**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Programul de studii: Tehnologia informației**



**RAPORT**

**Disciplina „Procesarea semnalelor”**

**Tema: Introducerea in MATLAB**

|  |  |  |
| --- | --- | --- |
| **Student(ă):** | **\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Vlași**ț**chi Ștefan , TI-216** |
|  |  |  |
| **Coordonator universitate:** | **\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Cazac Artiom, asist.univ.** |

**Chișinău, 2024**

**Obiective**: însușirea elementelor de bază din sistemul MATLAB precum: definirea unei matrice, operații cu matrice, tipuri de date, variabile, instrucțiuni, fișiere script, reprezentarea graficelor și implementarea funcțiilor în Matlab.

**Sarcini practice:**

**Ex1.1**

Fie vectorii linie a = [0, 0.1, 0.2, … , 2] și coloană

a) Ce lungime trebuie să aibă b astfel, încât să aibă sens înmulțirea (în sens matricial) a\*b? Inițializați în MATLAB cei doi vectori și efectuați înmulțirea. -

b) Efectuați înmulțirea b\*a.



**Figura 1.1 – Inițializarea variabilelor**

**Figura 1.2 – Rezultatele obținute**

a = [0:0.1:2];

b = ones(21,1);

c = a\*b;

**Ex 1.2**.

Se creează un fișier nou care trebuie salvat în directorul d:/student/pns/nrgrupa. Folosind sintaxele și indicațiile din secțiunile 1.2.6 și 1.2.7, elaborați un program MATLAB care să genereze un vector cu elemente aliatoare cu distribuție normală (gaussiană) și să afișeze elementele negative ale acestui vector.

% Generare vector cu elemente aleatoare cu distribu?ie normal?

n = 100; % Num?rul de elemente ?n vector

mu = 0; % Media distribu?iei normale

sigma = 1; % Devia?ia standard a distribu?iei normale

% Generarea vectorului cu distribu?ie normal?

random\_vector = mu + sigma \* randn(n, 1);

% Afi?area elementelor negative ale vectorului

negative\_elements = random\_vector(random\_vector < 0);

disp('Elemente negative ale vectorului:');

disp(negative\_elements);

% Verifica?i dac? directorul exist? ?i, dac? nu, crea?i-l

folder\_path = 'd:/student/pns/nrgrupa/';

if ~exist(folder\_path, 'dir')

mkdir(folder\_path);

disp(['Directorul ' folder\_path ' a fost creat.']);

end

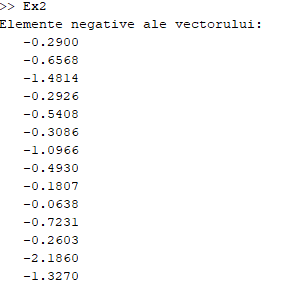
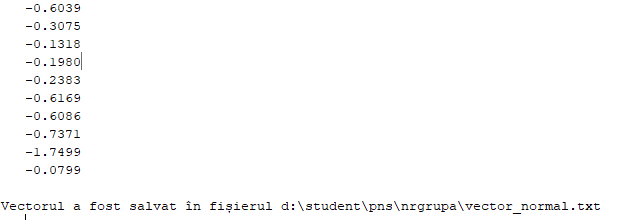
% Salva?i vectorul ?n fi?ier

file\_name = 'vector\_normal.txt';

full\_file\_path = fullfile(folder\_path, file\_name);

dlmwrite(full\_file\_path, random\_vector, 'precision', '%.6f');

disp(['Vectorul a fost salvat ?n fi?ierul ' full\_file\_path]);



**Figura 2.1 – Rezultatele obținute**

**Ex 1.3. Exercițiu:**

Elaborați un program MATLAB care generează un vector cu elemente complexe. Elaborați (un alt fișier) o funcție MATLAB care, având drept parametru de intrare vectorul cu valori complexe, returnează ca parametri de ieșire:

- media aritmetică a pârților reale ale elementelor vectorului;

- un vector ce conține elementele vectorului inițial ridicate la pătrat;

% Generare vector cu elemente complexe

n = 5; % Num?rul de elemente ?n vector

complex\_vector = randn(n, 1) + 1i \* randn(n, 1); % Elemente complexe aleatoare

% Apelarea func?iei ?i afi?area rezultatelor

[media\_real, vector\_patrat, rezultat\_inmultire] = functie\_operatii(complex\_vector);

disp('Media aritmetic? a p?r?ilor reale ale elementelor:');

disp(media\_real);

disp('Vectorul cu elementele ini?iale ridicate la p?trat:');

disp(vector\_patrat);

disp('Matricea ob?inut? din ?nmul?irea vectorului ini?ial cu transpusul s?u:');

disp(rezultat\_inmultire);

function [media\_real, vector\_patrat, rezultat\_inmultire] = functie\_operatii(complex\_vector)

% Calculeaz? media aritmetic? a p?r?ilor reale ale elementelor vectorului

media\_real = mean(real(complex\_vector));

% Ridic? la p?trat elementele vectorului ini?ial

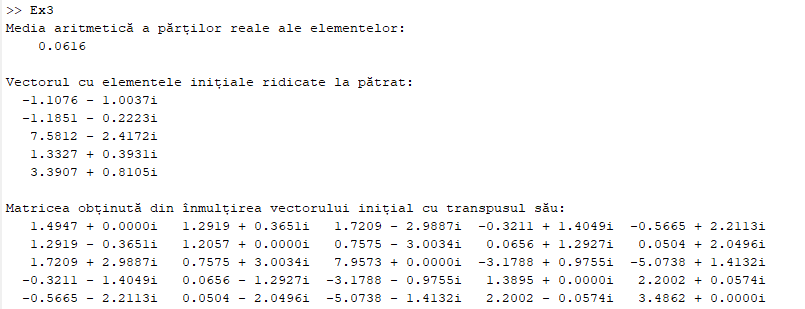
vector\_patrat = complex\_vector .^ 2;

% Calculeaz? ?nmul?irea vectorului ini?ial cu transpusul s?u

rezultat\_inmultire = complex\_vector \* complex\_vector';

% Returneaz? rezultatele ca parametri de ie?ire

end



- o matrice obținută din înmulțirea vectorului inițial cu transpusul său. Atenție: pentru a nu se afișa rezultate intermediare din funcție sau elementele unor variabile se va folosi ; la sfârșitul liniei respective de program.

**Figura 3.1 – Rezultatele obținute**

**Ex 1.4. Exercițiu**

Elaborați un program MATLAB în care să generați și să reprezentați grafic folosind funcția stem următorii vectori:

1. z = [0,0,0,0,0,1,0,0,...,0], vectorul z, având lungimea 21. Reprezentarea grafică se va face în două „mini ferestre” (funcția sublot), vectorul z în funcție de n=0:20, respectiv de m=-5:15;
2. t=|10-n |, reprezentat grafic în funcție de n=0:20;
3. 𝑥1 = 𝑠𝑖𝑛 ( 𝜋 17 𝑛), -15≤n≤25 și 𝑥2 = 𝑐𝑜𝑠 ( 𝜋 √23 𝑛), 0≤n≤50.

Cele două secvențe vor fi reprezentate:

în figura 1 – în același sistem de coordonate (pe același grafic);

în figura 2 – folosind două “mini ferestre” grafice plasate una sub alta.

Reprezentați cele două figuri, folosind comanda plot, apoi încercați reprezentarea lor, folosind comanda stem. Cu funcția plot se pot reprezenta grafic semnale sau funcții “continue”, deoarece se unesc cu linie continuă valorile care se reprezintă. Astfel, se pot reprezenta semnale continue, alegând variabila timp cu pasul mai mic decât variația semnalului reprezentat. De exemplu, dacă perioada semnalului e 0.01 secunde se poate alege variabila temporală cu pasul de 0.001s: t = 0:0.001:5 (secunde).

% a)

z = [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];

% Cre?m prima figur? cu subplot

subplot(2, 1, 1)

stem(0:20,z) % asigur?m c? z are aceea?i lungime cu n

title('Vectorul z ?n func?ie de n=0:20')

% Cre?m a doua figur? cu subplot

subplot(2, 1, 2)

stem((-5:15), z) % asigur?m c? z ?ncepe de la indicele corespunz?tor din m

title('Vectorul z ?n func?ie de m=-5:15')

% b)

% Definim vectorul t

t = abs(10-(0:20));

% Reprezent?m grafic vectorul t

stem(0:20,t)

title('Vectorul t in functie de n=0:20')

%c) % Definirea variabilei temporale pentru x1

n1 = -15:25;

x1 = sin(pi/17 \* n1);

% Definirea variabilei temporale pentru x2

n2 = 0:50;

x2 = cos(pi/sqrt(23) \* n2);

% Reprezentarea celor dou? secven?e ?n aceea?i figur? folosind plot

figure;

plot(n1, x1, 'r', n2, x2, 'b');

title('Reprezentarea celor dou? secven?e ?n acela?i sistem de coordonate');

xlabel('n');

ylabel('Amplitudine');

legend('x1 = sin(pi/17 \* n)', 'x2 = cos(pi/sqrt(23) \* n)');

% Reprezentarea celor dou? secven?e ?n dou? miniferestre folosind subplot ?i plot

figure;

subplot(2, 1, 1);

plot(n1, x1, 'r');

title('Reprezentarea secven?ei x1 = sin(pi/17 \* n)');

xlabel('n');

ylabel('Amplitudine');

subplot(2, 1, 2);

plot(n2, x2, 'b');

title('Reprezentarea secven?ei x2 = cos(pi/sqrt(23) \* n)');

xlabel('n');

ylabel('Amplitudine');

% Reprezentarea celor dou? secven?e ?n aceea?i figur? folosind stem

figure;

stem(n1, x1, 'r', 'filled');

hold on;

stem(n2, x2, 'b', 'filled');

hold off;

title('Reprezentarea celor dou? secven?e folosind stem');

xlabel('n');

ylabel('Amplitudine');

legend('x1 = sin(pi/17 \* n)', 'x2 = cos(pi/sqrt(23) \* n)');

% Reprezentarea celor dou? secven?e ?n dou? miniferestre folosind subplot ?i stem

figure;

subplot(2, 1, 1);

stem(n1, x1, 'r', 'filled');

title('Reprezentarea secven?ei x1 = sin(pi/17 \* n) folosind stem');

xlabel('n');

ylabel('Amplitudine');

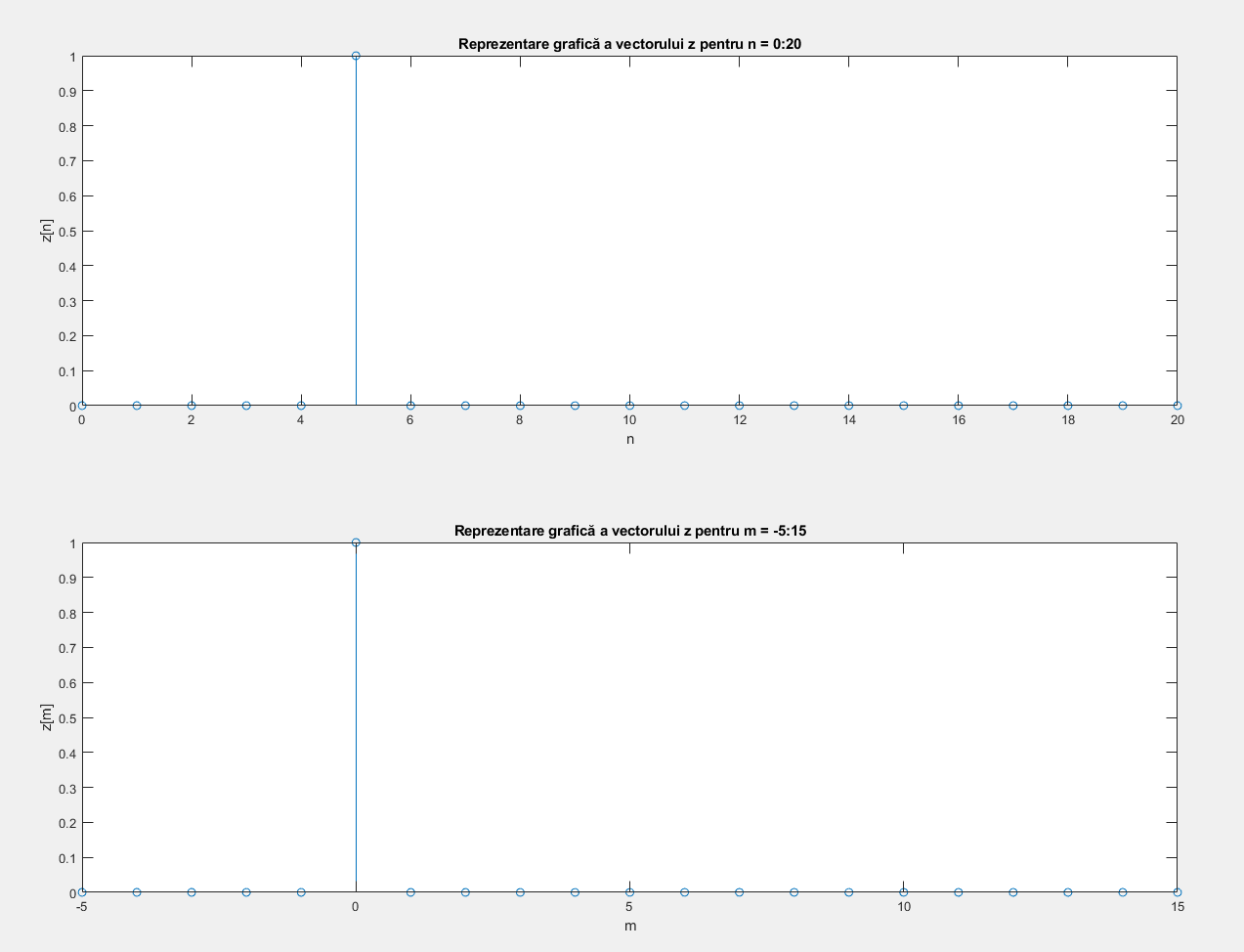
subplot(2, 1, 2);

stem(n2, x2, 'b', 'filled');

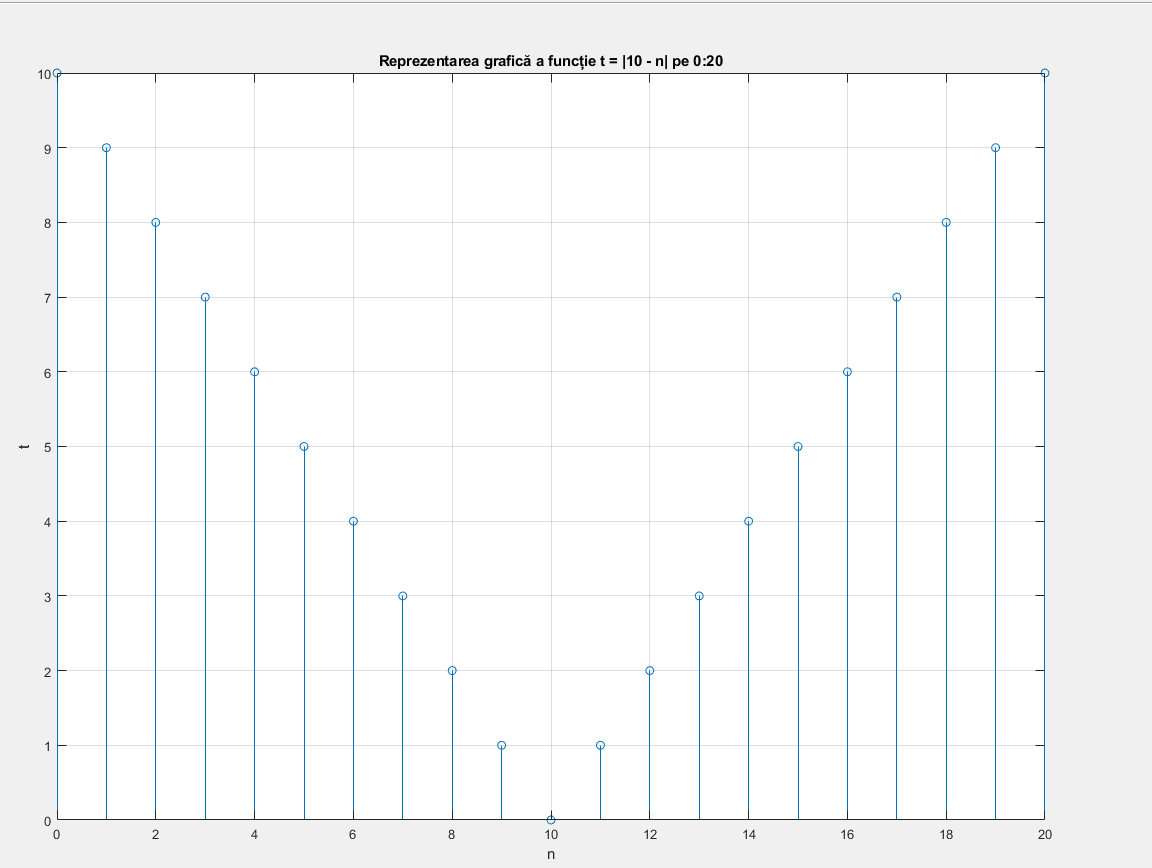
title('Reprezentarea secven?ei x2 = cos(pi/sqrt(23) \* n) folosind stem');

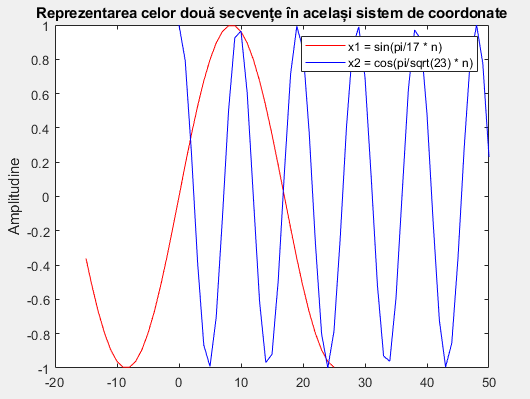
xlabel('n');

ylabel('Amplitudine');

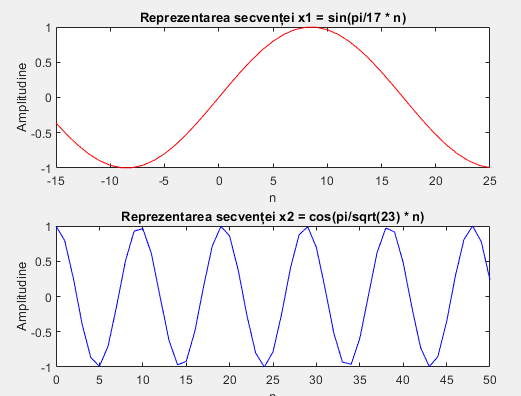


**F**i**gura 4.1 – Rezultatele obținute (a)**

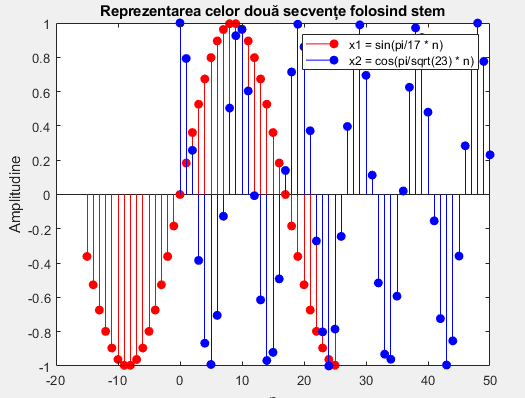


****

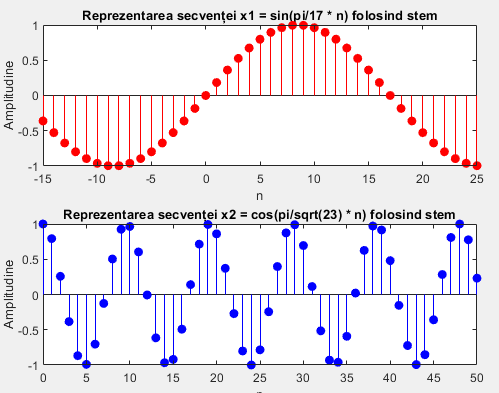
**F**i**gura 4.2 – Rezultatele obținute (c)**

****

**Figura 4.3 – Rezultatele obținute (c)**

****

**Figura 4.4 – Rezultatele obținute (c)**

****

**Figura 4.5 – Rezultatele obținute (c)**

**Ex 1.5. Exercițiu:**

1. Să se reprezinte grafic cu funcția plot un semnal sinusoidal de frecvență 50 Hz, de durată 0.2 secunde și amplitudine 2. Se va alege rezoluția temporală 1ms.

F = 50;

t = 0:0.001:0.2;

s = 2\*sin(2\*pi\*F\*t);

plot(t, s, '.-'), xlabel('Timp [s]'), grid

Modificați pasul de variație a variabilei t la 0.01, apoi la 0.0002. Comentați diferențele;

F = 50;

Rez\_temp0 = 0.001;

Rez\_temp1 = 0.01;

Rez\_temp2 = 0.0002;

% Definirea variabilei timp

t0 = 0:Rez\_temp0:0.2;

t1 = 0:Rez\_temp1:0.2;

t2 = 0:Rez\_temp2:0.2;

% Calculul semnalului sinusoidal

s0 = 2\*sin(2\*pi\*F\*t0);

s1 = 2\*sin(2\*pi\*F\*t1);

s2 = 2\*sin(2\*pi\*F\*t2);

%Reprezentarea grafică a semnalului sinusoidal cu rezoluția temporală 0.001 s

figure;

plot(t0, s0, '.-');

title(['Reprezentarea semnalului sinusoidal (F = ', num2str(F), ' Hz, R = ', num2str(Rez\_temp0), ' s)']);

xlabel('Timp [s]');

grid;

%Semnalul sinusoidal cu rezoluția temporală 0.01 s

figure;

plot(t1, s1, '.-');

title(['Reprezentarea semnalului sinusoidal (F = ', num2str(F), ' Hz, R = ', num2str(Rez\_temp1), ' s)']);

xlabel('Timp [s]');

grid;

%Semnalul sinusoidal cu rezoluția temporală 0.0002 s

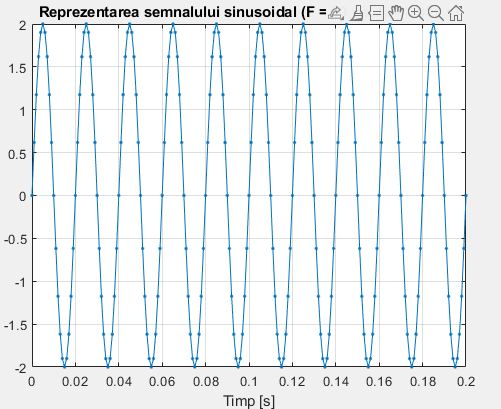
figure;

plot(t2, s2, '.-');

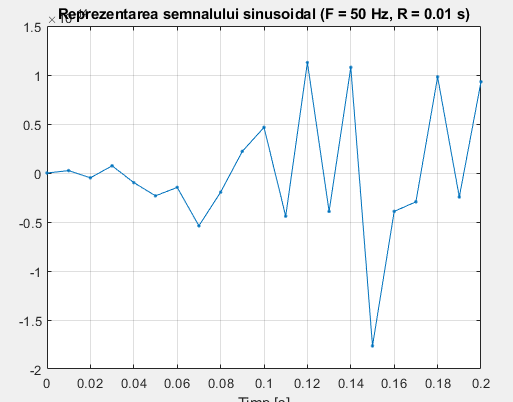
title(['Reprezentarea semnalului sinusoidal (F = ', num2str(F), ' Hz, R = ', num2str(Rez\_temp2), ' s)']);

xlabel('Timp [s]');

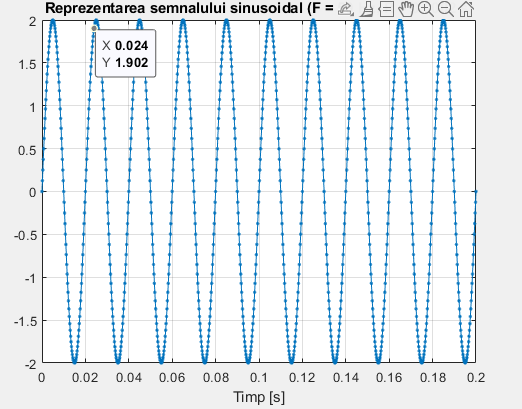
grid;



**Figura 5.1– Rezultatele obținute**



**Figura 5.2– Rezultatele grafice obținute**



**Figura 5.3– Rezultatele grafice obținute**

**Ex 1.6. Exercițiu:**

Să se genereze vectorul x conținând valorile 1, 2, 3, ..., 99, 100 și vectorul y având valorile 2, 4, 6, 8, ..., 198, 200:

1. să se reprezinte grafic y în funcție de x, folosind funcția stem;
2. să se reprezinte grafic y în funcție de x, folosind funcția plot;
3. să se reprezinte grafic în aceeași figură, în același sistem de coordonate, y în funcție de x, folosind funcțiile plot și stem (se vor folosi culori diferite);
4. să se reprezinte grafic în aceeași figură, în sisteme de coordonate diferite, y în funcție de x, folosind funcția plot și y în funcție de x, folosind funcția stem.

% Generarea vectorilor x ?i y

x = 1:100;

y = 2:2:200;

% a) Reprezentarea grafic? a lui y ?n func?ie de x folosind func?ia stem

figure;

stem(x, y);

title('a) Reprezentarea grafic? a lui y ?n func?ie de x folosind stem');

xlabel('x');

ylabel('y');

% b) Reprezentarea grafic? a lui y ?n func?ie de x folosind func?ia plot

figure;

plot(x, y);

title('b) Reprezentarea grafic? a lui y ?n func?ie de x folosind plot');

xlabel('x');

ylabel('y');

% c) Reprezentarea grafic? a lui y ?i x folosind plot ?i stem ?n aceea?i figur?

figure;

plot(x, y, 'r'); % y ?n func?ie de x folosit plot

hold on;

stem(x, y, 'filled'); % y ?n func?ie de x folosit stem

title('c) Reprezentarea grafic? a lui y ?i x folosind plot ?i stem');

xlabel('x');

ylabel('y');

legend('plot', 'stem');

hold off;

% d) Reprezentarea grafic? a lui y ?i x folosind plot ?i stem ?n sisteme de coordonate diferite

figure;

subplot(2, 1, 1);

plot(x, y, 'r');

title('d) Reprezentarea grafic? a lui y ?n func?ie de x folosind plot');

xlabel('x');

ylabel('y');

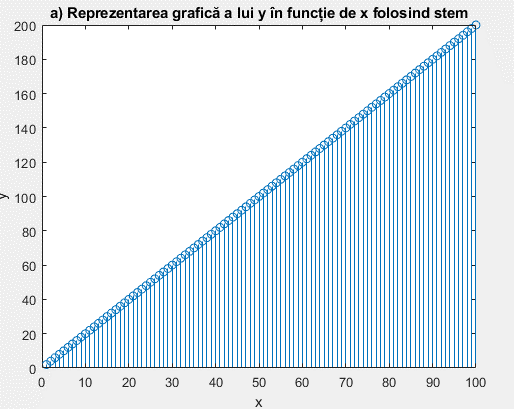
subplot(2, 1, 2);

stem(x, y, 'filled');

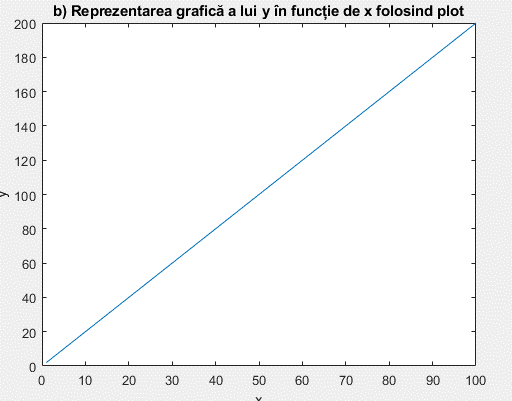
title('d) Reprezentarea grafic? a lui y ?n func?ie de x folosind stem');

xlabel('x');

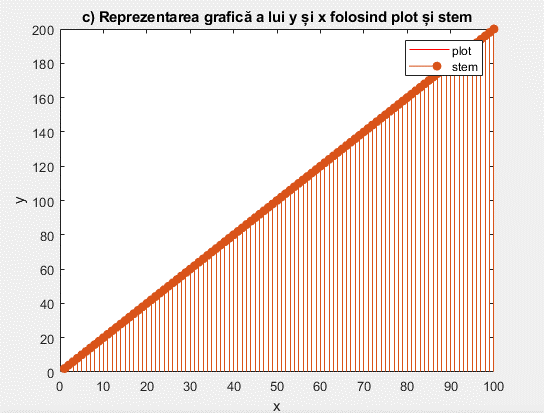
ylabel('y');



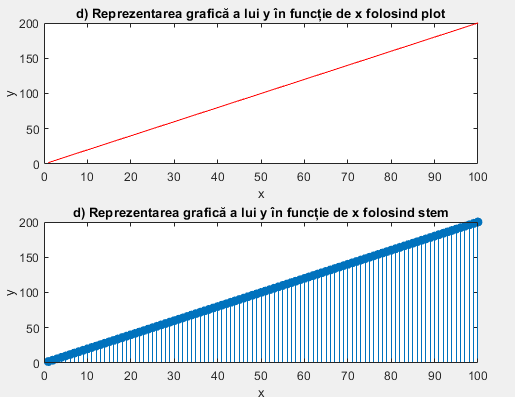
**Figura 6.1– Rezultatele grafice obținute**



**Figura 6.2– Rezultatele grafice obținute**



**Figura 6.3– Rezultatele grafice obținute**



**Figura 6.4– Rezultatele grafice obținute**

**Ex 1.7. Exercițiu:**

Fie f o funcție reală de o variabilă reală, precizată de următorii vectori:

x=[-2 -1.5 0 0.5 2.5 3];

y=[6 9 11 10 7 5].

Să se aproximeze derivata de ordinul I a funcției f în punctele:

a) -2, 0 și 2.5, folosind diferențe progresive;

b) -1.5, 0 și 3, folosind diferențe regresive;

c) -0.75, 0.25, 0.5 și 1, folosind diferențe centrate. Să se reprezinte grafic în aceeași fereastră grafică derivata de ordinul I a funcției f, obținută prin aproximare cu cele trei tipuri de diferențe finite.

Fie funcția polinomială f : R → R, f(x)=x4 +13⋅x3 -7⋅x2 +x-1.

a) Să se calculeze derivata funcției f pentru valorile -2, -1.3, 0.1 și 2.45.

b) Să se reprezinte grafic derivata funcției f pe intervalul [-2,3].

Să se calculeze , unde funcția f este dată prin relațiile:

, 𝑥j = −1.1 + 0.1 ∙ 𝑗, 𝑗 = 1,2, … ,11.

Să se calculeze integrala .

% a) Aproximarea derivatei de ordinul I folosind diferen?e finite progresive

x = [-2 -1.5 0 0.5 2.5 3];

y = [6 9 11 10 7 5];

df = diff(y) ./ diff(x); % Calculul aproxim?rii derivatei

% Afisarea derivatei ?n punctele dorite

vx1 = [-2 0 2.5];

for i = 1:length(vx1)

% Determinarea pozi?iei pe care se afl? vx(i) ?n x

k = find(x == vx1(i));

% Afisarea derivatei cu diferen?e progresive

disp(['f''(' num2str(vx1(i)) ') = ' num2str(df(k))])

end

% b) Aproximarea derivatei de ordinul I folosind diferen?e finite regresive

vx2 = [-1.5 0 3];

for i = 1:length(vx2)

% Determinarea pozi?iei pe care se afl? vx(i) ?n x

k = find(x == vx2(i));

% Afisarea derivatei cu diferen?e regresive

disp(['f''(' num2str(vx2(i)) ') = ' num2str(df(k-1))])

end

% c) Aproximarea derivatei de ordinul I folosind diferen?e finite centrale

vx3 = [-0.75 0.25 0.5 1];

n = length(x);

der = spline(x(1:n-1), df, vx3); % Interpolare pentru a g?si derivatele ?n punctele dorite

for i = 1:length(der)

disp(['f''(' num2str(vx3(i)) ') = ' num2str(der(i))])

end

% d) Calculul derivatei func?iei polinomiale ?i reprezentarea grafic?

c = [1 13 -7 1];

dc = polyder(c);

vx4 = [-2 -1.3 0.1 2.45];

der = polyval(dc, vx4); % Calculul derivatei pentru valorile date

disp('Derivatele functiei polinomiale:');

disp(der);

% Reprezentarea grafic? a derivatei pe intervalul [-2, 3]

x\_plot = -2:0.1:3;

df\_plot = polyval(dc, x\_plot);

plot(x\_plot, df\_plot);

title('Reprezentarea grafica a derivatei pe [-2,3]');

% Calculul integralei definite

for j = 1:11

x(j) = -1.1 + 0.1 \* j;

y(j) = j \* x(j)^2 / (x(j) - 1) - 2 / (j + 1);

end

integrala\_definita = trapz(x, y);

disp('Integrala definita a functiei f(x):');

disp(integrala\_definita);

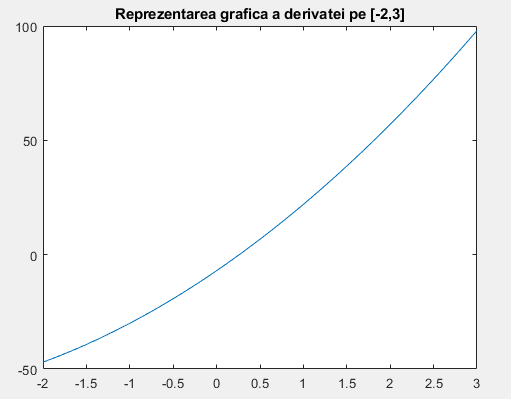
% Calculul integralei definite pentru functia data

f = @(x) 1 ./ (sin(x) + cos(x));

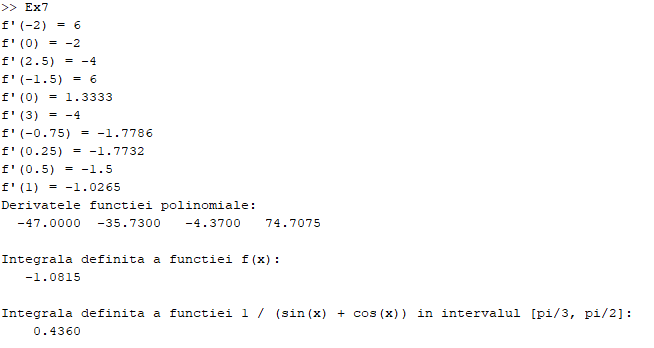
integrala\_definita2 = integral(f, pi/3, pi/2);

disp('Integrala definita a functiei 1 / (sin(x) + cos(x)) in intervalul [pi/3, pi/2]:');

disp(integrala\_definita2);



**Figura 7.1– Rezultatele grafice obținute**



**Figura 7.2– Rezultatele obținute**

**Concluzie:**

În cursul elaborării acestei lucrări de laborator, am avut oportunitatea de a explora funcționalitățile programului MATLAB și de a rezolva diverse sarcini care acoperă un spectru larg de concepte și funcții ale acestui program. Am lucrat cu vectori, funcții de generare a numerelor aleatoare, operații matematice cu numere complexe și reprezentarea grafică a semnalelor.

Exercițiile date au contribuit la dezvoltarea cunoștințelor de bază în utilizarea MATLAB și la înțelegerea diferitelor funcții și metode disponibile în program. Prin rezolvarea acestor sarcini, am obținut o mai bună familiarizare cu mediul de lucru MATLAB și am avansat în abilitățile noastre de analiză și rezolvare a problemelor în domeniul științific și tehnic.