

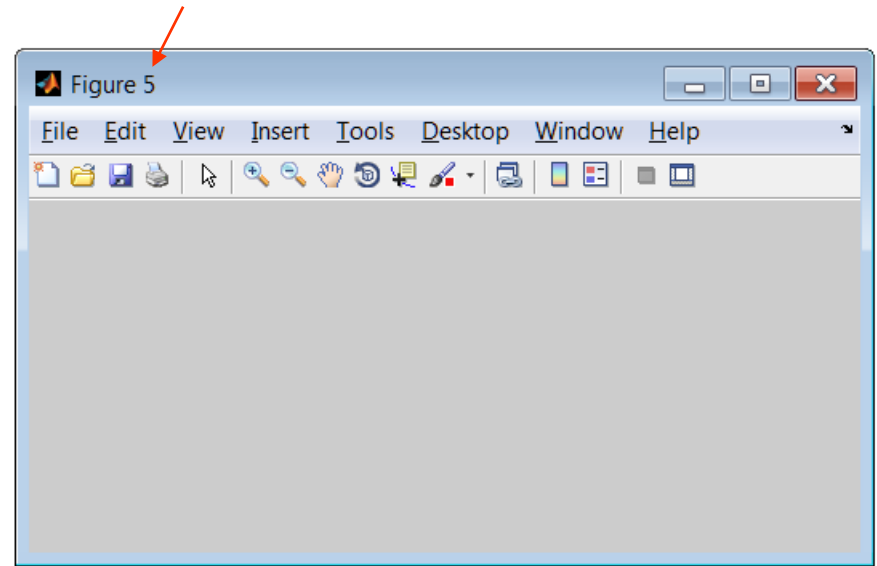
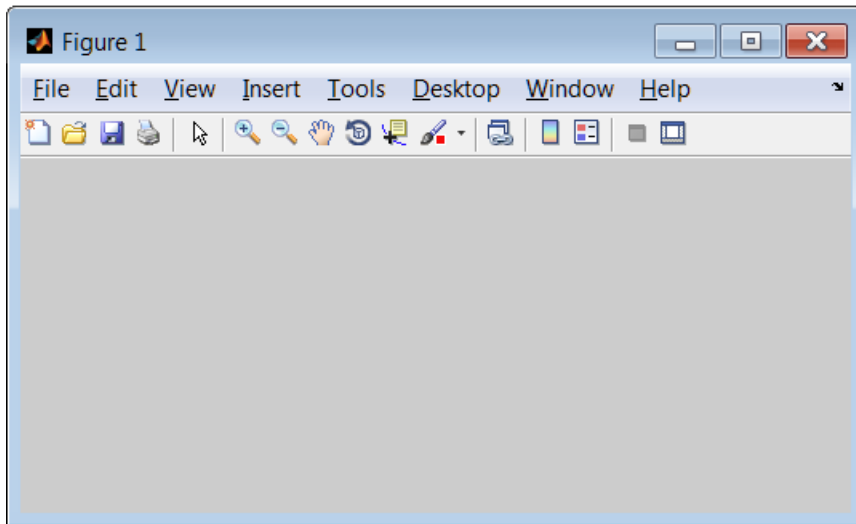


# Osnove računarstva II

**MATLAB - 2D i 3D grafika;**  
Određivanje osnovnih statističkih  
veličina

# Grafički prozori

- Svakom otvorenom grafičkom prozoru se dodjeljuje celobrojni identifikator, počev od broja 1.
- Novi grafički prozor se otvara naredbom **figure** ili **figure(K)**, gde je  $K$  identifikator. Na primer, **figure** i **figure(5)** otvaraju sljedeće prozore:



- Naredba **close** zatvara poslednji aktivni grafički prozor.
- Naredba **close(K)** zatvara grafički prozor sa identifikatorom  $K$ .
- Naredba **close all** zatvara sve otvorene grafičke prozore.

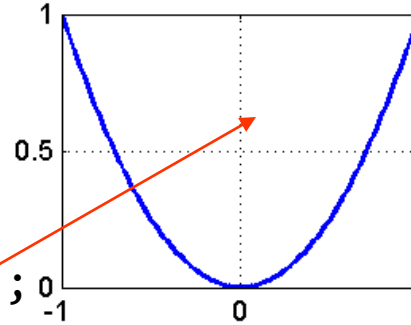
# Podjela grafičkog prozora

- `subplot(x,y,z)` – podjela grafičkog prozora na **x** dijelova po vertikali, **y** po horizontali i pozicioniranje u dijelu **z** za crtanje narednog grafika.

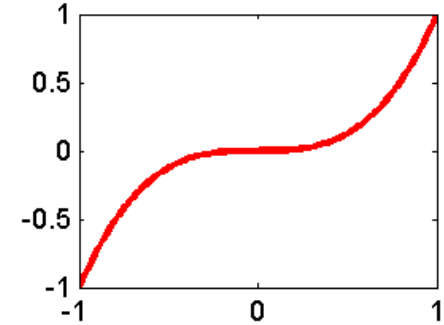
- Primjer:

```
x = linspace(-1,1,199);  
y1 = x .^ 2; y2 = x .^ 3; y3 = exp(x);  
subplot(2,2,1)  
plot(x,y1,'linewidth',2); grid  
subplot(2,2,2)  
plot(x,y2,'color','r','linewidth',3);  
subplot(2,2,3)  
plot(x,y3,'k-.'); grid  
subplot(2,2,4)  
plot(x,y1,'g--',x,y2,'k:',x,y3,'m-');  
legend('x^2','x^3','e^x','Location','Best')  
legend boxoff
```

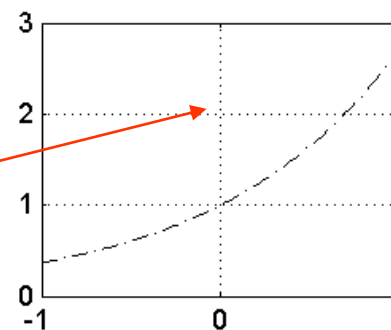
subplot(2,2,1)



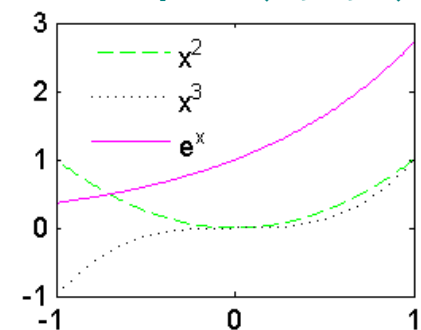
subplot(2,2,2)



subplot(2,2,3)



subplot(2,2,4)



Operator `^` za ispis superscript teksta. Za ispis subscript teksta se koristi `_`

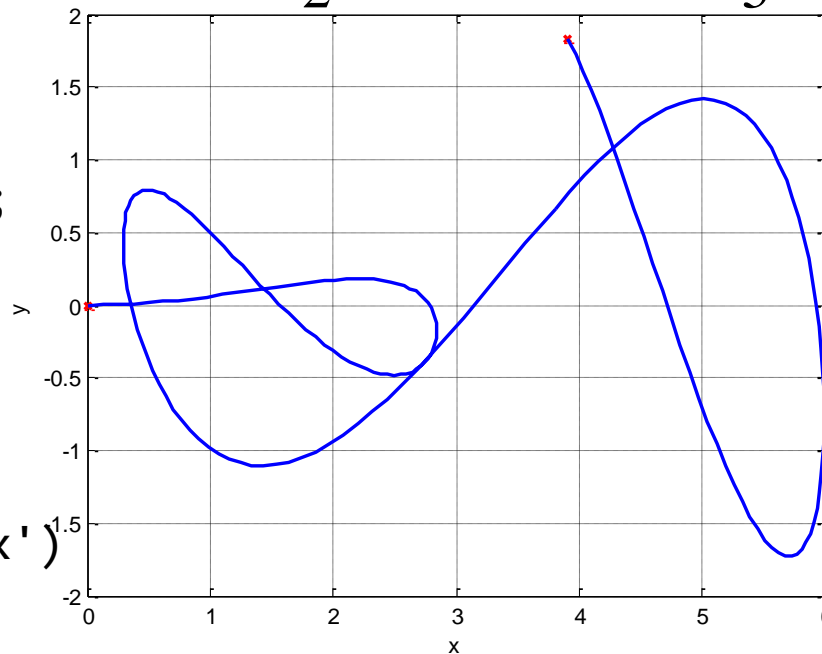
# Crtanje parametarski zadate funkcije

- Nacrtajmo trajektoriju tačke koja se u vremenu od 0 do 10 sekundi kreće po zakonu:

$$x(t) = \frac{t}{2} + 2 \sin t \quad y(t) = \frac{t}{5} \sin 2t$$

- Rješenje:

```
>> t = 0 : 0.05 : 10;  
>> x = t / 2 + 2 * sin(t);  
>> y = t / 5 .* sin(2 * t);  
>> plot(x, y)  
>> axis equal  
>> hold on  
>> plot(x(1), y(1), 'ro')  
>> plot(x(end), y(end), 'rx')  
>> hold off  
>> grid
```



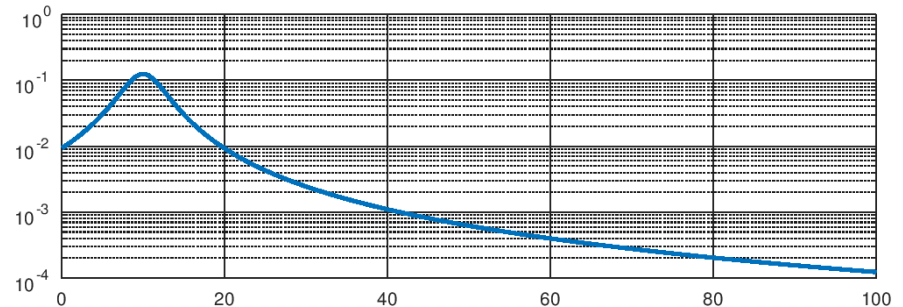
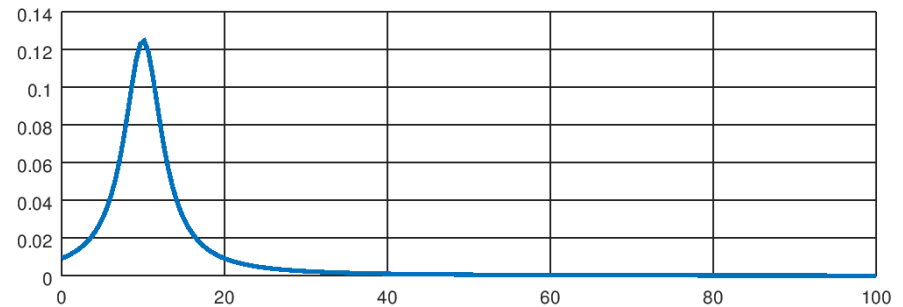
Zadržava postojeći grafik, tako da se narednim funkcijama za crtanje može dodavati novi sadržaj postojećem grafiku.

# Grafici sa logaritamskom podjelom

- Logaritamska raspodjela po y-osi – **semilogy(x,y)**

- Primjer

```
>> x = linspace(0.1, 100);  
>> y = 1 ./ ((x-10) .^ 2 + 8);  
>> subplot(211)  
>> plot(x, y, 'LineWidth',2)  
>> grid  
>> subplot(212)  
>> semilogy(x, y, 'LineWidth',2)  
>> grid
```



# Grafici sa logaritamskom podjelom

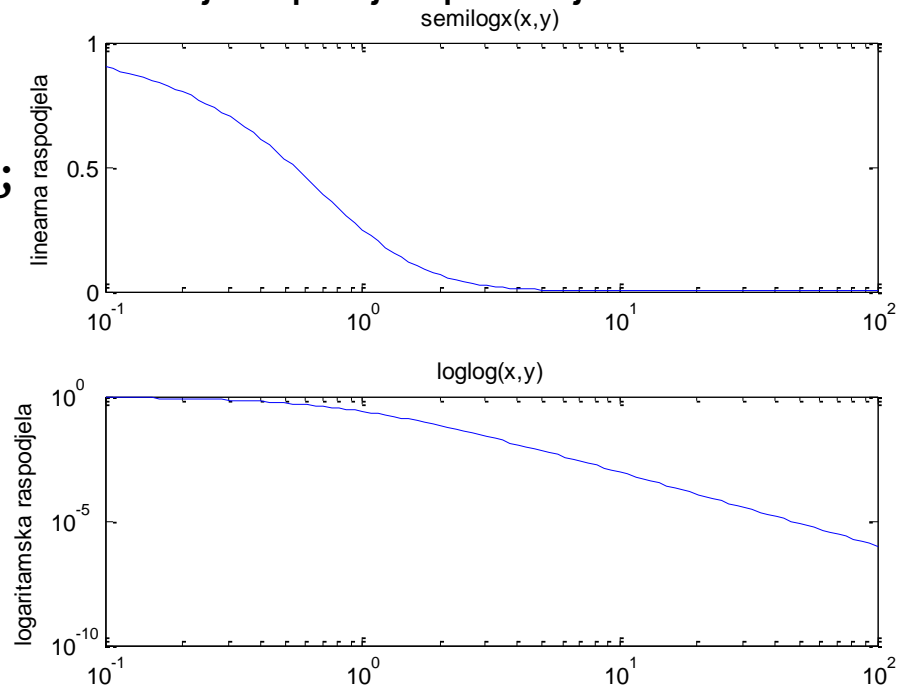
- Komanda: **semilogx(x, Sy)** – logaritamska podjela na x osi.
- Komanda: **loglog(x, y)** – logaritamska podjela na obje ose.
- Ukoliko se koriste prethodno navedene komande, nezavisno promjenljiva se mora zadati komandom: **logspace(e1, e2, N)**.
- **logspace(e1, e2, N)** – generiše niz od N vrijednosti koje su na logaritamskoj osi ravnomjerno raspoređene od  $10^{e1}$  do  $10^{e2}$ .
- Ukoliko se ne zada N, podrazumijevano se generiše 50 tačaka
- **Koordinata čija osa ima logaritamsku podjelu mora biti pozitivna!!!.**

# Primjer

Nacrtati grafik funkcije  $y = \frac{1}{x^3 + x^2 + x + 1}$  za  $0.1 \leq x \leq 100$

a) U logaritamskoj raspodjeli po x-osi, b) u logaritamskoj raspodjeli po obje ose.

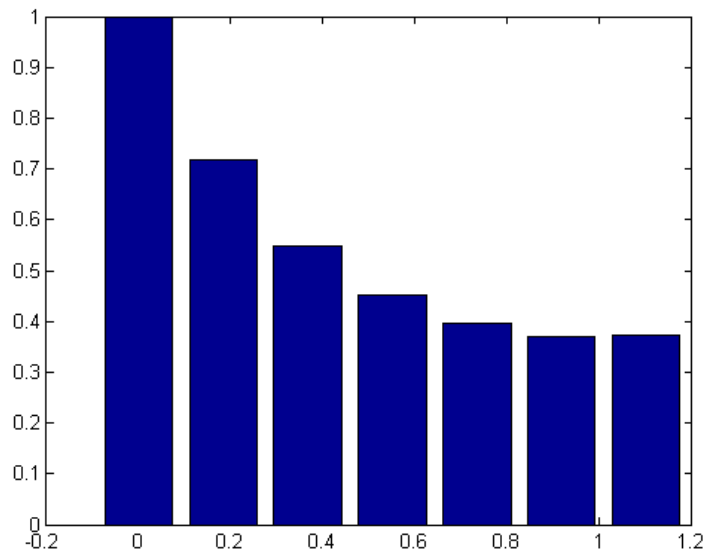
```
>> x = logspace(-1, 2, 100);  
>> y = 1 ./ (x.^ 3 + x.^ 2 + x + 1);  
>> subplot(211)  
>> semilogx(x,y);  
>> title('semilogx(x,y)')  
>> ylabel('linearna raspodjela')  
>> subplot(212)  
>> loglog(x,y)  
>> title('loglog(x,y)')  
>> ylabel('logaritamska raspodjela')
```



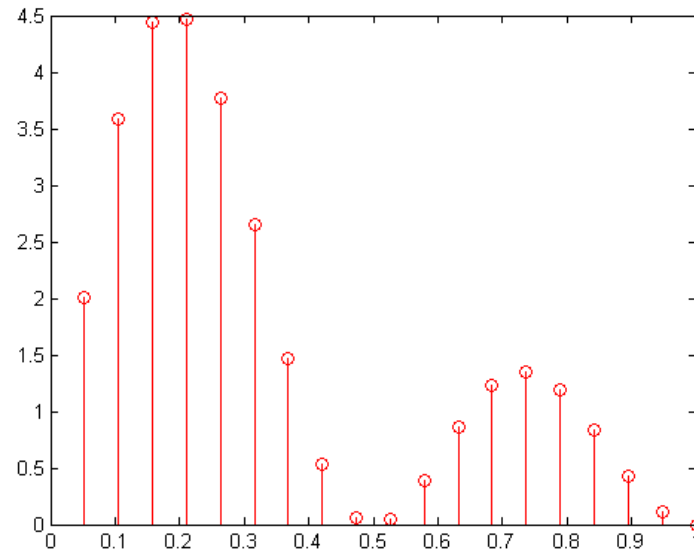
# Stepenasti grafici

- Koriste se za prikaz diskretnih veličina
- **bar(x)** – crtanje bar grafika
- **stem(x)** – crtanje “stem” grafika
- Nemaju mogućnost prikazivanja više funkcija na istom grafiku

```
>> close  
>> x = linspace(0, 1.1, 7);  
>> y = exp(x.^2-2*x);  
>> bar(x,y)
```



```
>> x = linspace(0, 1, 20);  
>> y = sin(2*pi*x).^2./x;  
>> stem(x,y,'r')
```





# Grafik matrice

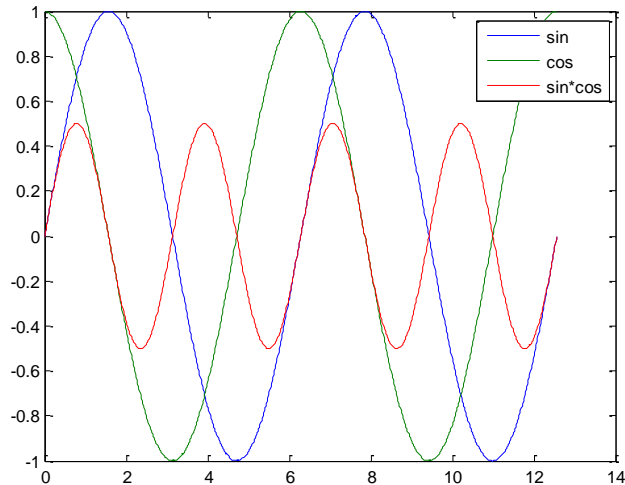
- Ukoliko je u `plot(x,y)` varijabla `y` matrica, tada ona mora imati onoliko vrsta kolika je dužina vektora `x`. Svaka kolona matrice `y` se crta kao posebna linija na grafiku, za koju `x` predstavlja nezavisno promjenljivu.

```
x=(0 : pi/180 : 4 * pi)';
```

```
y=[sin(x), cos(x), sin(x).*cos(x)];
```

```
plot(x, y)
```

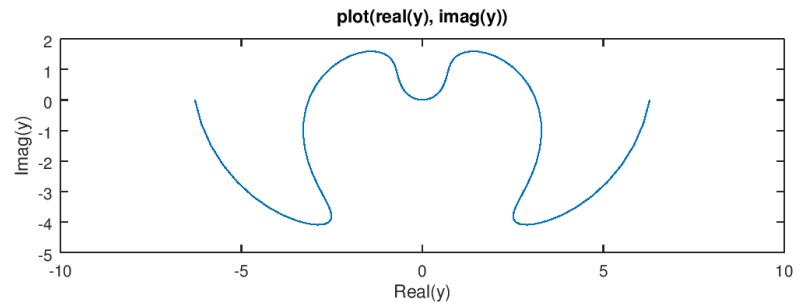
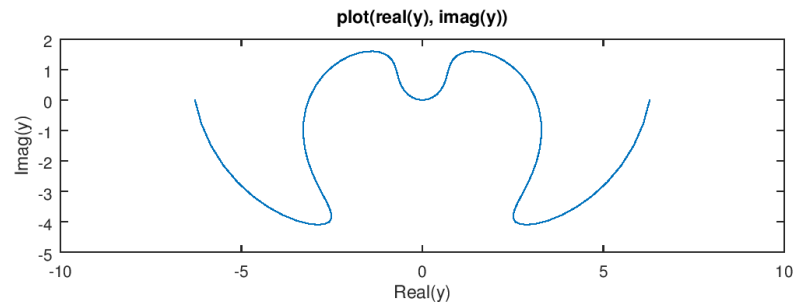
```
legend('sin', 'cos', 'sin*cos')
```



# Grafik kompleksnih funkcija

- Ukoliko vektor  $y$  ima kompleksne vrijednosti, `plot(y)` crta zavisnost realnog od imaginarnog dijela. Isti rezultat se dobija sa `plot(real(y), imag(y))`.

```
>> x=linspace(-2*pi,2*pi);  
>> y=x .* exp(j * sin(x));  
>> plot(y)  
>> xlabel('Real(y)')  
>> ylabel('Imag(y)')  
>> subplot(212)  
>> plot(real(y), imag(y))  
>> xlabel('Real(y)')  
>> ylabel('Imag(y)')  
>> title('plot(real(y), imag(y))')  
>> subplot(211)  
>> title('plot(real(y), imag(y))')
```



Naslov, mreža, oznake osa se mogu dodavati i naknadno grafiku, samo se pristupi željenom dijelu grafičkog prozora preko `subplot(...)`

# Snimanje grafika

- Komanda: `print [ime] [-drajver]` snima grafički fajl pod nazivom ime u formatu koji je odabran specifikacijom drajver.
- Snimljeni fajl možemo uključiti u naše dokumente (MS Word, Libre Office Writer, LaTeX...).
- Često korišćeni, grafički formati (drajveri) su:
  - `-deps2`, `-depvc2`, `-dpng`, `-djpeg`, `-dtiff`, `-dmeta`.
- Ukoliko želimo snimiti grafik u folderu u kojem se ne nalazimo trenutno, moramo navesti kompletnu putanju do željenog foldera, npr: `print C:\Temp\graf1 -djpeg` snima fajl pod imenom *graf1* u folderu *Temp* koji se nalazi na disku C. Fajl se snima u formatu JPEG.

# 3D grafika

- Predstaviti 3D liniju ili površ koristeći se dvodimenzionom površinom nije jednostavan zadatak. Mora se na neki način izvršiti projekcija 3D objekta na ravan (ravan ekrana ili papira).
- Posmatračemo dva tipa zadataka: crtanje linija u tri dimenzije i crtanje površi u tri dimenzije.
- Linija se crta kao niz povezanih tačaka pri čemu je potrebno za svaku tačku zadati tri koordinate  $(x, y, z)$ .
- Površ se crta preko elementarnih površi (oblika trougla ili četvorougla). Često se mora voditi računa da li je površ „vidljiva“ ili je zaklonjena nekom drugom površi.

# 3D grafika

- Komanda `plot3(x,y,z)` crta liniju u 3D. Nizovi `x`, `y` i `z` imaju jednake dužine i predstavljaju koordinate tačaka koje čine liniju.
- Grafiku se mogu dodati oznake standardnim komandama `xlabel`, `ylabel`, `title`, `grid`. Dodaje se funkcija `zlabel`, a funkciji `text` prosljeđujemo 4 argumenta, tri koordinate i tekst koji specificira boju i tip linije i markera.
- Komandom `view(azimut,elevacija)` se podešava „pogled“ na 3D grafik. Parametri *azimut* i *elevacija* su uglovi u stepenima.
- Komandi `axis` kojom se određuje koji će dio grafika biti vidljiv, se u ovom slučaju prosljeđuje niz od 6 vrijednosti

`axis([xmin,xmax,ymin,ymax,zmin, zmax]).`

# 3D grafika – primjer linijskog grafika

■ Neka je 3D funkcija zadana kao  $x = t \sin t$ ,  $y = t \cos t$ ,  $z = t$  za  $0 \leq t \leq 10\pi$

```
>> t = 0 : 0.01 : 10*pi;
```

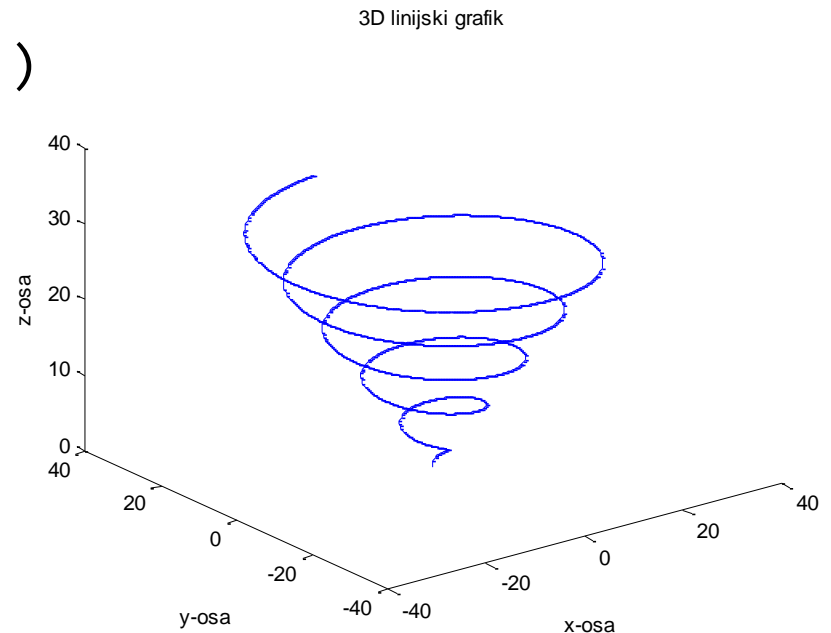
```
>> plot3(t .* sin(t), t .* cos(t), t)
```

```
>> xlabel('x-osa')
```

```
>> ylabel('y-osa')
```

```
>> zlabel('z-osa')
```

```
>> title('3D linijski grafik')
```

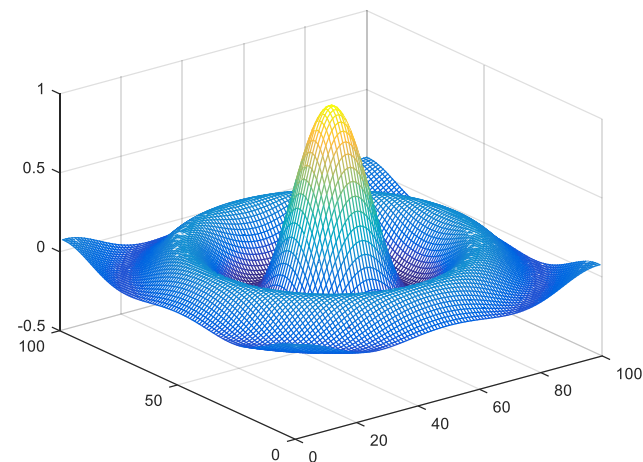


# 3D grafika

- Kod 3D grafika, prvo se nezavisno promjenljive zadaju sa  
**`[X Y] = meshgrid(x1, y1)`**
- čime se kreiraju dvodimenzione matrice  $X$  i  $Y$ , čiji su elementi određeni vektorima  $x1$  i  $y1$ . Ovo će biti „domen“ naše površi. Za svaku tačku domena izračunaju se vrijednosti funkcije  **$Z=f(X, Y)$** .
- Za crtanje 3D grafika se koriste fukcije:  
**`mesh(z)`** (mrežasta površina), **`surf(z)`** (boje se elementarne površine),  
**`imagesc(z)`** (bojom predstavlja vrijednost z-koordinate),  
**`contour(X, Y, Z)`** crta konturne linije površi.

## Primjer:

```
>> x = linspace(-10, 10);  
>> y = x;  
>> [X, Y] = meshgrid(x, y);  
>> P = sqrt(X .^ 2 + Y .^ 2);  
>> Q = sin(P) ./ P;  
>> mesh(Q);
```



# 3D grafika – funkcije mesh i imagesc,F

Primjer ...

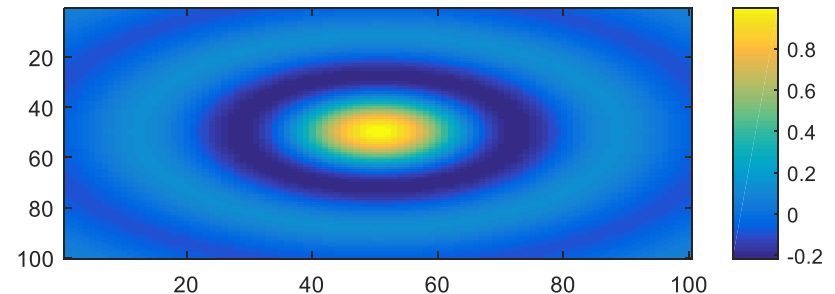
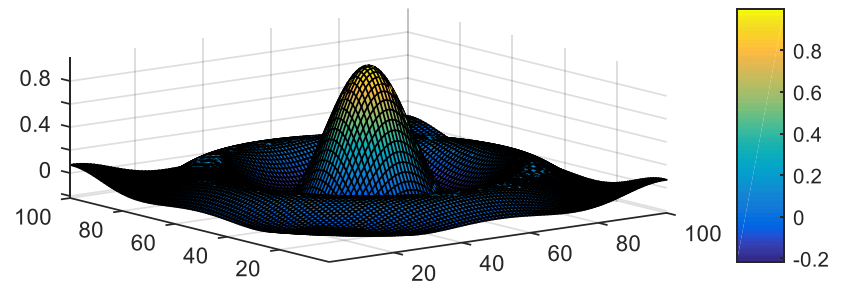
```
figure(2)
subplot(211)
surf(Q), colorbar
axis tight
subplot(212)
imagesc(Q), colorbar
```

Granice osa na grafiku izjednačiti sa opsegom podataka u XY ravni.

Komanda **colorbar** na grafik dodaje objašnjenje numeričkih vrijednosti boja, dok **colormap(paleta)** bira paletu boja.

Često korišćene palete su jet, hot, gray,...

Na podrazumijevanu paletu boja se vraćamo komandom **colormap('default')**



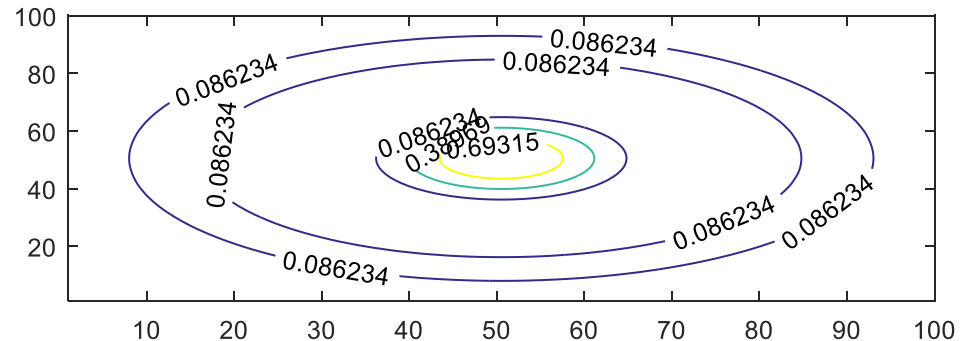
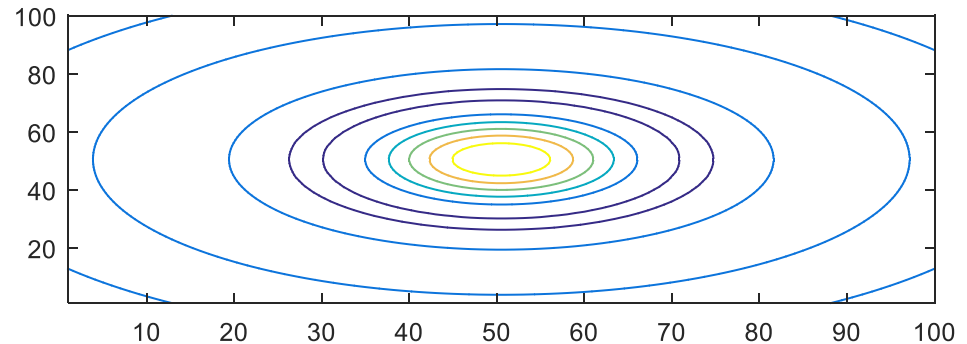


# 3D grafika – funkcije mesh i imagesc,F

Primjer ...

```
figure(3)
subplot(211)
contour(Q)
subplot(212)
[c, h] = contour(Q, 3);
clabel(c, h)
```

[c, h] izlazne vrijednosti funkcije **contour** se mogu proslijediti komandi **clabel** za obilježavanje iscrtanih konturnih linija.



Nakon navođenja funkcije koja se želi nacrtati konturnim linijama funkciji **contour** se može proslijediti broj konturnih ravni ili vrijednosti funkcije u kojima se žele crtati konture.

# Sabiranje elemenata matrice

- Funkcija **sum(a)** izvršava sabiranje elemenata matrice po kolonama.

- Primjer:

```
>> a=[1 5 6; 4 7 8];
```

```
>> sum(a)
```

```
ans =
```

```
5 12 14
```

```
>> sum(a')
```

```
ans =
```

```
12 19
```

$$a = \begin{bmatrix} 1 & 5 & 6 \\ 4 & 7 & 8 \end{bmatrix}$$

$$a' = \begin{bmatrix} 1 & 4 \\ 5 & 7 \\ 6 & 8 \end{bmatrix}$$

# Sabiranje elemenata matrice

- Zbir svih elemenata matrice  $a$  se dobija pozivanjem funkcije `sum` dva puta, odnosno traženjem sume elemenata niza koji se dobija kao rezultat funkcije `sum(a)`:

```
>> a = [1 5 6; 4 7 8];
```

```
>> b = sum(a)
```

```
b =
```

```
    5  12  14
```

```
>> sum(b)
```

```
ans =
```

```
    31
```

```
>> sum(sum(a))
```

```
ans =
```

```
    31
```

$$a = \begin{bmatrix} 1 & 5 & 6 \\ 4 & 7 & 8 \end{bmatrix}$$

`cumsum(a)` – kumulativna suma. Svaka kolona se dobija kao niz parcijalnih suma elemenata iste kolone matrice  $a$ .

```
>> cumsum(a)
```

```
ans =
```

```
    1    5    6  
    5   12   14
```

# Množenje elemenata matrice

- Funkcija **prod(a)** vrši množenje elemenata matrice po kolonama.

- Primjer:

```
>> a=[1 5 6; 4 7 8];
```

```
>> prod(a)
```

```
ans =
```

```
4 35 48
```

```
>> prod(a')
```

```
ans =
```

```
30 224
```

$$a = \begin{bmatrix} 1 & 5 & 6 \\ 4 & 7 & 8 \end{bmatrix}$$

$$a' = \begin{bmatrix} 1 & 4 \\ 5 & 7 \\ 6 & 8 \end{bmatrix}$$

# Množenje elemenata matrice

- Proizvod svih elemenata matrice  $a$  se dobija pozivanjem funkcije `prod` dva puta, odnosno traženjem proizvoda elemenata niza koji se dobija kao rezultat funkcije **`prod(a)`**:

```
>> a = [1 5 6; 4 7 8];
```

```
>> b = prod(a)
```

```
b =  
    4 35 48
```

```
>> prod(prod(b))
```

```
ans =  
    6720
```

- >> **`cumprod(a)`** – kumulativni proizvod. Svaka kolona se dobija kao niz parcijalnih proizvoda elemenata iste kolone matrice  $a$ .

```
ans =  
    1    5    6  
    4   35   48
```

$$a = \begin{bmatrix} 1 & 5 & 6 \\ 4 & 7 & 8 \end{bmatrix}$$

# Minimum i maksimum matrice

- Za traženje minimuma i maksimuma matrice, koriste se funkcije **min** i **max**, respektivno. Obje funkcije rade po kolonama.

```
>> a=[4 5 6; 1 7 8];
```

```
>> min(a)
```

```
ans =
```

```
    1    5    6
```

```
>> min(min(a))
```

```
ans =
```

```
    1
```

```
>> max(a)
```

```
ans =
```

```
    4    7    8
```

```
>> max(max(a))
```

```
ans =
```

```
    8
```

$$a = \begin{bmatrix} 4 & 5 & 6 \\ 1 & 7 & 8 \end{bmatrix}$$

# Minimum i maksimum sa pozicijom

- Ukoliko se funkcije `min` i `max` pozovu sa dva izlazna argumenta, dobija se i pozicija minimuma i maksimuma.

```
>> [maks, pozMaks] = max(a)
```

```
maks =
```

```
    4    7    8
```

```
pozMaks =
```

```
    1    2    2
```

```
>> [min, pozMin] = min(a)
```

```
min =
```

```
    1    5    6
```

```
pozMin =
```

```
    2    1    1
```

$$a = \begin{bmatrix} 4 & 5 & 6 \\ 1 & 7 & 8 \end{bmatrix}$$

- Razmisliti kako dobiti minimum i maksimum i čitave matrice. Kako bi se dobila pozicija minimuma i maksimuma čitave matrice?

## Srednja vrijednost, medijan i sortiranje

- Funkcija **mean(X)** daje srednju vrijednost elemenata niza X.
- Funkcija **median(X)** daje „srednji“ element niza X. Ukoliko niz X ima neparan broj elemenata sortiraće elemente u neopadajući poredak i vratiće element u sredini dobijenog niza, a ukoliko ima paran broj elemenata, rezultat je aritmetička sredina elemenata  $X(N/2)$  i  $X(N/2+1)$ , sortiranog niza X.
- Funkcija **sort(X)** vraća elemente niza X sortirane u neopadajući poredak. Ukoliko od ove funkcije tražimo dvije izlazne vrijednosti **[S, p] = sort(X)** tada se u varijablu p upisuju pozicije sortiranih elemenata u originalnom nizu.
- Ako navedenim funkcijama prosljedimo matricu, one navedenu operaciju izvode za svaku kolonu matrice i vraćaju niz vrijednosti.



## Srednja vrijednost, medijan i sortiranje

```
>> a=[4 5 6; 1 7 8; 2 1 3];
```

```
>> mean(a)
```

```
ans =
```

```
2.3333 4.3333 5.6667
```

```
>> median(a)
```

```
ans =
```

```
2 5 6
```

```
>> [a1, n] = sort(a)
```

```
a1 =
```

```
1 1 3
```

```
2 5 6
```

```
4 7 8
```

```
n =
```

```
2 3 3
```

```
3 1 1
```

```
1 2 2
```

$$a = \begin{bmatrix} 4 & 5 & 6 \\ 1 & 7 & 8 \\ 2 & 1 & 3 \end{bmatrix}$$

## Razlika susjednih elemenata

- Funkcija **D = diff(X)** vraća niz sastavljen od razlika susjednih elemenata  $D_n = X_{n+1} - X_n$ . Dobijeni niz ima jedan element manje od polaznog.

### Primjer

```
>> a=[4 5 6; 1 7 8; 2, 1, 3];
```

```
>> d = diff(a)
```

```
d =
```

```
    -3     2     2
```

```
     1    -6    -5
```

```
>> [m, n] = size(d)
```

```
m = 2
```

```
n = 3
```

$$a = \begin{bmatrix} 4 & 5 & 6 \\ 1 & 7 & 8 \\ 2 & 1 & 3 \end{bmatrix}$$

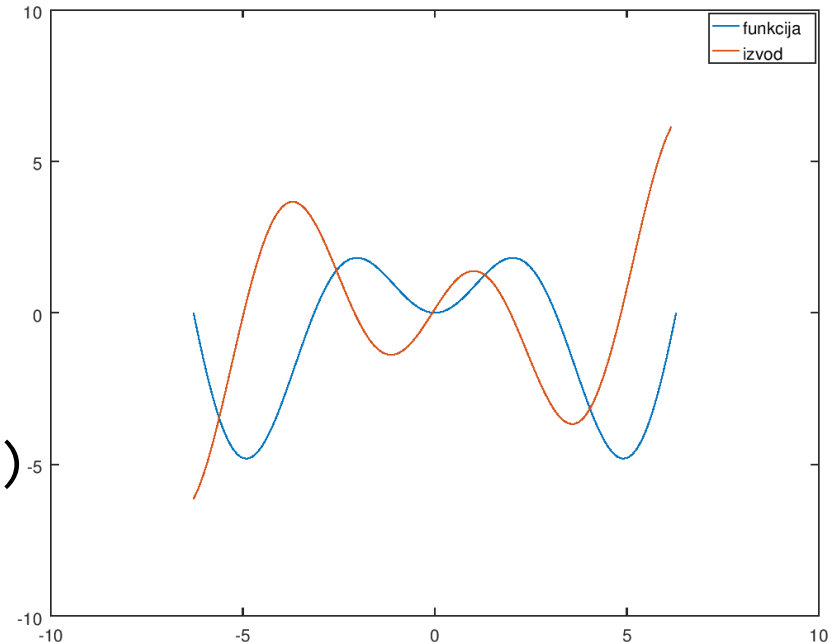
# Numeričko izračunavanje izvoda

- Izvod neke funkcije  $f(x)$  može se približno izračunati kao:

$$f'(x) = \frac{df(x)}{dx} \approx \frac{f(x + \Delta_x) - f(x)}{\Delta_x}$$

**Primjer:** Za funkciju  $x = x\sin(x)$  izračunati i nacrtati prvi izvod u intervalu  $-2\pi \leq x \leq 2\pi$

```
>> x = linspace(-2*pi, 2*pi);  
>> y = x .* sin(x);  
>> y1 = diff(y) ./ diff(x);  
>> plot(x, y, x(1 : end -1), y1)  
>> legend('funkcija', 'izvod')
```



# Numeričko izračunavanje određenog integrala

- Određeni integral neke funkcije  $f(x)$  može se izračunati korišćenjem pravougaonog pravila:

$$\int_a^b f(x) dx \approx \sum_{i=1}^{N-1} f(x_i) \Delta x_i$$

gdje je interval  $(a, b)$  podijeljen na  $N$  jednakih podintervala

Primjer: Za funkciju  $y = x \sin(x)$  izračunati integral od  $-2\pi \leq x \leq 2\pi$

```
>> x = linspace(-2*pi, 2*pi);
```

```
>> y = x .* sin(x);
```

```
>> dx = x(2) - x(1);
```

```
>> I = sum(y(1 : end -1) * dx)
```

```
I = -12.549
```

# Numeričko izračunavanje neodređenog integrala

- Numeričko izračunavanje neodređenog integrala u nekom intervalu se svodi na računanje određenog integrala sa promjenljivom gornjom granicom  $\int_a^x f(x)dx$ , gdje je  $x$  u intervalu  $(a, b)$

Primjer: Za funkciju  $x = x\sin(x)$  izračunati neodređeni integral u intervalu  $-2\pi \leq x \leq 2\pi$

```
>> x = linspace(-2*pi, 2*pi);  
>> y = x .* sin(x);  
>> dx = x(2) - x(1);  
>> I = cumsum(y(1 : end -1)) * dx;  
>> plot(x(1 : end -1) + dt, I)
```

