

Laborator 1 - RNSF

# Introducere în OCTAVE

## 1. Introducere

GNU Octave reprezintă un limbaj de programare de nivel înalt, destinat cu precădere calculelor numerice și reprezentărilor grafice în domeniul științei și ingineriei. Acesta integrează analiza numerică, calculul matriceal, procesarea semnalului și reprezentările grafice. Octave reprezintă implementarea opensource pentru o parte din bibliotecile de funcții MATLAB.

Elementul de bază cu care operează Octave este matricea. În acest limbaj de programare nu se declară variabile (ca în cazul limbajului C) și nici nu se folosesc tipuri de date predefinite, deoarece orice set de date manipulat de utilizator este văzut ca o matrice (o zonă continuă de memorie). Toate programele scrise în Octave sunt interpretate (nu compilate) și sunt executate linie cu linie. Octave pune la dispoziție o consolă pentru lansarea în execuție a comenzilor, însă se pot edita și fișiere sursă cu ajutorul editorului de text.

## 2. Programul Octave

La deschiderea programului Octave, va apărea pe ecranul calculatorului fereastra de comenzi, în care simbolul „>” reprezintă prompterul Octave care se află la începutul fiecărei linii de comandă. În dreptul acestui simbol se poate scrie comanda dorită, aceasta fiind rulată prin apăsarea tastei ENTER.

Exemplu (tastarea unei comenzi):

```
> x=3
```

```
x =
```

```
3
```

În urma acestei comenzi, în fereastra Workspace se poate observa apariția variabilei x, împreună cu o serie de informații corespunzătoare variabilei respective, informații precum: tipul de date, dimensiunea variabilei etc.

Dacă se va introduce semnul „;” după instrucțiune, în fereastra de comandă nu se va mai afișa comanda introdusă, însă variabila x va fi stocată în memoria Octave și poate fi vizualizată în fereastra Workspace.

Pe lângă fereastra de comandă, Octave pune la dispoziție și posibilitatea de utilizare a fișierelor sursă cu ajutorul editorului de text. Fișierul poate fi salvat sub format „.m”, care este, în esență, un fișier ASCII, deci poate fi modificat cu ajutorul oricărui instrument de modificare a textului.

Pentru crearea unui fișier script se poate utiliza comanda File → New.

În cadrul acestui laborator, se vor utiliza fișiere text.

### 3. Vectori și Matrice

Octave oferă o soluție simplă de a defini un vector, utilizând sintaxa:

inițializare:incrementare:terminare

Exemplu:

```
> vect = 1:2:9
```

```
vect =
```

```
1    3    5    7    9
```

Această linie de cod definește o variabilă vect, care reprezintă un vector ce conține valorile 1,3,5,7 și 9. Cu alte cuvinte, vectorul pornește de la valoarea inițială: 1, incrementează valoarea inițială la fiecare pas cu valoarea de incrementare: 2 și se oprește când ajunge (sau înainte de a depăși) valoarea finală: 9. În cazul în care nu este specificat pasul de incrementare, acesta este implicit 1.

Exemplu:

```
> vect = 1:5
```

```
vect =
```

```
1    2    3    4    5
```

Matricile pot fi definite prin separarea elementelor corespondente unei linii cu spațiu liber și terminarea unei linii (trecerea pe rândul următor) cu ajutorul simbolului ;.

```
> A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

```
A =
```

```
16    3    2   13
 5   10   11    8
 9    6    7   12
 4   15   14    1
```

Elementele unei matrici se definesc în paranteze drepte [], iar pentru accesarea elementelor matricii se folosesc paranteze rotunde: ().

Exemplu:

```
> A(2,3)
```

```
ans =
```

```
11
```

În cazul limbajului de programare Octave, indecșii sunt numerotați pornind de la 1, nu de la 0, ca în cazul altor limbaje precum C.

Limbajul Octave permite, de asemenea, și extragerea unei întregi coloane, sau linii din matrice.

Exemplu:

```
> A(:,3)
```

```
ans =
```

```
2
11
7
14
```

Utilizând această comandă, au fost extrase toate elementele din matricea A corespunzătoare coloanei 3.

```
> A(2,:)
```

```
ans =
```

```
5    10    11    8
```

Asemănător extragerii unei coloane, poate fi extrasă din matrice și o linie, conform exemplului anterior.

De asemenea, se poate crea și o submatrice cu elementele matricei A.

Exemplu:

```
> A(2:4,3:4)
```

```
ans =
```

```
11    8
7     12
14     1
```

Această comandă a extras din matricea A toate elementele corespunzătoare liniilor de la 2 la 4 (2,3 și 4) de pe coloanele 3 și 4.

Limbajul Octave permite crearea ușoară a matricei unitate, matricei de zero și unu, utilizând funcțiile: `eye`, `zeros` și `ones`.

Exemplu:

```
> eye(3)
```

```
ans =
```

```
1    0    0
0    1    0
0    0    1
```

```
> zeros(2,3)
```

```
ans =
```

```
    0    0    0
    0    0    0
```

```
> ones(3,2)
```

```
ans =
```

```
    1    1
    1    1
    1    1
```

Pentru a afla dimensiunea unei matrice, se utilizează funcția `size`.

Exemplu:

```
> size(A)
```

```
ans =
```

```
    4    4
```

Pentru transpunerea unei matrice se utilizează simbolul `'`.

Exemplu:

```
> A'
```

```
ans =
```

```
    16     5     9     4
     3    10     6    15
     2    11     7    14
    13     8    12     1
```

Funcția `sum` se poate utiliza pentru sumarea tuturor elementelor de pe linie sau de pe coloana unei matrice.

Exemplu:

```
> sum(A)
```

```
ans =
```

```
    34    34    34    34
```

În cazul în care nu se specifică dimensiunea pe care să se realizeze suma elementelor matricei, aceasta se va realiza implicit pe coloane (instrucțiunea fiind echivalentă cu instrucțiunea `sum(A,1)`).

Pentru calcularea sumei elementelor de pe fiecare linie, se va utiliza instrucțiunea `sum(A,2)`.

Pentru calcularea sumei tuturor elementelor din matrice se poate utiliza instrucțiunea: `sum(sum(A))`.

Funcțiile `min` și `max` pot fi utilizate în cadrul limbajului Octave pentru determinarea valorii minime, respectiv maxime, de pe liniile sau coloanele unei matrice.

Exemplu:

```
> min(A)

ans =

     4     3     2     1

> max(A)

ans =

    16    15    14    13

> min(A,[],2)

ans =

     2
     5
     6
     1

> max(A,[],2)

ans =

    16
    11
    12
    15
```

În cazul funcțiilor `min` și `max`, al doilea parametru de intrare nu este specific dimensiunii pe care se dorește calcularea minimului, motiv pentru care va fi înlocuit cu o variabilă goală: `[]`, urmând ca al treilea parametru de intrare să corespundă dimensiunii.

## 4. Instrucțiuni

### 4.1. Instrucțiunea if

Sintaxa utilizată în Octave pentru instrucțiunea if este următoarea:

```
if expresie
    instrucțiuni
end
```

Efect: dacă expresia este adevărată, atunci se execută instrucțiuni, altfel, dacă expresia este falsă, nu se execută nicio instrucțiune.

```
if expresie
    instrucțiuni_1
else
    instrucțiuni_2
end
```

Efect: dacă expresia este adevărată, se execută instrucțiuni\_1, altfel, se execută instrucțiuni\_2.

### 4.2. Switch

Sintaxa:

```
switch expresie
    case const_1
        instructiune_1
    case const_2
        instructiune_2
    case const_n
        instructiune_n
end
```

Efect: Rezultatul expresiei va conduce la una dintre constante: const\_1, const\_2, ...,const\_n, urmând să fie executată instrucțiunea corespunzătoare constantei rezultate în urma expresiei.

### 4.3. For

Sintaxă:

```
for index=val_init:pas:val_fin
    instrucțiuni
end
```

Efect: Pentru index pornind de la valoarea inițială (val\_init) până la valoarea finală (val\_fin) cu pasul pas se execută instrucțiuni

### 4.4. While

Sintaxă:

```
while expresie
    instrucțiuni
end
```

Efect: Cât timp expresia este adevărată, se execută instrucțiuni.

## 5. Funcții

Sintaxa:

```
function [lista_var_iesire] = nume_fct(lista_var_intrare)
    corp_functie
end
```

Unde:

- *nume\_fct* reprezintă numele funcției
- *lista\_var\_intrare* reprezintă lista variabilelor de intrare
- *lista\_var\_iesire* reprezintă lista variabilelor care vor fi transmise ca parametri de ieșire din funcție
- *corp\_functie* reprezintă instrucțiunile din interiorul funcției care vor fi executate în momentul apelului acesteia

O funcție în Octave este asemănătoare la nivel funcțional cu o funcție dintr-un alt limbaj de programare, precum C/C++: primește parametri de intrare, execută instrucțiuni, returnează rezultate.

Această funcție trebuie salvată într-un fișier cu **același** nume corespunzător funcției (*nume\_fct*).

Apelul funcției se realizează utilizând sintaxa: `[lista_var_iesire] = nume_fct(lista_var_intrare)`.

## 6. Operații

Limbajul Octave oferă posibilitatea de a realiza cu ușurință operații atât asupra vectorilor cât și asupra matricelor.

Astfel că, limbajul Octave permite ca operatorul + și - să adune sau să scadă două matrice element cu element.

Exemplu:

```
> A = [1 2 1; 0 1 1; 2 1 1];
> B = [0 0 1; 1 1 0; 1 1 2];
> A+B
```

ans =

1	2	2
1	2	1
3	2	3

```
> A-B
```

ans =

1	2	0
-1	0	1
1	0	-1



Utilizând operatorul `*`, acesta va realiza înmulțirea matriceală dintre A și B, în timp ce operatorul `.*` va înmulți element cu element cele două matrice.

Exemplu:

```
>> A*B
```

```
ans =
```

3	3	3
2	2	2
2	2	4

```
>> A.*B
```

```
ans =
```

0	0	1
0	1	0
2	1	2

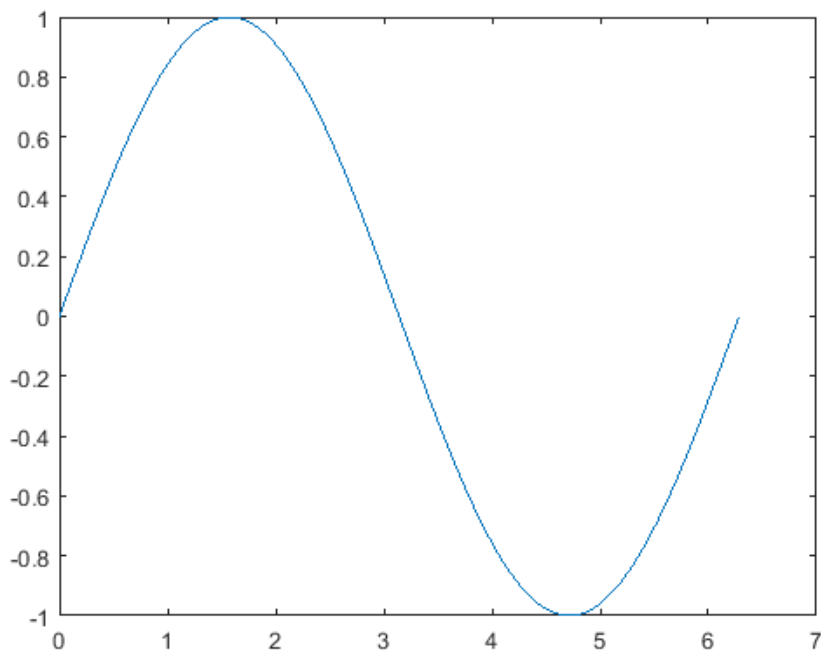
## 7. Reprezentarea grafică în Octave

Pentru reprezentarea grafică a unui semnal unidimensional se poate utiliza funcția plot.

Exemplu:

```
> x = 0:pi/100:2*pi;  
y = sin(x);  
plot(x,y)
```

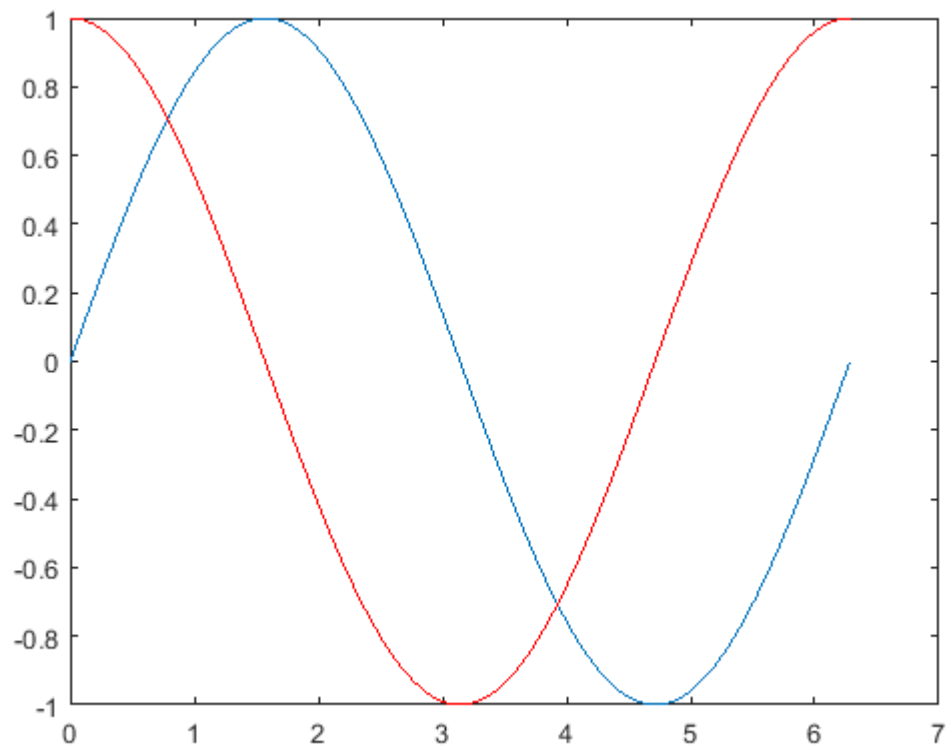
Programul va afișa următoarea imagine:



În cazul în care se dorește suprapunerea a două grafice, se poate utiliza instrucțiunea hold on.

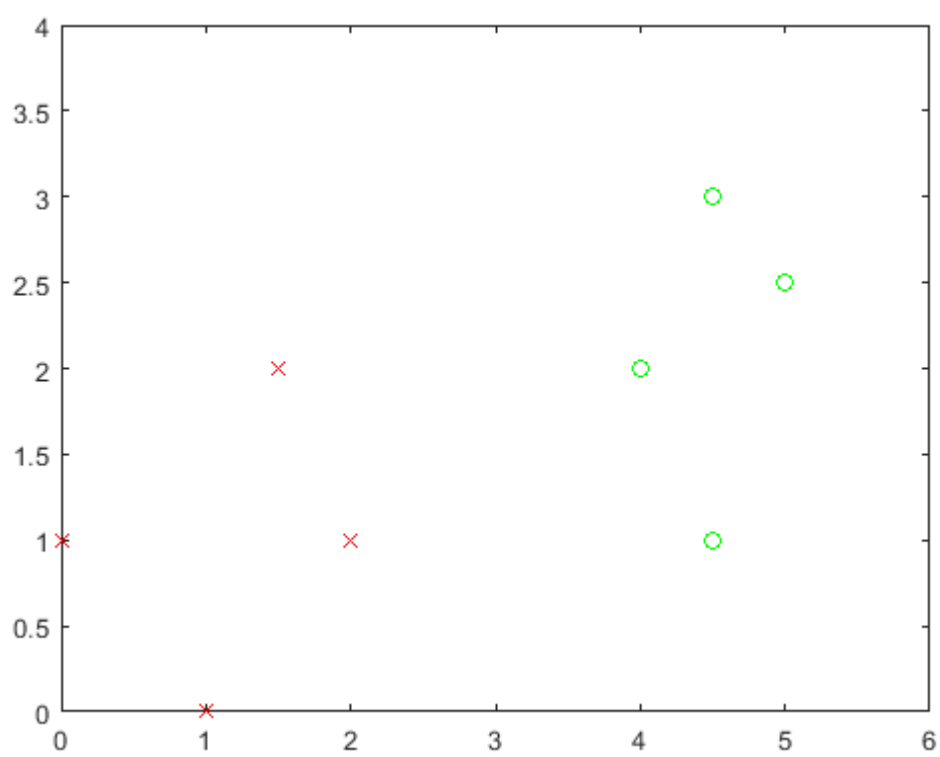
Exemplu:

```
> x = 0:pi/100:2*pi;  
y = sin(x);  
plot(x,y)  
hold on, plot(x, cos(x), 'r')
```



Exemplu:

```
> x1=[0 1 1.5 2];  
> y1=[1 0 2 1];  
> x2 = [4 4.5 4.5 5];  
> y2 = [2 1 3 2.5];  
> plot(x1,y1,'xr')  
> hold on, plot(x2,y2,'og')  
> axis([0 6 0 4])
```



## 8. Exerciții

1. Se dau vectorii:

$$x = [5 \ 7 \ 3 \ 2 \ 1 \ 6];$$

$$y = [7 \ 6 \ 12 \ 9 \ 8 \ 10];$$

Să se calculeze media aritmetică a celor doi vectori precum și variația și covariația acestora.

Formula varianței este:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_x)^2$$

Unde:

- $N$  reprezintă numărul de elemente ale vectorului  $x$
- $\mu_x$  reprezintă media vectorului  $x$

Rezultatul obținut în urma calculării varianței pentru  $x$  este: 4.66.

Rezultatul obținut în urma calculării varianței pentru  $y$  este: 3.88.

Formula covarianței dintre doi vectori este:

$$\sigma_{x,y}^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$$

Rezultatul obținut în urma calculării covariației dintre vectori este: -1.5.

2. Se dau vectorii:

$$x = [2 \ 3 \ 0 \ 9 \ 7 \ 8];$$

$$w = [0.2 \ 1.5 \ -0.1 \ -0.7 \ 0.7 \ 0.02];$$

Să se obțină valoarea lui  $R$ , aplicând formula:

$$R = \sum_{i=1}^N x_i w_i$$

Rezultatul obținut:  $R = 3.66$

3. Se dă matricea  $M$ :

$$M = \begin{bmatrix} 4.5 & 1 & 1 \\ 1.5 & 2 & 0 \\ 4. & 3 & 1 \\ 5 & 2.5 & 1 \\ 4 & 2 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 2 & 1 & 0 \end{bmatrix};$$

Unde:

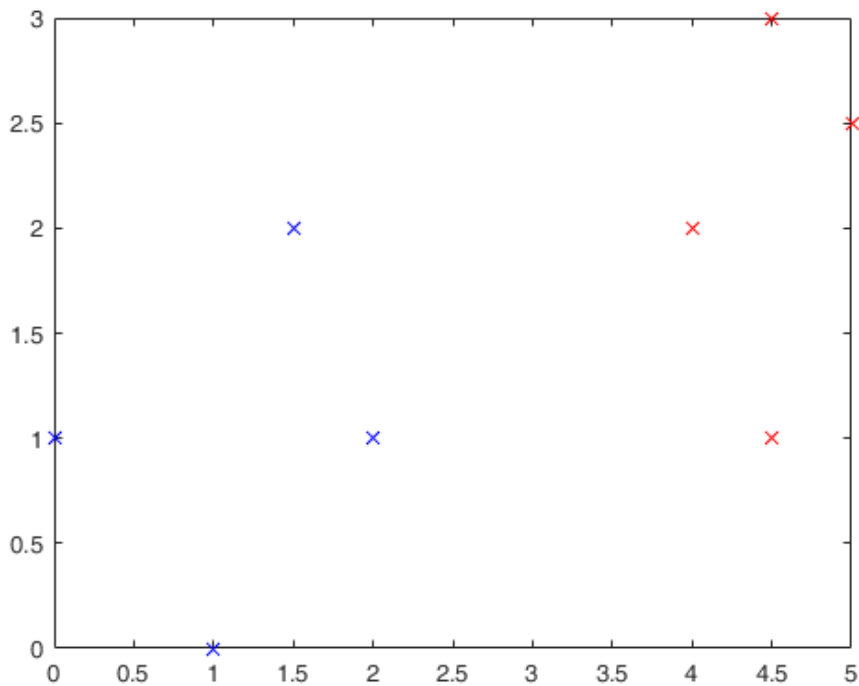
- prima coloană reprezintă punctele de coordonate  $x$

- a doua coloană reprezintă punctele de coordonate y
- iar a treia coloană reprezintă o etichetă a fiecărui punct de coordonate [x;y]

Exemplu: Primul punct (corespondent primei linii din matrice) prezintă coordonatele [4.5;1] și are eticheta 1.

Se dorește reprezentarea pe un grafic a tuturor punctelor din matrice, astfel încât punctele ce prezintă eticheta 0 să fie reprezentate cu albastru, iar cele ce prezintă eticheta 1 să fie reprezentate cu roșu.

Rezultat final:



4. Să se realizeze o funcție care permută două linii i și j (parametrii de intrare) dintr-o matrice.
5. Să se realizeze o funcție care realizează histograma unui vector și o afișează.