

MATLAB - II

24-May-8

Definirea functiilor (I)

- $f = @(x) 2 * x.^2 - 3 * x + 1$
- $f =$
- $@(x) 2 * x.^2 - 3 * x + 1$
- $f(0)$
- $ans =$
- 1

- `>> v=[1 2 3]`
- `v =`
- `1 2 3`
- `>> f(v)`
- `ans =`
- `0 3 10`
- am scris `x.^2` si nu `x^2` care ar fi insemnat $x*x$ si daca aplicam vectorului `v` atunci `v*v` ar fi insemnat produsul vectorului `v` cu vectorul `v`

Definirea functiilor (II)

- `>> g=@ (x) 2*x.^2-3*x+1`
- `g =`
- `@(x) 2 * x .^ 2 - 3 * x + 1`
- `>> g(2)`
- `ans = 3`
- `>> v=[1 2 3]`
- `v =`
- `1 2 3`
- `>> g(v)`
- `ans =`
- `0 3 10`

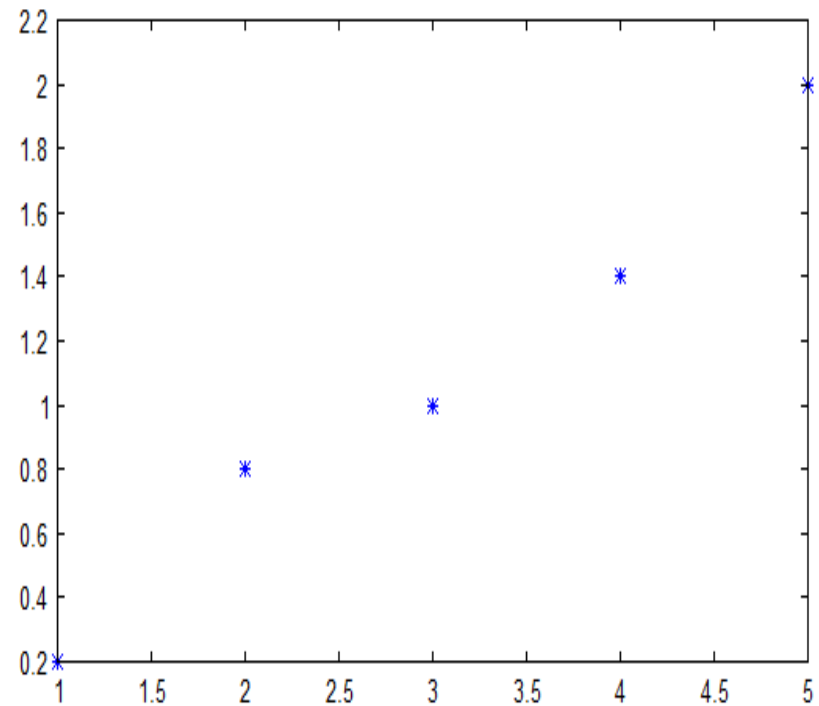
- `>>g=@ (x) a*x+b*y`
- `g =`
- `@(x) a * x + b * y`
- `>> g(2)`
- error: 'y' undefined near line 1 column 15
- `>> y=3`
- `y = 3`
- `>> g(2)`
- error: 'y' undefined near line 1 column 15
- `>> g=@ (x) a*x+b*y`
- `g =`
- `@(x) a * x + b * y`
- `>> g(2)`
- `ans = 22`

Grafice 2D

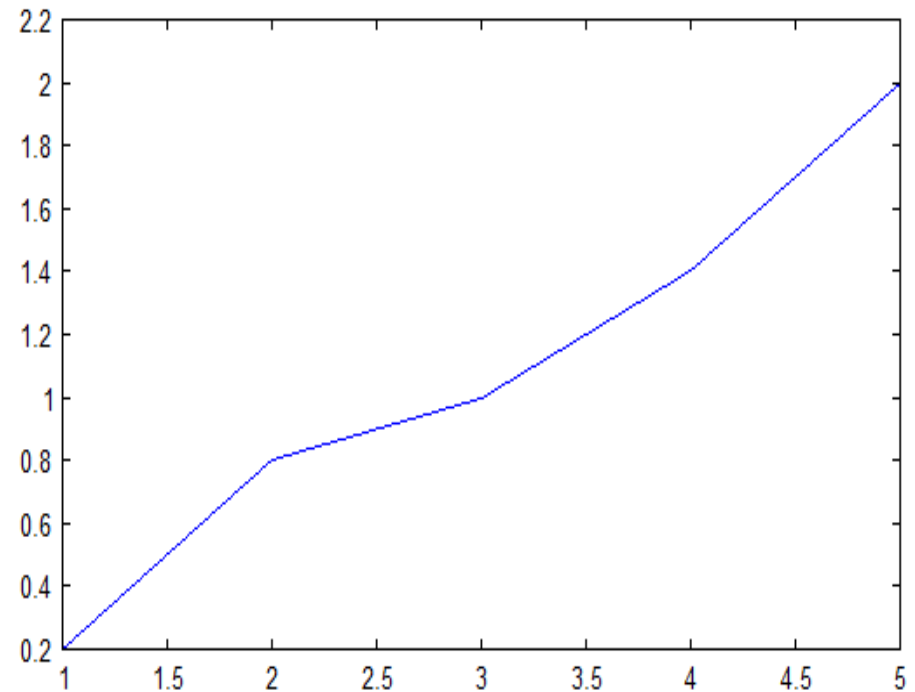
Se dau punctele de coordonate

$(1, 0.2)$, $(2, 0.8)$, $(3, 1)$, $(4, 1.4)$ si $(5, 2)$.

- `>>x=[1 2 3 4 5]`
- `x =`
- `1 2 3 4 5`
- `>> y=[0.2 0.8 1 1.4 2]`
- `y =`
- `0.2000 0.8000`
`1.0000 1.4000 2.0000`
- `>> plot(x,y, '*')`

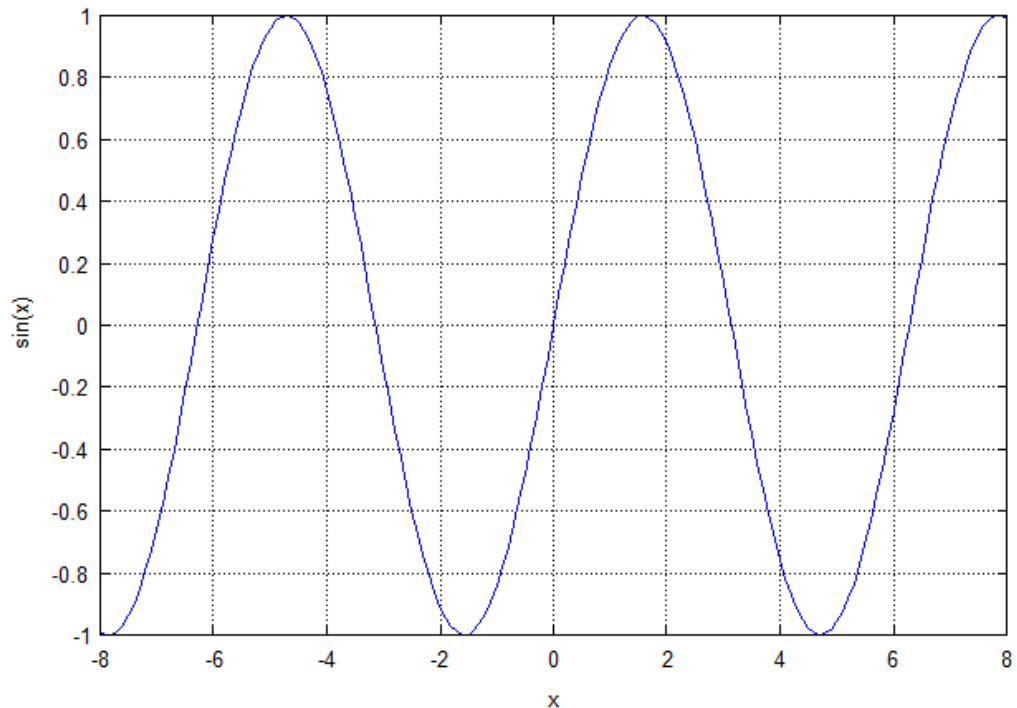


- `>>plot(x,y)`
- va afisa in aceeaasi fereastră punctele unit printr-o linie.
- `>>clf`
- va sterge figura existenta in fereastră de imagini.



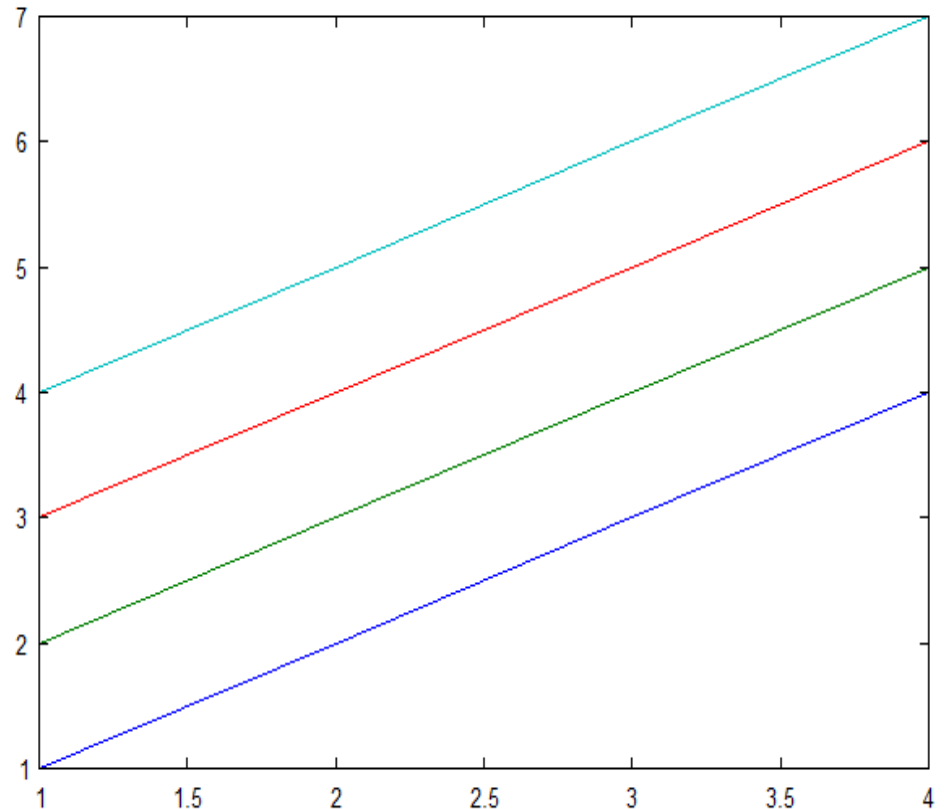
Grafice de functii

- `>>x=-8:0.1:8;`
- `>> plot(x, sin(x))`
- `>> xlabel('x')`
- `>> ylabel('sin(x)')`
- `>> grid on`
-

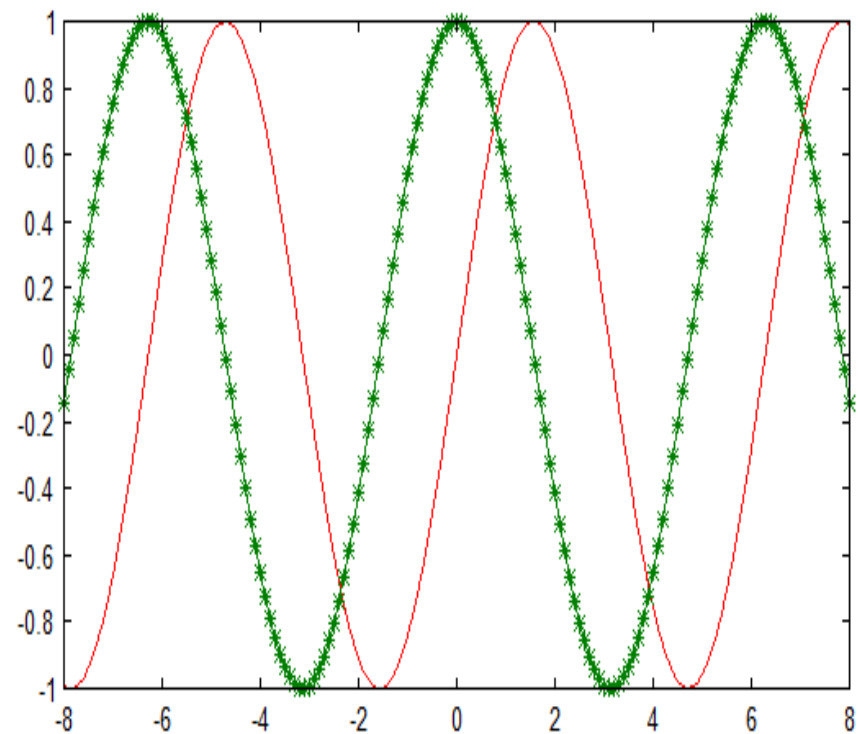


Mai multe functii pe acelasi grafic

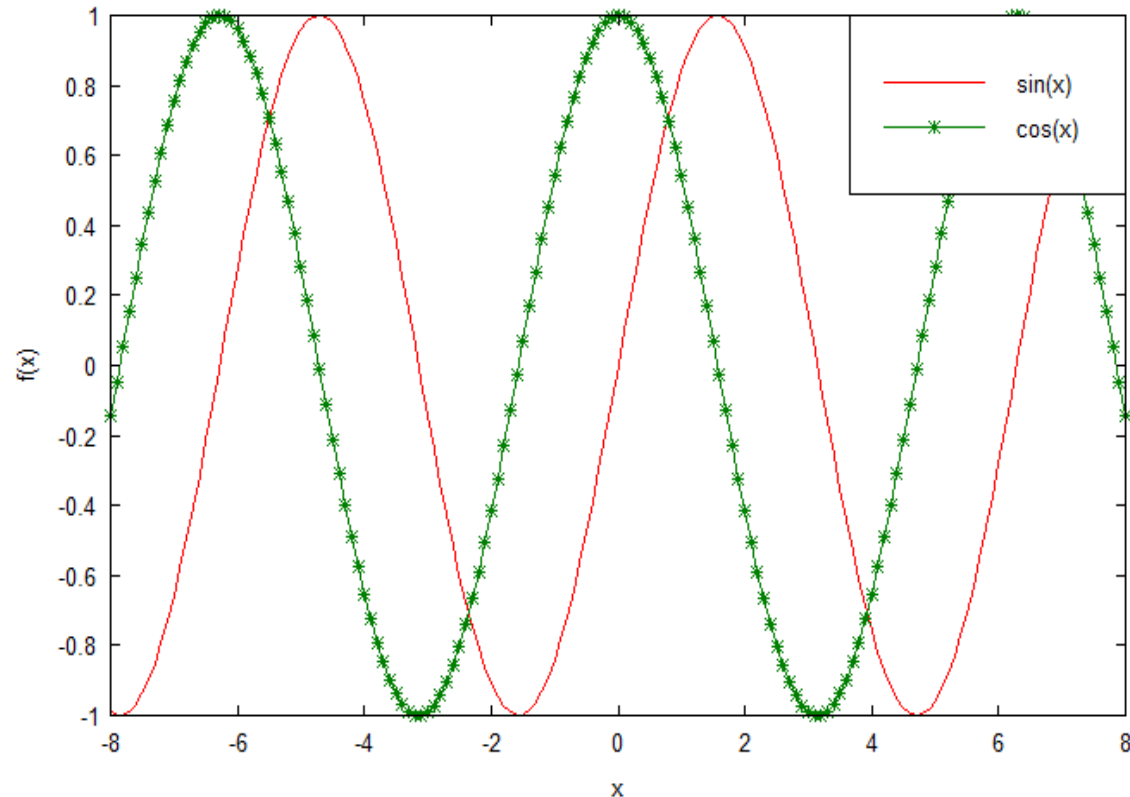
- `>> x=1:4`
- `x =`
- `1 2 3 4`
- `>> y=[x;x+1;x+2;x+3]`
- `y =`
- `1 2 3 4`
- `2 3 4 5`
- `3 4 5 6`
- `4 5 6 7`
- `>> plot(x,y)`



- `plot(x, sin(x), 'r-', x, cos(x), 'g*-')`
- Specificatiile
 - r- inseamna reprezentare cu linie solida, rosie
 - g*- inseamna reprezentare cu linie solida, formata din *, verde



- Putem adauga etichete axelor Ox si Oy precum si legenda care sa descrie cele doua functii reprezentate.
- `>> xlabel('x')`
- `>> ylabel('f(x)')`
- `>>`



Format arguments:

linestyle

'-.' Use dash-dotted lines.

'-' Use solid lines (default).

'.' Use dotted lines.

'--' Use dashed lines.

's' square

'x' cross

marker

'd' diamond

'.' point

'v' downward-facing triangle

'>' right-facing triangle

'<' left-facing triangle

'p' pentagram

'^' upward-facing triangle

'h' hexagram

'o' circle

'*' star

color

'k' black

'y' Yellow

'm' Magenta

'c' Cyan

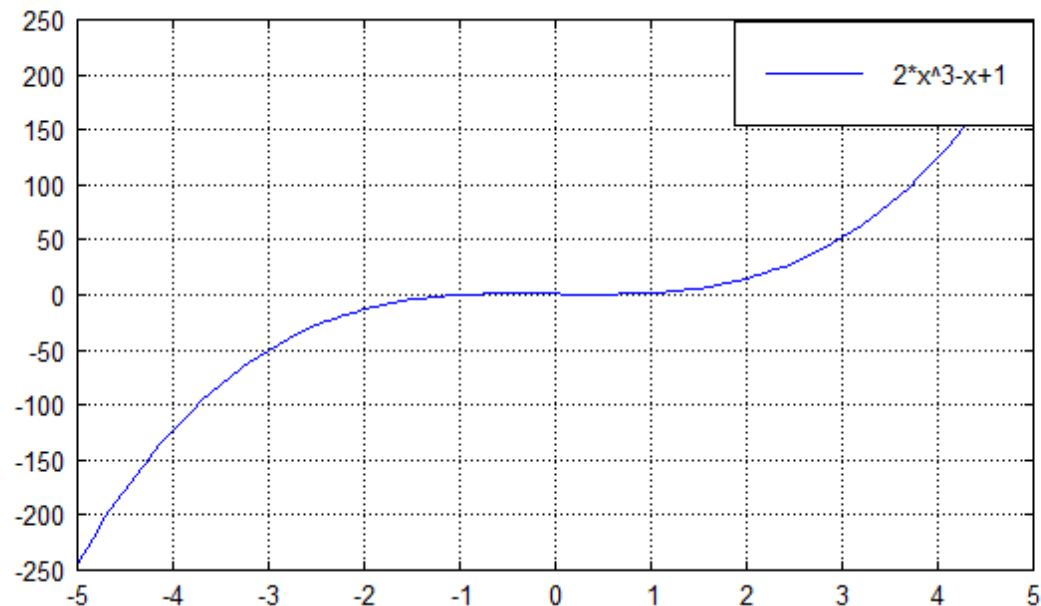
'b' Blue

'w' White

'g' Green

'r' Red

- `>> x=-5:0.1:5;`
- `>> plot(x, 2*x.^3 - x +1)`
- `>> plot(x, 2*x.^3 - x +1), grid on`
- `>> legend ('2*x^3-x+1')`



Exercitii

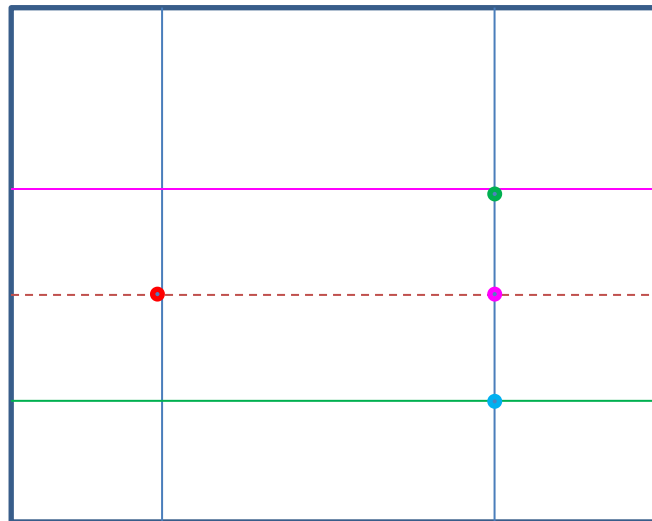
1. Sa se reprezinte grafic funcția $f(x)=x^2$.
Adaugati etichete pentru axe, legenda, titlu.
2. Să se reprezinte pe acelasi grafic funcțiile **$f(x)=\text{tg}(x)$** și **$g(x)=\text{ctg}(x)$** , x ia valori între $[0,2\pi]$. Adaugati etichete pentru axe, legenda, titlu.

Funcțiile matem in MATLAB sunt tan si cot.

3. Sa se reprezinte grafic multimea de puncte:
 $(-1, 2.5)$, $(0, 3)$, $(-2, 5)$, $(2, 4)$ si $(-1, 6)$

Tema 1 lab

- Sa se scrie un script care genereaza urmatorul desen sau un desen asemanator . Se va folosi functia plot.



Programe in MATLAB

- Programele MATLAB au extensia .m.
- Pot fi create cu orice editor de texte sau pot fi create in interfata MATLAB:
- - meniu File/ New/ Script sau File/ New/ Function
- In MATLAB exista doua feluri de programe, functii si scripturi.

Functii

- Din meniu File/New/function, scriem
`function f=func(x)`
`f=2*x.^3-3*x+1;`
- func este un nume dat de noi si trebuie sa coincida cu numele fisierului.
- Salvam cu numele `func.m`
- x - variabila de intrare, iar f –variabila de iesire

- Aceasta functie poate fi apoi folosita la linia de comanda cu numele **func** pentru orice valori numerice, vectori sau matrici.
- ```
>> func(9)
```

```
ans =
```

```
1432
```
- ```
>> v=[1 2 3]
```
- ```
v =
```

```
1 2 3
```
- ```
>> func(v)
```

```
ans =
```

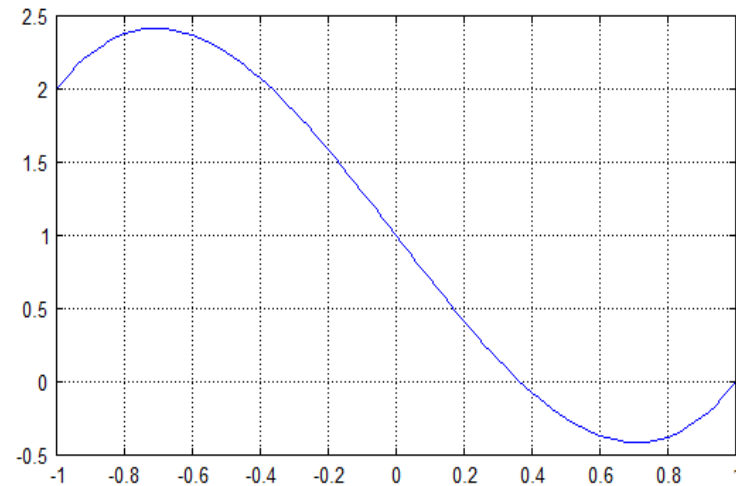
```
0 11 46
```

- Putem afisa si graficul functiei:

```
>> x=-1:0.02:1;
```

```
>> y=func(x);
```

```
>> plot(x,y);grid on
```



Reguli

- Numele variabilei x de la apelare si al variabilei din programul .m nu trebuie sa coincida. Ceea ce trebuie sa coincida este doar numele fisierului .m si numele functiei .
- Toate functiile trebuie sa inceapa cu cuvantul cheie **function**.
- Trebuie sa aiba parametri de intrare si de iesire. In cazul functiei func: x este de intrare si y de iesire.
- Parametrii de iesire, numele functiei si parametrii de iesire trebuie sa apara in prima linie.
- Corpul functiei trebuie sa dea valori parametrilor de iesire.

- Putem avea functii cu mai multi parametrii de intrare si/sau de iesire.
- De ex,
- Functia de mai jos are un param de intrare, x si trei de iesire x2, x3, x4

```
function [x2 x3 x4] = puteri(x)
```

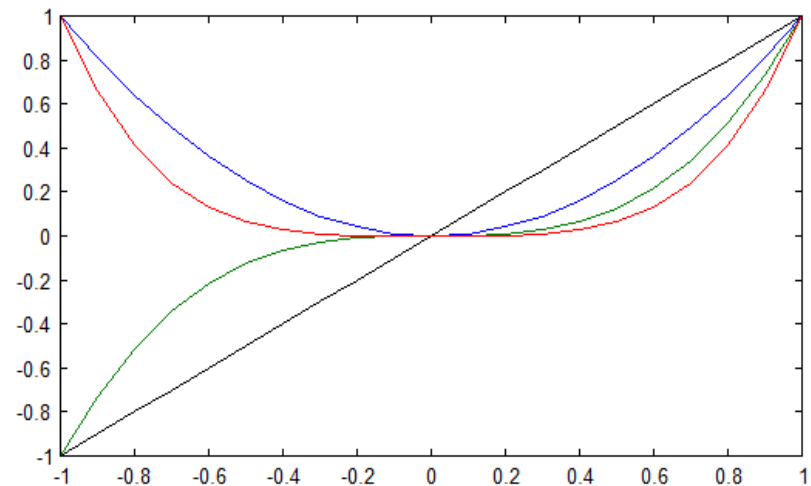
```
x2 = x.^2;
```

```
x3 = x.^3;
```

```
x4 = x.^4;
```

- Salvam fisierul cu numele puteri.m

- `>> x = -1:1:1;`
- `>> [x2 x3 x4] = puteri(x);`
- `>> plot(x, x, 'k', x, x2, 'b', x, x3, 'g', x, x4, 'r')`



Scripturi

- De ex, scrieti urmatorul text si salvati-l cu numele p1.m
 - `>> p1`
 - `A =`
 - `1 2 3`
 - `4 5 6`
 - `b =`
 - `3`
 - `2`
 - `1`
 - `c =`
 - `1 2 3 4 5`
 - `w =`
 - `10`
 - `28`
 - `>> help p1`
 - `%testare matrici si vectori`
- `A=[1 2 3 ; 4 5 6] %matrice A 2x3`
- `b=[3;2;1] %vector 3x1`
- `c=[1 2 3 4 5] %vector linie`
- `w=A*b`
- Rolul comentariilor de la inceputul fisierului .m este sa apara atunci cand comanda **help** urmata de numele fisierului este data la linia de comanda.
- Se va afisa primul bloc de comentarii.