
>> calcul_matr(A)
```

```
Matricea este inversabila.
inversa =
     1     0
     1     1
>> calcul_matr(B)
Matricea nu este inversabila.
rang =     1
>> calcul_matr(C)
Matricea nu este patratica!
```

2. Probleme de rezolvat

P1.1. Fie A=[2 5 6], B=[4 3 2], p=2. Să se calculeze: C=A.*B, D=A./B, E=A.\B, F=A.^B, G=B.^p, H=p.^A.

P1.2. Fie Z=$\begin{bmatrix} 1 & 1+i \\ i & 3-2*i \end{bmatrix}$. Să se calculeze Z1=Z.' şi Z2=Z' şi să se constate diferenţele apărute.

P1.3. Se cere:

- să se determine codurile ASCII ale şirului de caractere 'Matlab';
- să se compare şirurile 'test' şi 'Test' şi să se afişeze rezultatul;
- să se afle poziţia şirului de caractere S2='test' în şirul de caractere S1='Acest test este interesant.'.

P1.4. Fie matricele A=$\begin{bmatrix} 1 & -i & 0 \\ 2-i & 0 & 1+i \end{bmatrix}$ şi B=$\begin{bmatrix} 1 & -i & i \\ 2-i & 2 & 1-5i \end{bmatrix}$; să se realizeze comparaţiile: C=A>=B şi D=A= =B.

P1.5. Să se genereze:

- matricea Hilbert de ordinul n=4, ale cărei elemente sunt date de expresia:

$$H_{i,j} = \frac{1}{i+j-1}, \quad i,j=\overline{1,n};$$

- o matrice A cu 4 linii şi 5 coloane, ale cărei elemente sunt:

$$A_{i,j} = \begin{cases} 3, & dacă\ i=j \\ -3, & dacă\ |i-j|=2 \\ 1, & dacă\ i+j=3 \\ 0, & în\ rest \end{cases}, \quad i=\overline{1,4},\ j=\overline{1,5}.$$

P1.6. Să se scrie un program, utilizând o buclă *while*, care calculează produsul elementelor vectorului x=[5 2 9 10 -1 9 –1] până când întâlneşte un număr negativ.

P1.7. Se consideră următoarea matrice:

$$A = \begin{bmatrix} 3 & 1 & 0 & -2 & -9 \\ 0 & -5 & 4 & 8 & 10 \\ 6 & 6 & 1 & -5 & 7 \\ 11 & -2 & 6 & 9 & 4 \\ -8 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Se cere să se extragă linia a 3-a, ultima coloană, ultima linie, submatricea determinată de liniile 2-4 şi coloanele 1-3.

P1.8. Aceleaşi cerințe de la problema P1.7. pentru o matrice cu cel puțin 4 linii şi 4 coloane transmisă ca argument unei funcții Matlab.

P1.9. Fie vectorii:

$$\vec{v} = 2\vec{i} - \vec{j} + 3\vec{k}, \qquad \vec{w} = 3\vec{j} - 2\vec{k}$$

Să se determine normele euclidiene, produsul scalar, cosinusul unghiului şi produsul vectorial pentru cei doi vectori.

P1.10. Să se determine transpusa, rangul şi determinantul următoarei matrice:

$$A = \begin{bmatrix} 3 & 1 & 0 & -2 & -9 \\ 0 & -5 & 4 & 8 & 10 \\ 6 & 6 & 1 & -5 & 7 \\ 11 & -2 & 6 & 9 & 4 \\ -8 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Este matricea A inversabilă? În caz afirmativ, să se determine inversa matricei A.

P1.11. a) Să se scrie o funcție Matlab care primeşte ca argument o matrice pătratică şi afişează transpusa, rangul şi determinantul matricei.

b) Să se scrie o funcție Matlab care primeşte ca argument o matrice pătratică nesingulară şi care returnează determinantul şi inversa matricei.

P1.12. Să se realizeze factorizările LR, respectiv QR, pentru matricela de la problema P1.7. Să se verifice ortogonalitatea matricei Q obținute.

3. Întrebări recapitulative

Î1.1. Care sunt ferestrele cu care ați lucrat în Matlab?

Î1.2. Care este elementul de bază în Matlab şi de la ce vine denumirea de „Matlab”?

Î1.3. Care utilizări ale operatorului Matlab „;” (punct-virgulă) cunoaşteți (unde l-ați utilizat şi ce efect a avut)?

Î1.4. Care sunt cele două tipuri de operații asupra matricelor în Matlab şi care este diferența de notare a operatorilor corespunzători celor două tipuri de operații?

Î1.5. Specificați care este funcția Matlab pentru generarea matricei identitate I_n şi modurile de apel.

Î1.6. Specificați care este comanda Matlab pentru generarea unui interval real [a,b].

Î1.7. Ce se înțelege prin „vector” în Matlab?

Î1.8. Fie x un număr real. Specificați instrucțiunea de calcul a lui e^x în Matlab.

Î1.9. Fie x un număr real strict pozitiv. Specificați instrucțiunea de calcul a lui ln(x) în Matlab.

Î1.10. Precizați care este funcția Matlab pentru calculul arctangentei.

Î1.11. Precizați care este funcția Matlab pentru calculul cosinusului hiperbolic.

Î1.12. Precizați cum se poate ieşi forțat dintr-o buclă infinită.

Î1.13. Definiți noțiunea de „transpusă a unei matrice”.

Î1.14. Precizați proprietățile operației de transpunere a matricelor.

Î1.15. Precizați care este diferența dintre operatorii ` (apostrof) şi .` (punct-apostrof).

Î1.16. Definiți noțiunea de „matrice superior triunghiulară”.

Î1.17. Definiți noțiunea de „matrice inferior triunghiulară”..

Î1.18. Definiția noțiunea de „matrice ortogonală”.

Î1.19. Precizați proprietățile operației de inversare a matricelor.

Î1.20. Ce se înțelege prin „factorizarea unei matrice pătratice”?

Î1.21. Ce tipuri de factorizări ale matricelor pătratice cunoaşteți?

Î1.22. Ce se înțelege prin „norma euclidiană” a unui vector?

Î1.23. Precizați la modul general cum se calculează produsul scalar a doi vectori cunoscuți prin proiecțiile lor pe axele sistemului de coordonate.

Î1.24. Precizați la modul general cum se calculează produsul vectorial a doi vectori cunoscuți prin proiecțiile lor pe axele sistemului de coordonate.

Î1.25. Definiți noțiunea de „produs vectorial a doi vectori”.

Î1.26. Definiți noțiunea de „unghi dintre doi vectori”.

Î1.27. Precizați care sunt funcțiile Matlab pentru determinarea dimensiunilor unei matrice, respectiv a unui vector, precum şi modul lor de apelare.