

# INTRODUCERE ÎN SOFTWARE MATEMATIC

## CURS #4

Liviu Marin

Facultatea de Matematică și Informatică, Universitatea din București, România

E-mails: marin.liviu@gmail.com; liviu.marin@fmi.unibuc.ro

### 1 Reprezentări grafice bidimensionale

- Comanda `plot`
- Comanda `fplot`
- Reprezentări grafice multiple în aceeași figură
- Formatarea reprezentărilor grafice 2D
- Alte reprezentări grafice
- Histograme

Liviu Marin

Introducere în software matematic: Curs #4

Reprezentări grafice bidimensionale

Comanda `plot`  
 Comanda `fplot`  
 Reprezentări grafice multiple în aceeași figură  
 Formatarea reprezentărilor grafice 2D  
 Alte reprezentări grafice  
 Histograme

Liviu Marin

Introducere în software matematic: Curs #4

Reprezentări grafice bidimensionale

Comanda `plot`  
 Comanda `fplot`  
 Reprezentări grafice multiple în aceeași figură  
 Formatarea reprezentărilor grafice 2D  
 Alte reprezentări grafice  
 Histograme

## Reprezentări grafice bidimensionale

### Comanda `plot`

- Reprezentarea grafică a **seturilor de date**.
- Reprezentarea grafică a **funcțiilor**.

`plot(x, y)`

- **x, y** – vectori (tablouri unidimensionale) de aceeași dimensiune;
- **x** – valori pe abscisă (axa orizontală);
- **y** – valori pe ordonată (axa verticală).

### EXEMPLUL #1: Reprezentare grafică simplă în Command Window

Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> x = [1 2 3 5 7 7.5 8 10];
>> y = [2 6.5 7 7 5.5 4 6 8];
>> size(x)
```

```
ans =

     1     8
```

```
>> size(y)

ans =

     1     8
```

```
>> plot(x,y)
fx>>
```

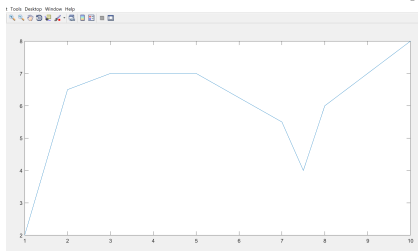
Liviu Marin

Introducere în software matematic: Curs #4

Liviu Marin

Introducere în software matematic: Curs #4

## EXEMPLUL#1: Figura generată în MATLAB



`plot(x, y, 'line specifiers', 'PropertyName', PropertyValue)`

- o **x, y** – vectori (tablouri unidimensionale) de aceeași dimensiune;
- o **x** – valori pe abscisă (axa orizontală);
- o **y** – valori pe ordonată (axa verticală);
- o **'line specifiers'** – specificatori ce definesc stilul și culoarea liniei și ale markerului (argument opțional);
- o **'PropertyName'** – numele proprietăților liniei (grosimea) și ale markerului (dimensiunea și marginile) (argument opțional);
- o **PropertyValue** – valorile proprietăților liniei (grosimea) și ale markerului (dimensiunea și marginile) (argument opțional).

## 1. Specificatori (de linie și de marker)

Stilul liniei	Specificator
linie continuă (valoare implicită)	—
linie întreruptă	— —
linie punctată	:
linie întreruptă-punctată	— .

Culoarea liniei	Specificator
roșie	r
verde	g
albastră (valoare implicită)	b
cyan	c
magenta	m
galbenă	y
neagră	k
albă	w

Tipul markerului	Specificator
plus	+
cerc	o
asterix	*
punct	.
cruce	x
pătrat	s
romb	d
stea cu cinci colțuri	p
stea cu șase colțuri	h
triunghi cu vârful în sus	^
triunghi cu vârful în jos	v
triunghi cu vârful la stânga	<
triunghi cu vârful la dreapta	>

**OBSERVAȚII:**

- Specificatorii (de linie și de marker) se scriu în comanda `plot` ca **șir de caractere/string**.
- Specificatorii (de linie și de marker) pot fi scriși în cadrul comenzii `plot`, în **orice ordine**.
- Specificatorii (de linie și de marker) sunt **opționali**.

**EXEMPLE:**

```
plot(x, y)
plot(x, y, 'r')
plot(x, y, '--b')
plot(x, y, '*')
plot(x, y, 'g:d')
```

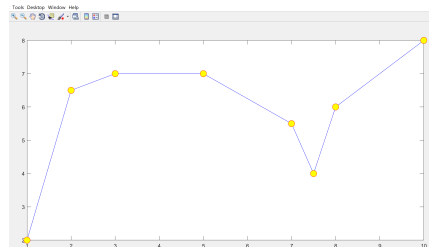
**2. Numele și valorile proprietăților liniei și ale markerului**

Nume proprietate	Descriere	Posibile valori
LineWidth (linewidth)	Grosimea liniei	Număr în unități de punct (implicit, 0.5)
MarkerSize (markersize)	Mărimea markerului	Număr în unități de punct
MarkerEdgeColor (markeredgecolor)	Culoarea markerului sau a muchiei markerului	Specificatori de culoare ca șir de caractere/string
MarkerFaceColor (markerfacecolor)	Culoarea feței markerului	Specificatori de culoare ca șir de caractere/string

**EXEMPLUL#2.1: Reprezentare grafică cu specificatori opționali, v.1, în Command Window**

New to MATLAB? See resources for [Getting Started](#).

```
>> x = [1 2 3 5 7 7.5 8 10];
>> y = [2 6.5 7 7 5.5 4 6 8];
>> plot(x, y, '-bo', 'MarkerSize', 12, ...
'MarkerEdgeColor', 'r', 'MarkerFacecolor', 'y')
fx >>
```

**EXEMPLUL#2.1: Figura generată în MATLAB**

**OBSERVAȚIE:** Specificatorii (de linie și de marker) pot fi definiți în mod similar prin intermediul unui nume și a unei valori ale proprietăților liniei/markerului.

Specificator	Nume proprietate	Possibile valori
Stilul liniei	LineStyle (linestyle)	Specificator de linie (v. tabelul) ca șir de caractere/string
Culoarea liniei	Color (color)	Specificator de culoare (v. tabelul) ca șir de caractere/string
Tipul markerului	Marker (marker)	Specificator de marker (v. tabelul) ca șir de caractere/string

## EXEMPLUL#2.2: Reprezentare grafică cu specificatori opționali, v.2, în Command Window

Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> x = [1 2 3 5 7 7.5 8 10];
>> y = [2 6.5 7 7 5.5 4 6 8];
>> plot(x, y, 'LineStyle', '-', 'Color', 'b', ...
'Marker', 'o', 'MarkerSize', 12, ...
'MarkerEdgeColor', 'r', 'MarkerFaceColor', 'y')
fx >>
```

## Comanda fplot

- Reprezentarea grafică a unei funcții  $y = f(x)$  între limite specificate.

`fplot(function, limits, 'line specifiers')`

- **function** – expresia funcției  $f(x)$  scrisă ca o funcție anonimă;
- **limits** – domeniul de definiție a funcției (axa orizontală) și, opțional, limite pentru axa  $Oy$  (axa verticală), scrise ca un vector linie, i.e.  $[x_{min}, x_{max}]$ , respectiv  $[x_{min}, x_{max}, y_{min}, y_{max}]$ ;
- **'line specifiers'** – specificatori ce definesc tipul și culoarea liniei și ale markerului (argument opțional).

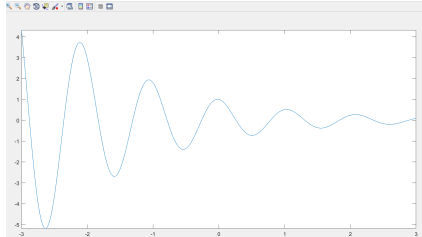
## EXEMPLUL#3: Reprezentarea grafică a funcției

$$f : [-3, 3] \rightarrow \mathbb{R}, \quad f(x) = 3.5^{x/2} \cos(6x)$$

```
>> fplot(@(x) 3.5^(-0.5 * x) * cos(6 * x), [-3 3])
```

## EXEMPLUL#3: Figura generată în MATLAB cu comanda fplot

Tools Desktop Window Help



## Reprezentări grafice multiple în aceeași figură

Mai multe grafice pot fi reprezentate în aceeași figură în trei moduri:

- ▶ prin folosirea comenzii `plot`;
- ▶ prin folosirea comenzilor `hold on` și `hold off`;
- ▶ prin folosirea comenzii `line`.

### 1. Reprezentări grafice multiple folosind comanda `plot`

```
plot(x1, y1, 'line specifiers1', 'PropertyName1', PropertyValue1, ...
     x2, y2, 'line specifiers2', 'PropertyName2', PropertyValue2, ...
     x3, y3, 'line specifiers3', 'PropertyName3', PropertyValue3)
```

- `x1, y1` – vectori de aceeași dimensiune pentru funcția#1;
- `x2, y2` – vectori de aceeași dimensiune pentru funcția#2;
- `x3, y3` – vectori de aceeași dimensiune pentru funcția#3;
- `'line specifiers1', 'line specifiers2', 'line specifiers3'` – specificați ce definesc stilul și culoarea liniei și ale markerului (argument opțional);
- `'PropertyName1', 'PropertyName2', 'PropertyName3'` – numele proprietăților liniei și ale markerului (argument opțional);
- `PropertyValue1, PropertyValue2, PropertyValue3` – valorile proprietăților liniei și ale markerului (argument opțional).

**EXEMPLUL#4.1:** Reprezentarea grafică a funcției  $f: [-2, 4] \rightarrow \mathbb{R}$ ,  $f(x) = 3x^3 - 26x + 10$ , și a derivatelor sale de ordinul întâi și de ordinul doi, folosind comanda `plot`.

```
1 x = -2 : 0.01 : 4;
2 y = 3*x.^3 - 26*x + 10;
3 y1= 9*x.^2 - 26;
4 y2= 18*x;
5 figure(1)
6 plot(x, y, 'LineStyle', '-', 'Color', 'k', ...
7      x, y1, 'LineStyle', '--', 'Color', 'b', ...
8      x, y2, 'LineStyle', ':', 'Color', 'r')
```

### 2. Reprezentări grafice multiple folosind comenzile `hold on` și `hold off`

```
plot(x1, y1, 'line specifiers1', 'PropertyName1', PropertyValue1)
hold on
plot(x2, y2, 'line specifiers2', 'PropertyName2', PropertyValue2)
plot(x3, y3, 'line specifiers3', 'PropertyName3', PropertyValue3)
hold off
```

- `x1, y1` – vectori de aceeași dimensiune pentru funcția#1;
- `x2, y2` – vectori de aceeași dimensiune pentru funcția#2;
- `x3, y3` – vectori de aceeași dimensiune pentru funcția#3;
- `'line specifiers1', 'line specifiers2', 'line specifiers3'` – specificați ce definesc stilul și culoarea liniei și ale markerului (argument opțional);
- `'PropertyName1', 'PropertyName2', 'PropertyName3'` – numele proprietăților liniei și ale markerului (argument opțional);
- `PropertyValue1, PropertyValue2, PropertyValue3` – valorile proprietăților liniei și ale markerului (argument opțional).

**EXEMPLUL#4.2:** Reprezentarea grafică a funcției  $f: [-2, 4] \rightarrow \mathbb{R}$ ,  $f(x) = 3x^3 - 26x + 10$ , și a derivatelor sale de ordinul întâi și de ordinul doi, folosind comenzile **hold on** și **hold off**.

```
1 x = -2 : 0.01 : 4;
2 y = 3*x.^3 - 26*x + 10;
3 y1 = 9*x.^2 - 26;
4 y2 = 18*x;
5 figure(1)
6 plot(x, y, 'LineStyle', '-', 'Color', 'k')
7 hold on
8 plot(x, y1, 'LineStyle', '—', 'Color', 'b')
9 plot(x, y2, 'LineStyle', ':', 'Color', 'r')
10 hold off
```

### 3. Reprezentări grafice multiple folosind comanda **line**

```
plot(x1, y1, 'line specifiers', 'PropertyName1', PropertyValue1)
line(x2, y2, 'PropertyName2', PropertyValue2)
line(x3, y3, 'PropertyName3', PropertyValue3)
```

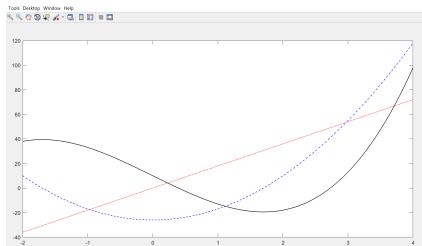
- o **x1, y1** – vectori de aceeași dimensiune pentru funcția#1;
- o **x2, y2** – vectori de aceeași dimensiune pentru funcția#2;
- o **x3, y3** – vectori de aceeași dimensiune pentru funcția#3;
- o **'line specifiers'** – specificatori ce definesc stilul și culoarea liniei și ale markerului (argument opțional);
- o **'PropertyName1', 'PropertyName2', 'PropertyName3'** – numele proprietăților liniei și ale markerului (argument opțional);
- o **PropertyValue1, PropertyValue2, PropertyValue3** – valorile proprietăților liniei și ale markerului (argument opțional).

**OBSERVAȚIE:** Comanda **plot** începe execuția unei figuri, în vreme ce comanda **line** adaugă noi linii unei figuri deja existente!

**EXEMPLUL#4.3:** Reprezentarea grafică a funcției  $f: [-2, 4] \rightarrow \mathbb{R}$ ,  $f(x) = 3x^3 - 26x + 10$ , și a derivatelor sale de ordinul întâi și de ordinul doi, folosind comanda **line**.

```
1 x = -2 : 0.01 : 4;
2 y = 3*x.^3 - 26*x + 10;
3 y1 = 9*x.^2 - 26;
4 y2 = 18*x;
5 figure(1)
6 plot(x, y, 'LineStyle', '-', 'Color', 'k', '
   LineWidth', 1.0)
7 line(x, y1, 'LineStyle', '—', 'Color', 'b', '
   LineWidth', 1.0)
8 line(x, y2, 'LineStyle', ':', 'Color', 'r', '
   LineWidth', 1.0)
```

**EXEMPLELE#4.1–4.3:** Figura generată în MATLAB cu comenzile **fplot**, **hold on** și **hold off**, respectiv **line**.



## Formatarea reprezentărilor grafice 2D

- ▶ Comenzile `plot` și `fplot` generează figuri destul de sumare în MATLAB, fără a furniza detalii suplimentare asupra graficului (graficelor) respectiv(e).
- ▶ Pentru a avea astfel de informații suplimentare asupra graficului, figurile trebuie **formate**, e.g. specificând etichetele axelor, titlul figurii, legenda, gridul, etichetele de text.
- ▶ Formatarea figurilor se poate face
  - prin intermediul unor **comenzi MATLAB**, folosite după comanda `plot` sau `fplot`;
  - utilizând **editorul grafic** din fereastra de editare a figurilor (Figure Window).

## Formatarea reprezentărilor grafice prin intermediul comenzilor MATLAB

### 1. Comenzile `xlabel` și `ylabel`

`xlabel('xLabel as string')`

`ylabel('yLabel as string')`

- Etichetele `'xLabel as string'` și `'yLabel as string'` sunt plasate sub axele  $Ox$ , respectiv  $Oy$ .

### 2. Comanda `title`

`title('Title as string')`

- Titlul `'Title as string'` este plasat deasupra figurii.

### 3. Comanda `text`

`text(x, y, 'Text as string')`

`gtext('Text as string')`

- Eticheta de text `'Text as string'` este plasată de comanda `text` în interiorul figurii, a.i. primul caracter este plasat în punctul de coordonate  $(x, y)$ .
- Eticheta de text `'Text as string'` este plasată de comanda `gtext` în interiorul figurii, în locul specificat de utilizator.

### 4. Comanda `legend`

`legend('string1', 'string2', ..., 'Location', 'LocationName')`

- ▶ `'string1', 'string2', ...` – etichete plasate în dreptul tipurilor de linie corespunzătoare folosite în reprezentarea grafică;
- ▶ `'Location'` – specificator de poziție a legendei (opțional);
- ▶ `'LocationName'` – valoare a specificatorului de poziție a legendei (opțional):
  - `'North'` – plasată în partea de sus-centru (N) a figurii;
  - `'West'` – plasată în partea stânga-centru (W) a figurii;
  - `'South'` – plasată în partea de jos-centru (S) a figurii;
  - `'East'` – plasată în partea dreaptă-centru (E) a figurii;
  - `'NorthEast'` – plasată în colțul din dreapta sus (NE) al figurii;
  - `'NorthWest'` – plasată în colțul din stânga sus (NW) al figurii;
  - `'SouthWest'` – plasată în colțul din stânga jos (SW) al figurii;
  - `'SouthEast'` – plasată în colțul din dreapta jos (SE) al figurii.

## 5. Formatarea textului în comenzile xlabel, ylabel, title, text și legend

- ▶ Formatarea textului poate fi folosită pentru a defini fontul, mărimea, poziția (subscript, superscript), stilul (italic, bold, etc.) și culoarea caracterelor, culoarea background-ului etc.
- ▶ O explicație completă a instrucțiunilor de formatare se poate găsi în [Help Window](#), la itemii [Text](#), [Text Properties](#).
- ▶ Formatarea textului se poate face
  - prin adăugarea [modificatorilor de text](#) în șirul de caractere/string din comanda respectivă;
  - prin adăugarea la comanda opțională [PropertyName](#) a argumentului corespunzător [PropertyValue](#) după șirul de caractere/stringul din comanda respectivă.

- ▶ **Modificatorii de text** sunt caractere inserate în șirul de caractere/string.
- ▶ **Modificatorii de text** afectează textul de la punctul în care sunt inserați până la sfârșitul șirului de caractere/stringului.
- ▶ Este posibil ca **modificatorii de text** să fie aplicați doar asupra unei secțiuni a șirului de caractere/stringului prin scrierea modificatului și a textului de interes între acolade.

Modificator de text	Efect
\bf	font bold
\it	font italic
\rm	font normal
\fontname{fontname}	tip de font specificat
\fontsize{fontsize}	mărime de font specificată

## Litere grecești

### Subscript

- ▶ Un caracter poate fi afișat ca subscript prin tipărirea semnului  $_$  ([underscore](#)) înaintea respectivului caracter.
- ▶ Mai multe caractere pot fi afișate ca subscript prin tipărirea semnului  $_$  înaintea respectivelor caractere incluse între acolade.

### Superscript

- ▶ Un caracter poate fi afișat ca superscript prin tipărirea semnului  $^$  înaintea respectivului caracter.
- ▶ Mai multe caractere pot fi afișate ca superscript prin tipărirea semnului  $^$  înaintea respectivelor caractere incluse între acolade.

Caractere în string	Litere grecești
\alpha	$\alpha$
\beta	$\beta$
\gamma	$\gamma$
\delta	$\delta$
\theta	$\theta$
\pi	$\pi$
\sigma	$\sigma$
\Gamma	$\Gamma$
\Delta	$\Delta$
\Lambda	$\Lambda$
\Theta	$\Theta$
\Omega	$\Omega$
\Sigma	$\Sigma$
\Phi	$\Phi$



- Numele proprietății **PropertyName** este tipărit ca un șir de caractere/string.
- Valoarea proprietății **PropertyValue** fie este un număr dacă așa este cazul, fie este tipărită ca un șir de caractere/string dacă este un cuvânt sau o literă.

Nume proprietate	Descriere	Posibile valori
Rotation	Orientarea textului	Scalar (grade)
FontAngle	Stilul caracterelor	normal, italic
FontName	Tip de font	Nume font disponibil
FontSize	Mărima fontului	Scalar (puncte)
FontWeight	Grosimea fontului	light, normal, bold
Color	Culoarea textului	Specificatori de culoare
BackgroundColor	Culoarea background-ului	Specificatori de culoare
EdgeColor	Culoarea muchiei încadrării dreptunghiulare a textului	Specificatori de culoare
LineWidth	Grosimea muchiei încadrării dreptunghiulare a textului	Scalar (puncte)

## 6. Comanda axis

- Când este executată comanda **plot(x, y)**, MATLAB creează axele cu limitele date de valorile minime și maxime ale vectorilor **x** și **y**.
- Comanda **axis** poate fi folosită pentru a modifica limitele și modul în care arată axele.
- Forme posibile ale comenzii **axis**:
  - **axis([xmin xmax ymin ymax])** – setează limitele pentru axele  $Ox$  (i.e. **xmin** și **xmax**) și  $Oy$  (i.e. **ymin** și **ymax**);
  - **axis equal** – setează aceeași scală pentru axele  $Ox$  și  $Oy$ ;
  - **axis square** – setează axele  $Ox$  și  $Oy$  a.i. regiunea graficului să fie pătratică;
  - **axis tight** – setează axele  $Ox$  și  $Oy$  în conformitate cu limitele datelor.

## 7. Comanda grid

- **grid on** – adaugă liniile de grid figurii;
- **grid off** – îndepărtează liniile de grid din figură.

## EXEMPLUL#5: Reprezentări grafice multiple folosind comenzi de formatare.

```

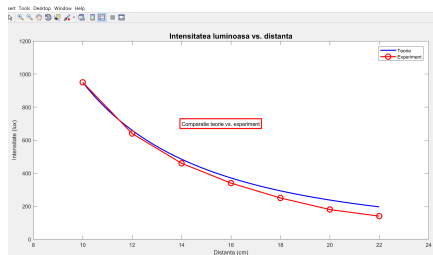
1 x = 10 : 0.1 : 22;
2 y = 95000./x.^2;
3 xd = 10 : 2 : 22;
4 yd = [950 640 460 340 250 180 140];
5 figure(1)
6 plot(x, y, '-b', 'LineWidth', 2)
7 xlabel('Distanța (cm)')
8 ylabel('Intensitate (lux)')
9 title('\fontname{Arial}Intensitatea luminoasă vs.
    distanța', ...
    'FontSize', 14)
10 axis([8 24 0 1200])
11 text(14, 700, 'Comparatie teorie vs. experiment',
12 ...
13 'EdgeColor', 'r', 'LineWidth', 2)

```

```

14 line(xd, yd, 'LineStyle', '-', 'Color', 'r', ...
15      'Marker', 'o', 'MarkerSize', 10, 'LineWidth',
    2)
16 legend('Teorie', 'Experiment', 'Location', '
    NorthEast')
  
```

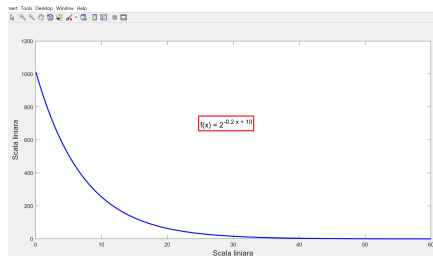
EXEMPLUL#5: Figura generată în MATLAB folosind comenzi de formatare.



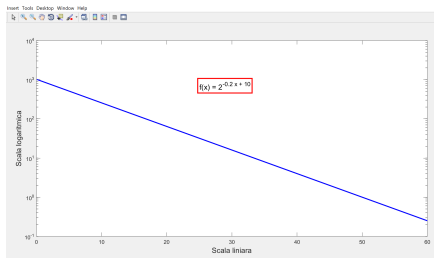
## Alte reprezentări grafice

## 1. Reprezentări grafice cu axe logaritmice

- **semilogy(x, y)** – reprezentare grafică a lui **y** vs. **x** folosind o scală liniară pentru axa **Ox** și o scală logaritmică (în baza 10) pentru axa **Oy**
- **semilogx(x, y)** – reprezentare grafică a lui **y** vs. **x** folosind o scală logaritmică (în baza 10) pentru axa **Ox** și o scală liniară pentru axa **Oy**
- **loglog(x, y)** – reprezentare grafică a lui **y** vs. **x** folosind o scală logaritmică (în baza 10) pentru ambele axe **Ox** și **Oy**;
- Specificatorii ce definesc tipul și culoarea liniei și ale markerului ('**line specifiers**'), numele proprietăților liniei și ale markerului ('**PropertyName**'), precum și valorile proprietăților liniei și ale markerului (**PropertyValue**) pot fi folosite la fel ca în cazul comenzii **plot** (argument opțional)

EXEMPLUL#6.1: Reprezentare grafică folosind comanda **plot**.

### EXEMPLUL#6.2: Reprezentare grafică folosind comanda `semilogy`.



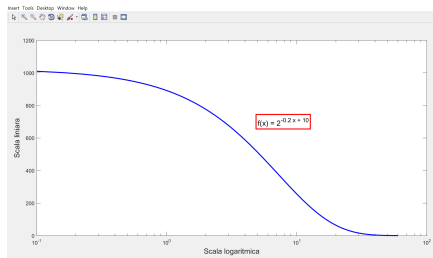
Liviu Marin

Introducere în software matematic: Curs #4

Reprezentări grafice bidimensionale

Comanda plot  
Comanda fplot  
Reprezentări grafice multiple în aceeași figură  
Formatarea reprezentărilor grafice 2D  
Alte reprezentări grafice  
Histograme

### EXEMPLUL#6.3: Reprezentare grafică folosind comanda `semilogx`.



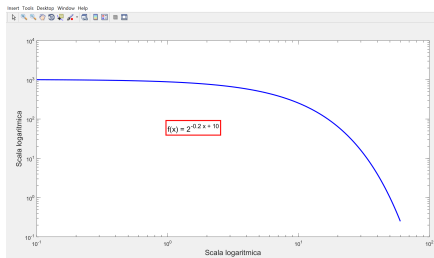
Liviu Marin

Introducere în software matematic: Curs #4

Reprezentări grafice bidimensionale

Comanda plot  
Comanda fplot  
Reprezentări grafice multiple în aceeași figură  
Formatarea reprezentărilor grafice 2D  
Alte reprezentări grafice  
Histograme

### EXEMPLUL#6.4: Reprezentare grafică folosind comanda `loglog`.



Liviu Marin

Introducere în software matematic: Curs #4

## 2. Reprezentări grafice cu bare de eroare

- Datele experimentale conțin, de regulă, erori și acestea pot fi reprezentate grafic odată cu datele respective.
- O metodă de reprezentare grafică a datelor ce conțin erori este dată de comanda MATLAB `errorbar`, i.e.

`errorbar(x, y, e)`

- $x, y, e$  – vectori (tablouri unidimensionale) de aceeași dimensiune;
- $x$  – valori pe abscisă (axa orizontală);
- $y$  – valori pe ordonată (axa verticală);
- $e$  – valori ale erorii în fiecare punct de coordonate  $(x, y)$ .
- Lungimea barei de eroare  $e(i)$  în punctul de coordonate  $(x(i), y(i))$  este  $2 \cdot e(i)$ , fiind simetrică, i.e. se întinde de la  $y(i) - e(i)$  până la  $y(i) + e(i)$ .

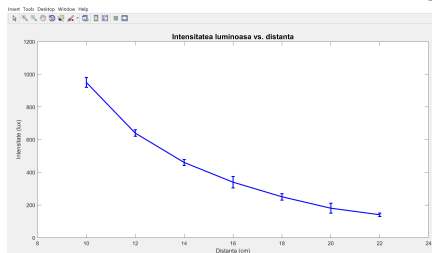
Liviu Marin

Introducere în software matematic: Curs #4

### EXEMPLUL#7: Reprezentare grafică folosind comanda `errorbar`.

```
1 clear; clc; close all
2 xd = 10 : 2 : 22;
3 yd = [950 640 460 340 250 180 140];
4 ed = [30 20 18 35 20 30 10];
5 figure(1)
6 errorbar(xd, yd, ed, '-b', 'LineWidth', 2)
7 xlabel('Distanța (cm)')
8 ylabel('Intensitate (lux)')
9 title('\fontname{Arial}Intensitatea luminoasă vs.
    distanța', ...
10       'FontSize', 14)
11 axis([8 24 0 1200])
```

### EXEMPLUL#7: Figura generată în MATLAB folosind comanda `errorbar`.

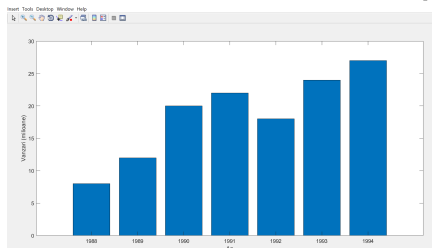


### 3. Reprezentări grafice cu bare verticale/orizontale

#### EXEMPLUL#8.1: Reprezentare grafică folosind comanda `bar`.

```
1 clear; clc; close all
2 An = 1988 : 1994;
3 Vanzari = [8 12 20 22 18 24 27];
4 figure(1)
5 bar(An, Vanzari)
6 xlabel('An')
7 ylabel('Vanzari (milioane)')
```

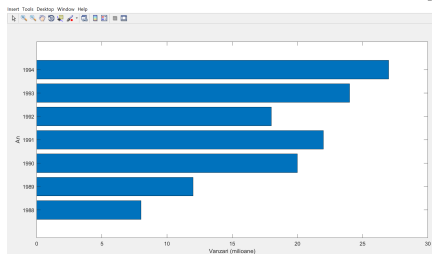
### EXEMPLUL#8.1: Figura generată în MATLAB folosind comanda `bar`.



#### EXEMPLUL#8.2: Reprezentare grafică folosind comanda `barh`.

```
1 clear; clc; close all
2 An = 1988 : 1994;
3 Vanzari = [8 12 20 22 18 24 27];
4 figure(1)
5 barh(An, Vanzari)
6 xlabel('Vanzari (milioane)')
7 ylabel('An')
```

#### EXEMPLUL#8.2: Figura generată în MATLAB folosind comanda `barh`.

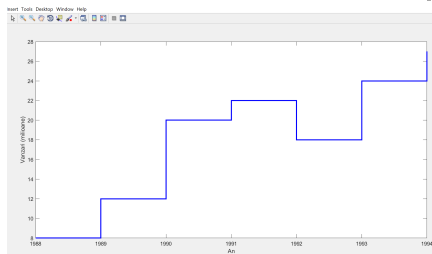


#### 4. Reprezentări grafice folosind funcția treaptă

#### EXEMPLUL#9: Reprezentare grafică folosind comanda `stairs`.

```
1 clear; clc; close all
2 An = 1988 : 1994;
3 Vanzari = [8 12 20 22 18 24 27];
4 figure(1)
5 stairs(An, Vanzari, '-b', 'LineWidth', 2)
6 xlabel('An')
7 ylabel('Vanzari (milioane)')
```

#### EXEMPLUL#9: Figura generată în MATLAB folosind comanda `stairs`.



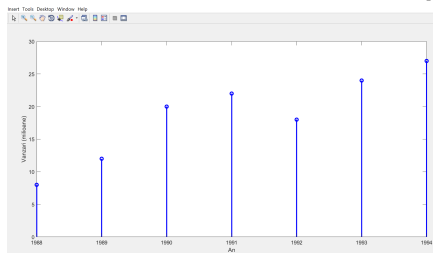
## 5. Reprezentări grafice folosind funcția stem

EXEMPLUL#10: Reprezentare grafică folosind comanda **stem**.

```

1 clear; clc; close all
2 An = 1988 : 1994;
3 Vanzari = [8 12 20 22 18 24 27];
4 figure(1)
5 stem(An, Vanzari, '-b', 'LineWidth', 2)
6 xlabel('An')
7 ylabel('Vanzari (milioane)')

```

EXEMPLUL#10: Figura generată în MATLAB folosind comanda **stem**.

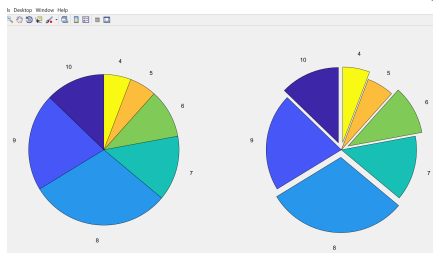
## 6. Reprezentări grafice folosind funcția pie

EXEMPLUL#11: Reprezentare grafică folosind comanda **pie**.

```

1 clear; clc; close all
2 grades = [11 18 26 12 9 5 5];
3 explode = [1 0 1 0 1 0 1];
4 labels = {'10' '9' '8' '7' '6' '5' '4'};
5 figure(1)
6 pie(subplot(1,2,1), grades, labels)
7 pie(subplot(1,2,2), grades, explode, labels)

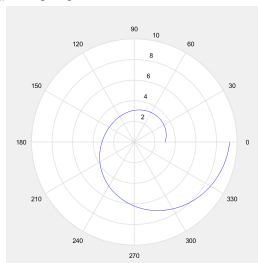
```

EXEMPLUL#11: Figura generată în MATLAB folosind comanda **pie**.

## 7. Reprezentări grafice în coordonate polare

EXEMPLUL#12: Reprezentare grafică folosind comanda **polar**.

```
1 clear; clc; close all
2 theta = linspace(0, 2*pi, 200);
3 rho = 3*cos(0.5*theta).^2 + theta;
4 figure(1)
5 polar(theta, rho, '-b')
```



## Histograme

## 8. Reprezentări grafice multiple în aceeași pagină

`subplot(m, n, p)`

- ▶ pagina este împărțită în  $m \times n$  subpagini;
- ▶  $m$  – numărul de linii în care este împărțită pagina;
- ▶  $n$  – numărul de coloane în care este împărțită pagina;
- ▶  $p$  – pagina actuală, în care se va face reprezentarea grafică,  $1 \leq p \leq m \times n$ ;
- ▶ numărătoarea subpaginilor se face de la stânga la dreapta, de sus în jos.

- ▶ **Histogramele** sunt reprezentări grafice care arată distribuția datelor.
- ▶ Mulțimea în care iau valori datele este împărțită într-un număr de submulțimi (**bins**), iar histogramele arată numărul de date conținute în fiecare submulțime considerată.
- ▶ **Histograma** este o reprezentare grafică prin bare verticale, lățimea fiecărei bare verticale este egală cu mărimea submulțimii considerate, iar înălțimea fiecărei bare verticale este dată de numărul de date conținute în submulțimea considerată.
- ▶ În MATLAB, histogramele se creează folosind comanda **hist**.

Comanda MATLAB **hist**`[n, xout] = hist(y)``[n, xout] = hist(y, nbins)``[n, xout] = hist(y, x)`

- o **y** – vector conținând valorile datelor;
- o MATLAB împarte mulțimea de valori ale datelor în 10 submulțimi (**bins**) egale și apoi reprezintă grafic numărul de date conținute în fiecare submulțime;
- o **nbins** – un scalar ce definește numărul de submulțimi (**bins**) în care se împarte mulțimea în care iau valori datele; MATLAB împarte mulțimea de valori ale datelor în **nbins** submulțimi (**bins**) egale;
- o **x** – vector care conține pozițiile centrilor submulțimilor (**bins**);
- o **n** – vector cu **nbins** elemente (nr. de submulțimi), iar fiecare element al său conține numărul de date conținute în submulțimea respectivă (opțional);
- o **xout** – vector ale cărui elemente conțin pozițiile centrilor submulțimilor (opțional).

Liviu Marin

Introducere în software matematic: Curs #4

Liviu Marin

Introducere în software matematic: Curs #4

EXEMPLUL#13: Reprezentare grafică folosind comanda **hist**.

```

1 clear; clc; close all
2 y = [58 73 73 53 50 48 56 73 73 66 ...
3     69 63 74 82 84 91 93 89 91 80 ...
4     59 69 56 64 63 66 64 74 63 69];
5 figure(1)
6 hist(subplot(1,2,1), y)
7 xlabel('\fontname{Arial}Temperatura (F)', 'FontSize',
8     ', 14)
9 ylabel('\fontname{Arial}Numar de zile', 'FontSize',
10    ', 14)
11 subplot(1,2,2)
12 hist(y, 4)
13 xlabel('\fontname{Arial}Temperatura (F)', 'FontSize',
14    ', 14)

```

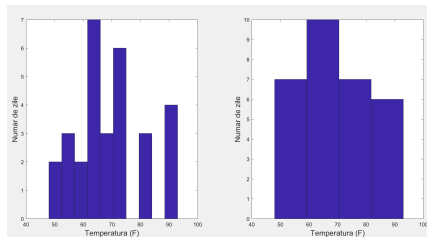
```

12 ylabel('\fontname{Arial}Numar de zile', 'FontSize',
13    14)
14 [n, xout] = hist(y);
15 fprintf('_____ \n')
16 fprintf('n(i)\t\t\t xout(i)\n')
17 fprintf('_____ \n')
18 for i=1:length(n)
19     fprintf('n(%2i) = %-4i\t xout(%2i) = %-.2f F\n',
20         i, n(i), i, xout(i))
21 end
22 fprintf('_____ \n')

```

Liviu Marin

Introducere în software matematic: Curs #4

EXEMPLUL#13: Figurile generate în MATLAB folosind comanda **hist**.

Liviu Marin

Introducere în software matematic: Curs #4



**EXEMPLUL#13:** Tabelul generat în MATLAB folosind comanda **hist**.

Command Window

New to MATLAB? See resources for [Getting Started](#).

```
-----  
n (1)          xout (1)  
-----  
n ( 1) = 2      xout ( 1) = 50.25 F  
n ( 2) = 3      xout ( 2) = 54.75 F  
n ( 3) = 2      xout ( 3) = 59.25 F  
n ( 4) = 7      xout ( 4) = 63.75 F  
n ( 5) = 3      xout ( 5) = 68.25 F  
n ( 6) = 6      xout ( 6) = 72.75 F  
n ( 7) = 0      xout ( 7) = 77.25 F  
n ( 8) = 3      xout ( 8) = 81.75 F  
n ( 9) = 0      xout ( 9) = 86.25 F  
n(10) = 4      xout(10) = 90.75 F  
-----
```

fx &gt;&gt; |