

# Reprezentări grafice

## Obiectivele lucrării de laborator:

- Descrierea funcțiilor trigonometrice predefinite
- Descrierea modului de reprezentare grafică în Matlab
- Reprezentarea grafică a unor funcții în sistem de coordonare X, Y

## 3.1 Funcții de reprezentare grafică.

Programul Matlab permite pe lângă manipularea vectorilor sau a matricilor, ori a operațiilor ce se pot realiza cu ajutorul acestora, reprezentarea soluțiilor ecuațiilor rezolvate într-un sistem de coordonate 2D, X, Y, ori într-un sistem de coordonate 3D, X, Y, Z, cu ajutorul unor funcții predefinite ce vor fi prezentate în cadrul acestei lucrări.

În general, în domeniul nostru de inginerie ne interesează răspunsul anumitor circuite la diverși stimuli. Acest răspuns poate fi urmărit printr-o reprezentare grafică a ecuațiilor matematice ce pot fi deduse din funcționarea sistemului. O reprezentare corectă a soluțiilor duce la o interpretare bună a rezultatelor și astfel se pot valida sau invalida rezultatele obținute. Tabelul 3.1 prezintă câteva funcții trigonometrice predefinite în Matlab cu ajutorul cărora vom încerca să înțelegem sistemul de reprezentare grafică a programului.

Funcția	Explicația funcției
<b>sin / asin</b>	Calculează sinusul / arcsinusul
<b>cos / acos</b>	Calculează cosinusul / arccosinusul
<b>tan / atan / atan2</b>	Calculează tangenta / arctangenta / arctangenta - argument complex
<b>cot / acot</b>	Calculează cotangenta / arccotangenta
<b>sinh / asinh</b>	Calculează sinus hiperbolic / arcsinus hiperbolic
<b>cosh / acosh</b>	Calculează cosinusul hiperbolic / arccosinusul hiperbolic
<b>tanh / atanh</b>	Calculează tangenta hiperbolică / arctangenta hiperbolică
<b>coth / acoth</b>	Calculează cotangenta hiperbolică / arccotangenta hiperbolică

Tabel 3.1. Funcții uzuale trigonometrice predefinite în Matlab

Cea mai simplă metodă de afișare și interpretare a rezultatelor este folosirea funcției **imread/image** prezentată în prima lucrare de laborator. Astfel, cu ajutorul acestei funcții se poate importa o imagine din afara programului, imagine ce poate fi analizată sau compartă cu o reprezentare în Matlab. Sintaxa funcției **imread/image** poate fi de forma:

Exemplu:

```
x=imread('poza.jpg');
image(x);
grid on;
```

### 3.1.2 Reprezentare 2D

În mare parte reprezentarea grafică a unui sistem se realizează într-un sistem cartezian de 2 axe X, Y. Pentru acest tip de reprezentare, programul Matlab pune la dispoziția utilizatorilor câteva funcții predefinite, funcții de sunt prezentate în tabelul 3.2.

Funcția	Proprietatea
<b>ezplot</b>	Plotare rapida
<b>plot</b>	Plotare în funcție de doi vectori X,Y
<b>plotyy</b>	Plotare cu două axe Y
<b>loglog</b>	Plotare logaritmică pe X și pe Y
<b>semilogx</b>	Plotare logaritmică pe X
<b>semilogy</b>	Plotare logaritmică pe Y
<b>bar</b>	Afișează un grafic segmental (diagrame bară)
<b>barh</b>	Afișează un grafic segmental (diagrame bară)
<b>histogram</b>	Afișează o histogramă a unui grafic segmental
<b>polar</b>	

Tabel 3.2. Funcții predefinite de reprezentare 2D.

Exemplu:

Să se reprezinte grafic funcția:

$$\frac{x^2}{4} + \frac{y^2}{5} = 15$$

O astfel de funcție poate fi reprezentată grafic utilizând funcția **ezplot**. Această funcție este o funcție de plotare (reprezentare grafică) rapidă. Sintaxa funcției este simplă de forma: **ezplot('funcția matematica',[intervalul de calcul])**.

Pentru exemplul anterior, programul Matlab se poate scrie de forma:

```
ezplot('x.^2/4+y.^2/5=15',[-15 15]);
grid on;
```

Rezultatul instrucțiunilor Matlab este prezentat în figura 3.1. Așa cum se poate observa programul Matlab returnează sub formă grafică soluțiile ecuației pe un interval definit.

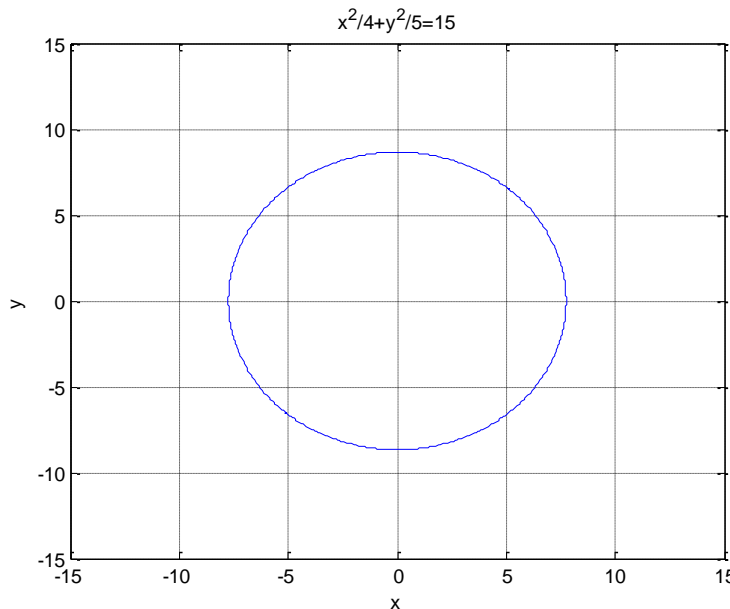


Figura 3.1. Reprezentare grafică utilizând funcția ezplot pentru  $\frac{x^2}{4} + \frac{y^2}{5} = 15$

Cea mai uzuală funcție de reprezentare grafică este funcția **plot**. Spre deosebire de funcția ezplot această funcție permite afișarea rezultatelor în funcție de 2 variabile (vectori sau matrice).

Exemplu:

Să se reprezinte grafic funcția:

$$y = x^3, \text{ pentru } x \text{ cu valori între } [-5,4]$$

```
x=-5:0.1:4; %x este o valoare din intervalul [-5,4] cu increment de 0.1
y = x.^3; %y este x^3
plot(y); %afisam grafic y
```

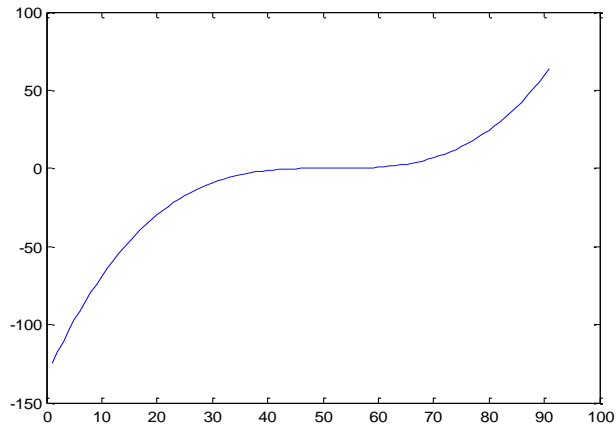


Figura 3.2. Reprezentarea grafică a funcției  $y = x^3$

Figura 3.2 ilustrează reprezentarea grafică a funcției  $y = x^3$  folosind funcția `plot`. Vizualizând figura se observă că pe axa  $x$  suntem într-un interval de la 1 la 91. Asta deoarece vectorul  $y$  are 91 de valori pentru valorile lui  $x$  în intervalul ales de noi. (verificare se poate face verificând lungimea vectorului  $y$  - `length(y)`). Așadar, reprezentarea nu este întocmai corectă, deoarece reprezentăm grafic valorile vectorului  $y$  în funcție de poziția lor în vector. O reprezentare corectă s-ar putea realiza dacă am folosi ambi vectori pentru reprezentare așa cum se arată în exemplu Matlab următor.

```
x=-5:0.1:4;    %x ia o valoare in intervalul [-5,4] cu increment de 0.1
y = x.^3;     %y este x^3
plot(x,y);    %afisam grafic y
grid on;      %plasam caroiaj
```

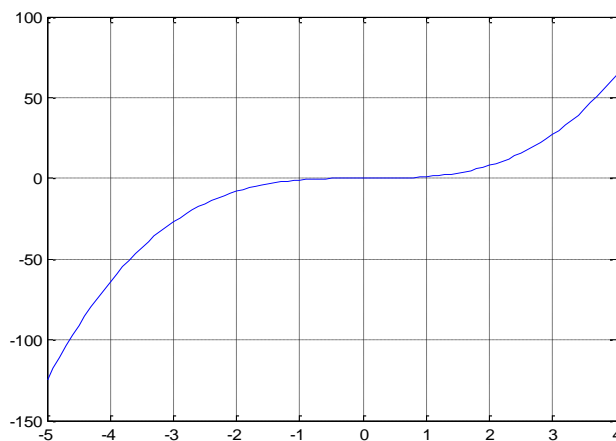


Figura 3.3. Reprezentare grafică a funcției  $y = x^3$ , în funcție de  $x$

Privind figura 3.3 se observă că pe axa x avem acum, exact intervalul pe care l-am stabilit inițial pentru vectorul x. Curbele grafice sunt identice în cele două figuri, doar că, în reprezentarea din figura 3.3 se observă clar dependența funcției y la variația variabilei x. Adăugarea comenzi **grid on**; implică prezența caroiajului asupra graficului astfel încât citirea valorilor de pe grafic să se realizeze mult mai ușor.

De asemenea un grafic realizat este sugestiv și ușor interpretabil dacă are un titlu, are specificațiile celor 2 axe, prezintă o legendă a formelor de undă, în cazul în care pe aceeași reprezentare se suprapun mai multe astfel de grafice, prezintă punctele de conjuncție ale funcției, dacă graficele au culori diferite sau alte elemente care pot ușura vizualizarea unei asemenea figuri.

Pentru toate aceste detalii MatLab-ul oferă o serie de proprietăți ale acestor funcții de reprezentare grafică.

Funcția	Proprietatea
<b>axis</b>	Setarea axelor vizibile sau invizibile
<b>title</b>	Atribuie graficului un titlu
<b>xlabel</b>	Atribuie un text axei X
<b>ylabel</b>	Atribuie un text axei Y
<b>text</b>	Afișează text pe grafic
<b>gtext</b>	Afișează text pe grafic la trecerea cu mouse-ul asupra lui
<b>color</b>	Setează o culoare graficului
<b>linestyle</b>	Definește forma liniilor cu care se face afișarea
<b>fontsize</b>	Setează dimensiunea caracterelor
<b>legend</b>	Permite inserarea unei legende
<b>grid</b>	Permite plasarea caroiajului
<b>linewidth</b>	Setează grosimea liniei de afișare a graficului
<b>hold</b>	Permite reținerea unui grafic pe axe

Tabel 3.3 Proprietăți ale funcțiilor de plotare

Culorile cu care se pot reprezenta grafic rezultatele unei instrucțiuni MatLab se pot defini fie în codul RGB (roșu, verde, albastru) prin crearea unei culori cu ajutorul unui vector de 3 elemente de forma: [R G B], sau prin utilizarea unor prescurtări recunoaste implicit de program:

<b>Culoarea</b>	<b>Numele</b>	<b>Prescurtarea</b>
albastru	blue	<b>b</b>
Verde	Green	<b>g</b>
Rosu	Red	<b>r</b>
Galben	Yellow	<b>y</b>
Cyan	Cyan	<b>c</b>
Purpuriu	Magenta	<b>m</b>
Negru	Black	<b>k</b>
Alb	White	<b>w</b>

Tabel 3.4 Culori predefinite în Matlab

Tipul liniei cu care se realizează afișarea grafică poate să fie de forma liniilor prezentate în tabelul 3.5.

<b>Tipul liniei</b>	<b>Prescurtarea</b>
Linie continua	-
Linie punctata	:
Linie si punct	-.
Linie intrerupta	--

Tabel 3.5 Forma și prescurtarea liniilor de reprezentare grafică

Pentru a marca pe grafic punctele de conjuncție ale funcția ce trebuie plotată, se pot utiliza simboluri din urmatorul tabel:

<b>Simbol</b>	<b>Tipul de Marker</b>
.	Punct
<b>o</b>	Cerculet
<b>x</b>	X
+	Plus
*	Steluta
<b>s</b>	Patrat
<b>d</b>	Romb
<b>v</b>	Triunghi sus
^	Triunghi jos
<	Triunghi stanga
>	Triunghi dreapta
<b>p</b>	Stea cu 5 varfuri
<b>h</b>	Stea cu 6 varfuri

Tabel 3.6 Simboluri de reprezentare a punctelor de conjuncție

Folosind funcțiile prezentate anterior se cere să se reprezinte grafic 3 semnale sinusoidale

prezentate în sistemul de mai jos: 
$$\begin{cases} x_1 = \sin(2 \cdot \pi \cdot 50 \cdot t) \\ x_2 = \sin(2 \cdot \pi \cdot 50 \cdot t + 60^\circ) \\ x_3 = \sin(2 \cdot \pi \cdot 50 \cdot t + 150^\circ) \end{cases}$$

```
f=50; %setam frecventa semnalelor
T=1/f %calculam perioada semnalelor
t=0:0.001:2*T; %alegem un interval de timp de reprezentare
ph1=60*pi/180; %calculam primul defazaj transformare in rad
ph2=150*pi/180; %calculam al doilea defazaj transformare in rad
x=sin(2*pi*f*t); %calculam primul sinus
x1=sin(2*pi*f*t+ph1); %calculam al 2 sinus
x2=sin(2*pi*f*t+ph2); %calculam al 3 sinus
plot(t,x,'-or'); %afisam primul sinus
hold on; %il mentinem pe axe
plot(t,x1,':db'); %afisam al doilea sinus
hold on; %il mentinem pe axe
plot(t,x2,'--Xg'); %afisam cel de-al treilea sinus
grid on;
title('Reprezentare sinus');%atribuim un titlu graficului
xlabel('timp [s]'); %scriem text pe axa x
ylabel('Amplitudine [V]'); %scriem text pe axa y
legend('x1','x2','x3'); %inseram o legenda
```

Rezultatul instrucțiunilor Matlab anterioare este prezentat în figura 3.4.

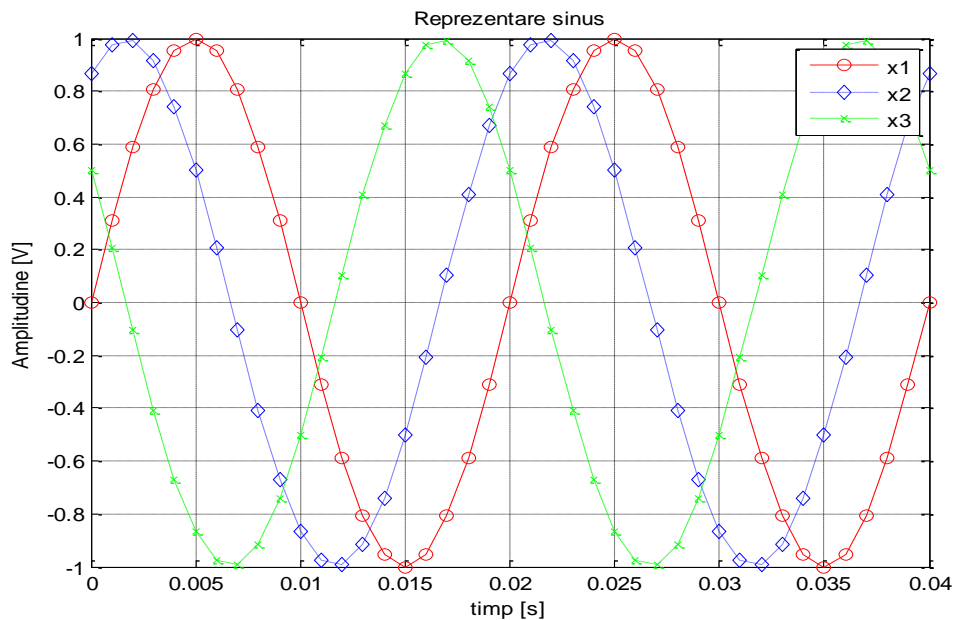


Figura 3.4. Reprezentarea sistemului de 3 sinuși

Se observa în aceasta figură că fiecare dintre cele 3 funcții sinusoidale au fost reprezentate prin culori diferite, puncte de conjuncție, iar graficul prezintă un titlu, specificații pentru cele 2 axe și o legendă. În codul scris anterior apare funcția **hold on** după fiecare plotare, astfel încât menținem fiecare grafic atunci când un altul urmează a fi plotat.

O metodă de reprezentare a color 3 funcții sinusoidale prin utilizarea funcției plot o singură dată se poate face utilizând linia de cod de forma:

```
plot(t,x,'-or',t,x1,':db',t,x2,'--Xg');  
grid on;
```

Această reprezentare este posibilă deoarece cele 3 funcții ce trebuie reprezentate au aceeași bază de timp.

Putem spune că afișarea unui grafic cu ajutorul funcției plot este simplă indiferent dacă datele de intrare sunt de forma unor vectori sau a unor matrici, dacă ținem cont de câteva reguli simple :

**plot(x)** – dacă x este un vector atunci vom genera un grafic în funcție de valoarea lui x și poziția acestuia în vector

**plot(x,y)** – dacă x și y sunt vectori de aceeași lungime atunci vom avea un grafic y în funcție de x

Să presupunem că avem de afișat urmatorul set de date:

```
x1 = 0:1:20;      %x1 i-a o valoare in intervalul [0 20]  
y1 = x1.*sin(x1); %functia y1  
x2 = 10:2:25;    %x2 i-a o valoare intre [10 25]  
y2 = 50*x2;      %functia y2
```

Dacă reprezentăm grafic cele 2 funcții y1 și y2 cu ajutorul funcției **plot** vom obține rezultatul din figura 3.5. Așa cum se poate observa și în figura 3.5 rezultatul nu pare deloc concludent, iar citirea valorilor este greoaie. Soluția unei reprezentări corecte este dată de utilizarea funcției **plotyy**, funcție cu ajutorul căreia avem posibilitatea de a seta 2 axe y pentru reprezentare dorită.



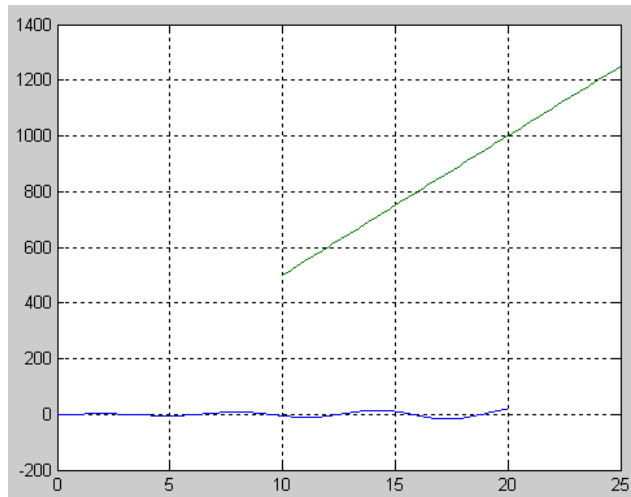


Figura 3.5. Reprezentarea funcțiilor  $y_1$ ,  $y_2$  cu ajutorul funcției plot

```
[axeshandles,line1handle,line2handle]=plotyy(x1,y1,x2,y2);
set(line1handle,'linestyle','-','color','red');
set(line2handle,'linestyle','--','color','g');
grid;
title('y1 is the blue line, y2 is the green dashed line');
axes(axeshandles(1));
ylabel('y1=x.*sin(x)');
axes(axeshandles(2));
ylabel('y2=50*x','color','k');
xlabel('x');
```

Rezultatul liniilor de cond anterioare se poate vizualiza in figura 3.6.

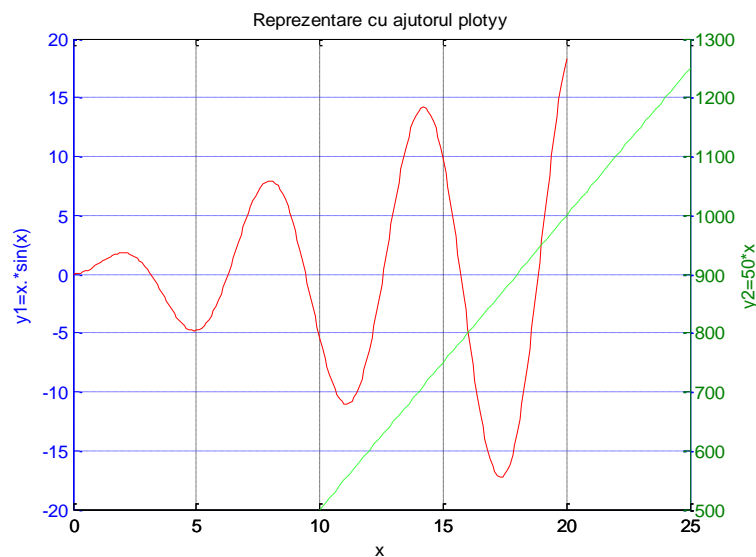


Figura 3.6. Reprezentare a funcțiilor  $y_1$  și  $y_2$  utilizând funcția **plotyy**

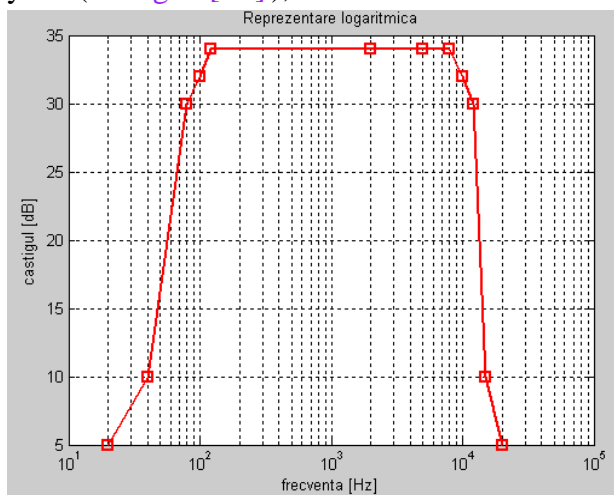
Figura 3.6 prezintă reprezentarea celor 2 funcții într-un sistem cu 2 axe y. Astfel citirea valorilor este mult mai corectă și exactă, iar cele 2 funcții se pot acum interpreta.

În analiza circuitelor electronice de multe ori este necesar să reprezentăm logaritmic rezultatele obținute (ex: filtre). Dacă ne gândim la un filtru, câștigul este exprimat de tensiunea de ieșire raportată la tensiunea de intrare în funcție de frecvență. Tabelul 3.7 prezintă valori ale câștigului unui filtru la diferite frecvențe. Se impune reprezentarea grafică logaritmică a acestor valori.

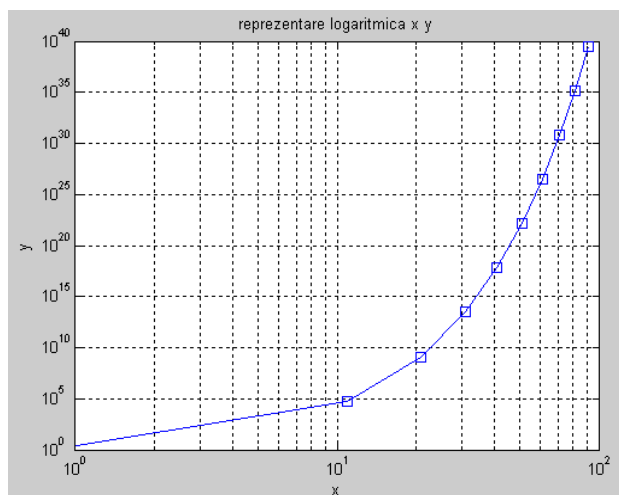
Frecvența	Câștigul	Frecvența	Câștigul
20	5	2000	34
40	10	5000	34
80	30	8000	34
100	32	10000	32
120	34	12000	30
15000	10	20000	5

Tabel 3.7 Valori ale frecvenței și câștigului pentru un filtru oarecare

```
f=[20 40 80 100 120 2000 5000 8000 10000 12000 15000 20000]; % incarc vectorul frecventa
g=[5 10 30 32 34 34 34 34 32 30 10 5]; % incarc vectorul castig
semilogx(f,g,'-sr','linewidth',2); % afisez logaritmic pe x
grid on;
title('Reprezentare logaritmica');
xlabel('frecventa [Hz]');
ylabel('castigul [dB]');
```



a)



b)

Fig. 3.7 Rezultatele obținute în urma reprezentării logaritmică a valorilor din tabelul 3.7. a) logaritizarea axei x; b) logaritizarea axelor x,y

În figura 3.7 a) se prezintă rezultatele obținute în urma codului Matlab prezentat anterior. Dacă se folosește funcția **semilogy**, vom obține o reprezentare de forma axei y logaritmată și axei x liniare. Reprezentare grafică logaritmică atât pe axa x cât și pe axa y se obține folosind funcția **loglog**. O astfel de reprezentare a valorilor din tabelul 3.7 este prezentată în figura 3.7 b).

Reprezentarea logaritmică are la baza trei funcții cu ajutorul cărora putem realiza logaritmarea axelor:

**semilogx(x,y)**: - generează un grafic de  $\log_{10}(x)$  versus axa y liniara

**semilogy(x,y)**: - generează un grafic cu axa x liniara versus  $\log_{10}(y)$

**loglog(x,y)**: - generează un grafic de  $\log_{10}(x)$  versus  $\log_{10}(y)$

O altă modalitate de reprezentare 2D este aceea a diagramelor de tip bară. Această reprezentare este caracterizată de funcțiile **bar** sau **barh**.

Se consideră următoare secvență de cod:

```
x=1:3:33;
y=[66 56 50 3 89 5 78 65 4 25 99];
bar(x,y);
grid on;
```

Rezultatul acestor linii este prezentat în figura 3.8.

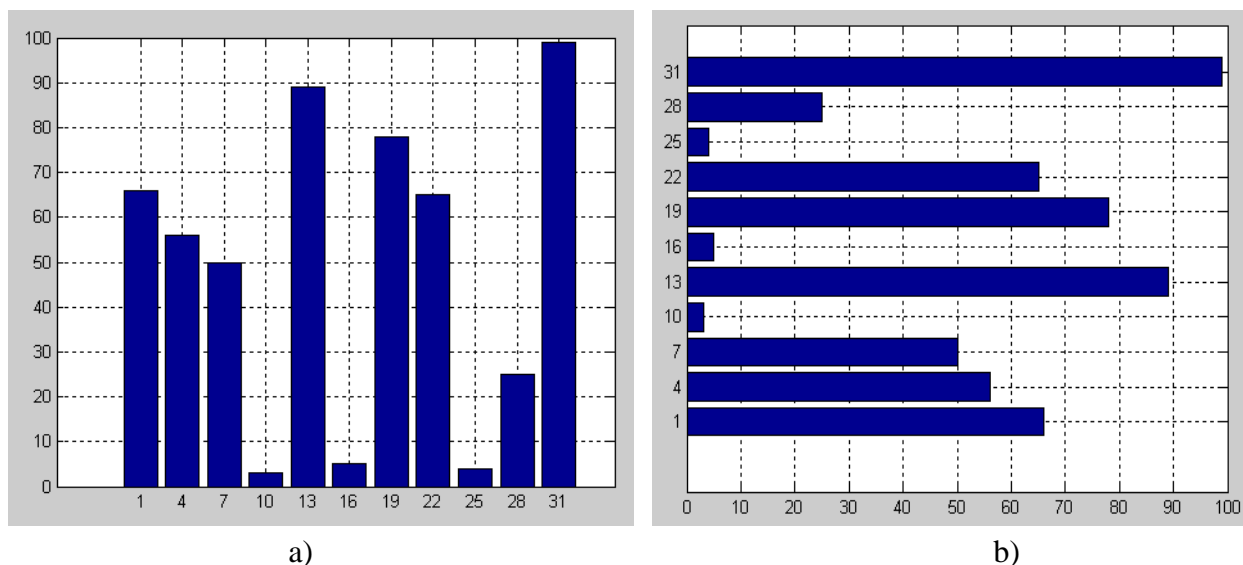


Figura 3.8. Reprezentare cu ajutorul funcției **bar** asupra vectorilor x,y a) și utilizând funcția **barh** b)

Programul MatLab permite de asemenea afișarea mai multor zone de plotare în cadrul unei ferestre de lucru. Pentru aceasta se utilizează funcția **subplot**. Această funcție este caracterizată printr-un vector de 3 valori ce pot fi definite de către utilizator astfel:

- subplot (a b c)** , unde a – reprezintă numărul de ferestre pe verticală
- b – reprezintă numărul de ferestre pe orizontală
- c – poziția în care se dorește afișarea.

În figura 3.9 se prezintă câteva exemple de utilizare ale funcției subplot.

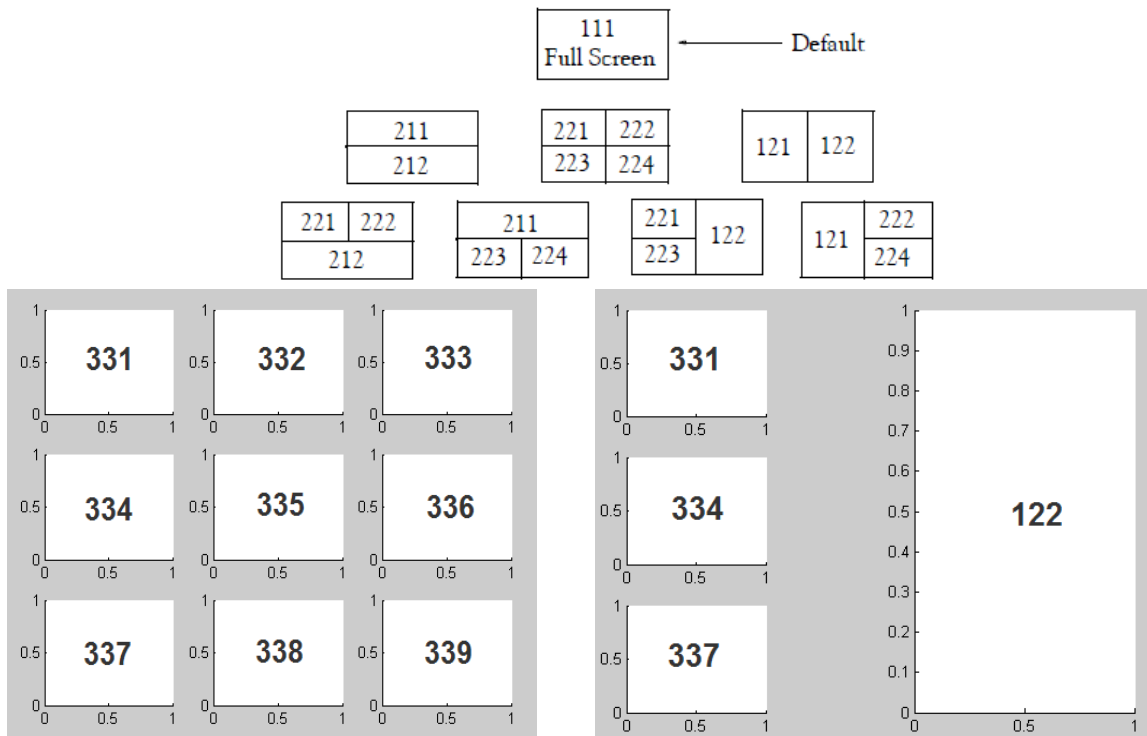


Figura 3.9. Exemple de valori ale vectorului funcției subplot

### 3.1.3 Reprezentări grafice 3D

Pe lângă sistemul de reprezentare pe două axe programul Matlab oferă posibilitatea de a reprezenta grafic soluțiile unui sistem și prin reprezentări grafice 3D. Astfel, vom avea un sistem de trei axe X,Y,Z, ceea ce implică un al treilea parametru al unei funcții de reprezentare.

Una din funcțiile ce permit reprezentarea grafică 3D este funcția **plot3**. Un exemplu al acestei funcții este prezentat în liniile de cod următoare:

```
t = 0:pi/50:10*pi;
subplot(221)
plot(t,sin(t),'-r'); %reprezintă 2D sinus
grid on;
subplot(223)
plot(t,cos(t),'-g') %reprezintă 2D cosinus
subplot(122)
plot3(sin(t),cos(t),t,'pb'); %reprezintă 3D
```

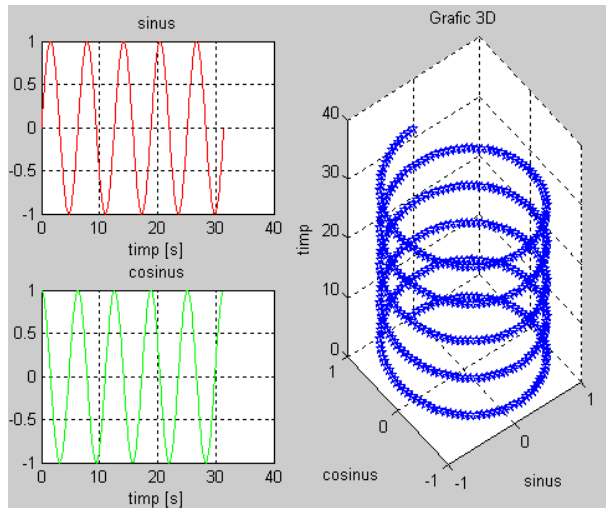


Figura 3.10. Reprezentare 3D utilizând funcția plot3

Figura 3.10 prezintă reprezentarea 3D a funcțiilor sinus și cosinus pe un interval de timp. La fel ca și în cazul funcției plot, funcția plot3 permite setarea unor elemente grafice ce pot ușura interpretarea graficului obținut.

De cele mai multe ori prin grafica 3D se dorește reprezentarea grafică a unor suprafețe. Astfel, programul Matlab dispune de funcții predefinite pentru astfel de reprezentări cum ar fi funcțiile: **surf** și **mesh**.

Funcția **Mesh** realizează un schelet al unei suprafețe dată de coordonatele x, y și z.

```
[X, Y] = meshgrid(-pi:pi/10:pi);
Z = sin (X) .* sin (Y);
mesh (Z);
```

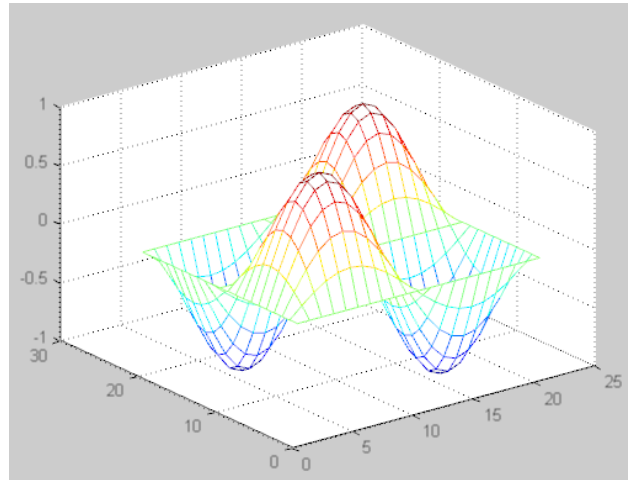


Figura 3.11. Reprezentare 3D utilizând funcția Mesh

Funcția **Surf** permite vizualizarea unor funcții matematice într-o suprafață rectangulară.

```
[X, Y] = meshgrid(-pi:pi/10:pi);
Z = sin (X) .* sin (Y);
surf (Z);
```

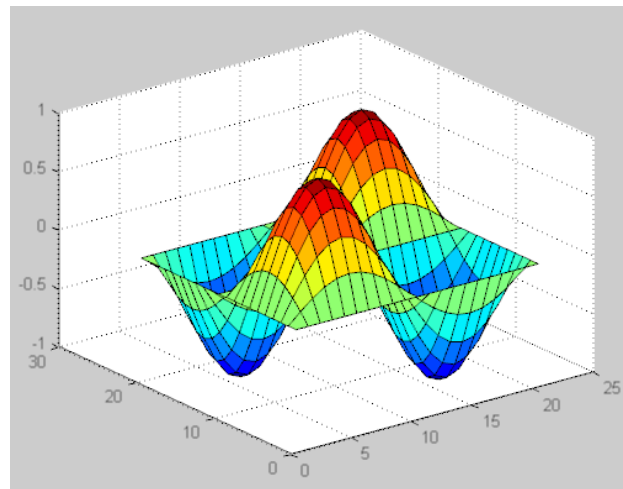


Figura 3.12. Reprezentare 3D cu ajutorul funcției Surf

O prezentare mai detaliată a reprezentărilor grafice 3D se realizează în lucrările următoare unde vom încerca să înțelegem sistemul matricial de compunere a unor astfel de grafice utilizând funcția **meshgrid**.

### 3.2 Desfășurarea laboratorului

1. Să se reprezinte grafic funcția:

$$f(x) = \frac{x-3}{x^2-5}, \quad \text{pentru } x \in [-5,5]$$

2. Să se reprezinte grafic funcția:

$$f(x) = 2e^{x^2} - \sin(x^2)$$

3. Se dau funcțiile  $m(t)$ ,  $c(t)$ , și  $s(t)$  caracteristice unui sistem de comunicații:

$$m(t) = 4\cos(120\pi t) + 2\cos(240\pi t)$$

$$c(t) = 10\cos(10000\pi t)$$

$$s(t) = m(t) \cdot c(t)$$

Reprezentați grafic cele 3 funcții astfel încât utilizând comanda subplot.

4. Utilizând funcția plotyy reprezentați grafic tensiunea și curentul unui circuit RL date de forma:

$$\begin{cases} v(t) = 10\cos(377t) \\ i(t) = 5\cos(377t + 60^\circ) \end{cases}$$