

**UNIVERZITET U NOVOM SADU**  
**FAKULTET TEHNIČKIH NAUKA**  
**KATEDRA ZA AUTOMATIKU I UPRAVLJANJE SISTEMIMA**

# Matlab - Grafika i vizuelizacija podataka

Modeliranje i simulacija sistema

Upravljanje, modelovanje i simulacija sistema

# O MATLAB-ovoj grafici

- MATLAB poseduje mnoštvo tehnika za predstavljanje i vizuelizaciju podataka
- postoje funkcije i visokog i niskog nivoa pristupa
- grafički sistem se sastoji od kolekcije *grafičkih objekata*, kao što su linije i površi, čiji prikaz zavisi od mnoštva parametara *osobina objekata*
- Grupe funkcija
  - 2-D grafika
  - 3-D grafika
  - funkcije opšte namene
  - upravljanje bojama
  - rukovanje objektima

# 2-D grafika

- elementarni X-Y dijagrami
- posebni X-Y dijagrami
- ulepšavanje dijagrama
  
- prevođenje koordinata između Dekartovih i polarnih
  - **cart2pol**
  - **pol2cart**
  
- uvećanje dela dijagrama (upotrebom miša)
  - **zoom**

# Elementarni X-Y dijagrami

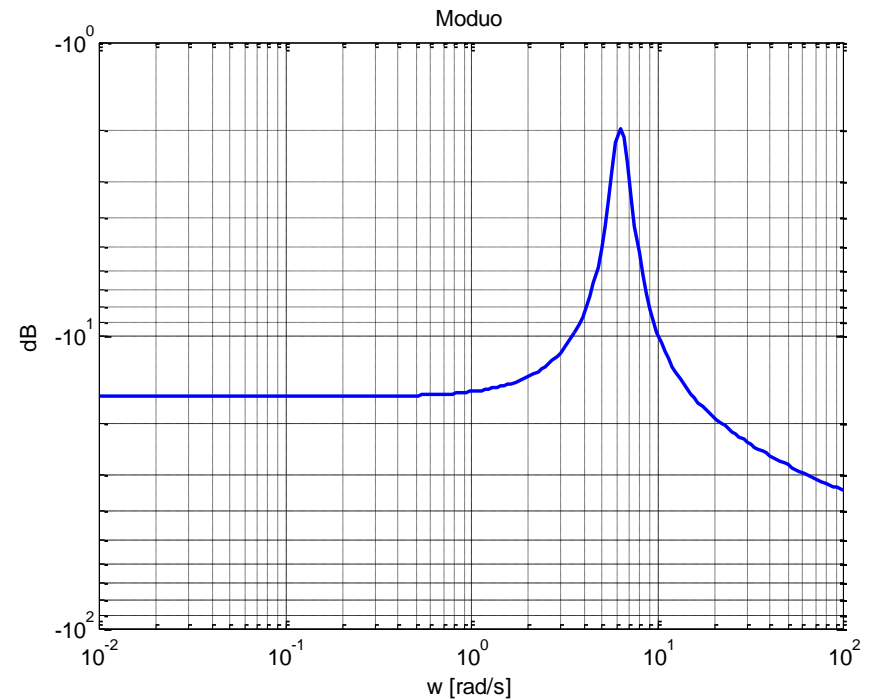
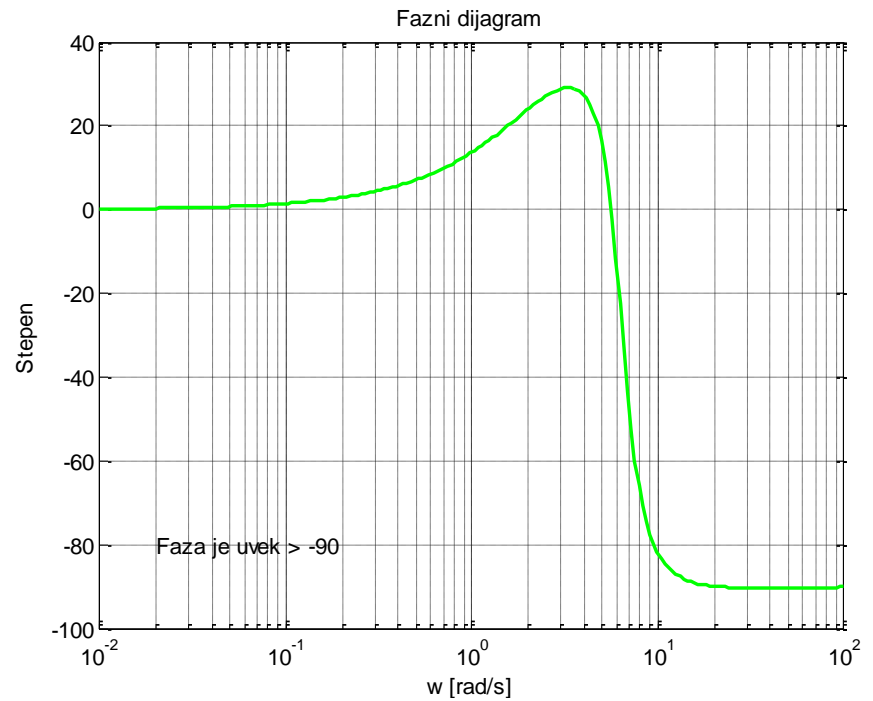
- crtaju dijagrame na osnovu pojedinačnih tačaka (povezanih)
- tačke se zadaju u obliku vektora ili matrica
- ose se automatski skaliraju
- postoji više skala za ose, te se razlikuju f-je za crtanje:
  - **plot** - crtanje dijagrama sa linearnom podelom na obe ose
  - **loglog** - logaritamska podela po obe ose
  - **semilogx** - apscisa logaritamska, ordinata linearna
  - **semilogy** - apscisa linearna, ordinata logaritamska
- dekoracija
  - **title** - naslov dijagrama
  - **xlabel, ylabel** - slovne oznake apscise i ordinate
  - **text, gtext** - ispis teksta na proizvoljnom mestu
  - **grid** - prikaz mrežice
  - **legend** - prikaz legende

# Primeri X-Y dijagrama

- *crtanje Bode-ovih krivih*  
w=logspace(-2,2,200);  
[m,f]=bode([10 31],[5 14 200],w);  
semilogx(w,f,'g')  
title('Fazni dijagram'), xlabel('w [rad/s]'), ylabel('Stepen'), grid  
text(0.02,-80,'Faza je uvek > -90')  
pause  
loglog(w,20\*log10(m))  
title('Moduo'), xlabel('w [rad/s]'), ylabel('dB'), grid
- *crtanje krive koja nije “matematička funkcija”*  
t=0:pi/50:2\*pi;  
x=sin(t); y=sin(t+0.25);  
plot(x,y,'c-',x,y,'m+'), title('Fazni pomeraj')

# Primer

```
w=logspace(-2,2,200);  
[m,f]=bode([10 31],[5 14 200],w);  
semilogx(w,f,'g','linewidth',2)  
title('Fazni dijagram')  
xlabel('w [rad/s]')  
ylabel('Stepen')  
grid  
text(0.02,-80,'Faza je uvek > -90')  
pause  
  
loglog(w,20*log10(m),'linewidth',2)  
title('Moduo'), xlabel('w [rad/s]'), ylabel('dB'), grid
```



# Stilovi linija, boje i markeri na X-Y dijagramu

- stil linije ili tip markera i boja se zadaju kao string S u komandi oblika **plot(x,y,S)**
- boje: yellow, magenta, cyan, red, green, blue, white, black
- tip markera: “.”, “o”, “x”, “+”, “\*”
- stil linije: “-” puna, “:” tačkasta, “-.” crtatačka, “--” crtice
- podrazumevani stil linije je “puna”, a boje se dodeljuju prema gornjem spisku (u zavisnosti od rednog broja linije)
- kod štampanja (crno-belog štampača) umesto razlikovanja linija prema boji, automatski se postavljaju različiti stilovi linija

# Prikaz više krivih na X-Y dijagramu

- zadavanjem dvojki parametara (trojki, ako postoji dekoracija) krive ne moraju biti poznate u jednakom broju tačaka

```
x1=0:0.1:20;          y1=sin(x1);  
x2=-10:0.25:30;      y2=cos(x2/4);  
plot(x1,y1,x2,y2.*y2), legend('sin(t)', 'cos(t/4)^2')
```

- zadavanjem nekoliko krivih kao kolone matrice krive moraju biti poznate u istim tačkama

```
x=(-2:0.01:2)'; plot(x,[x x.*x x.^3])
```

- naknadno dodavanje krivih (ose su “zamrzute”)  
boje krivih su iste ako se ne zadaju

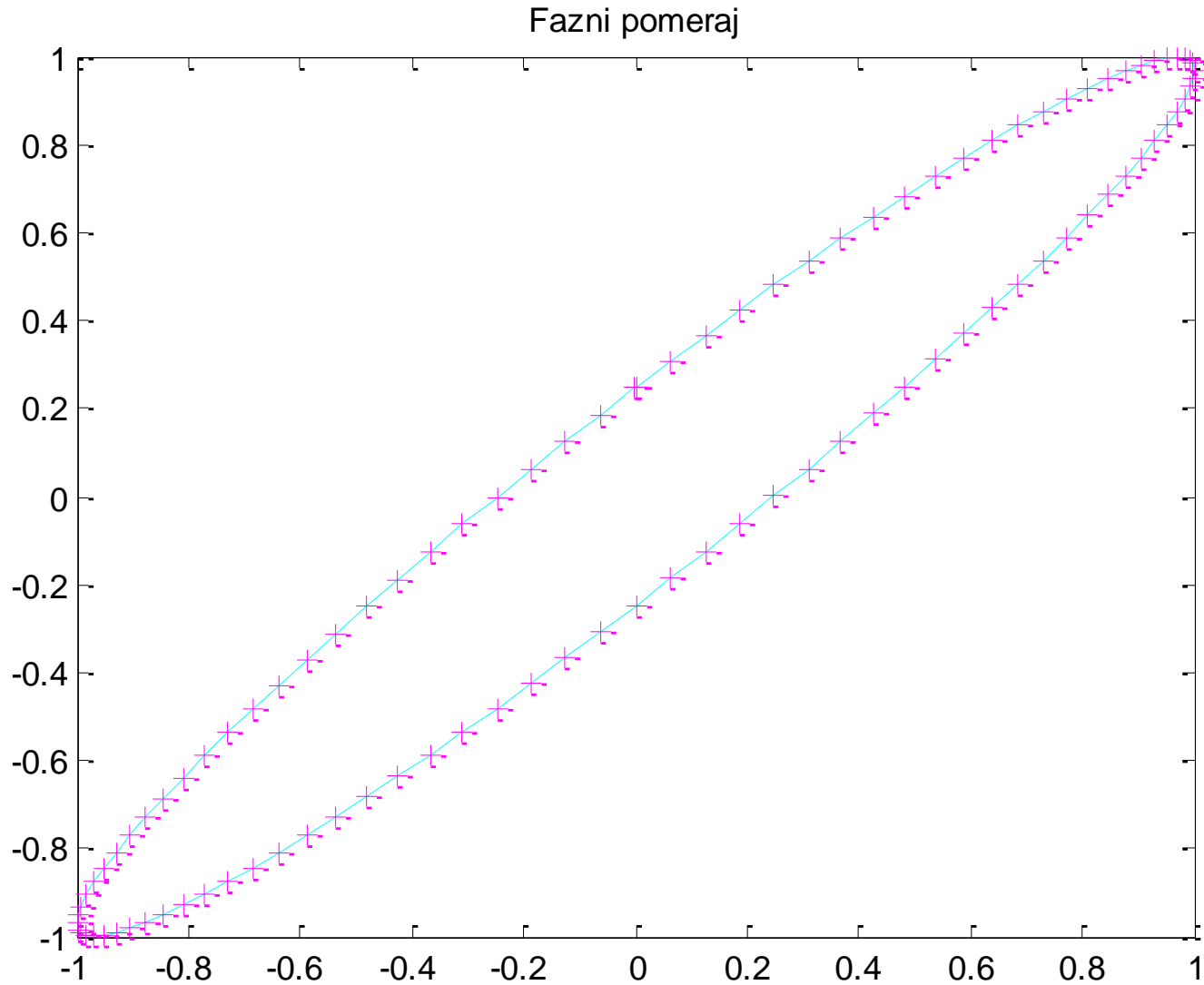
```
plot(x,x.^2), hold on, plot(x,x.^3), hold off
```

- kombinacija predhodnih načina



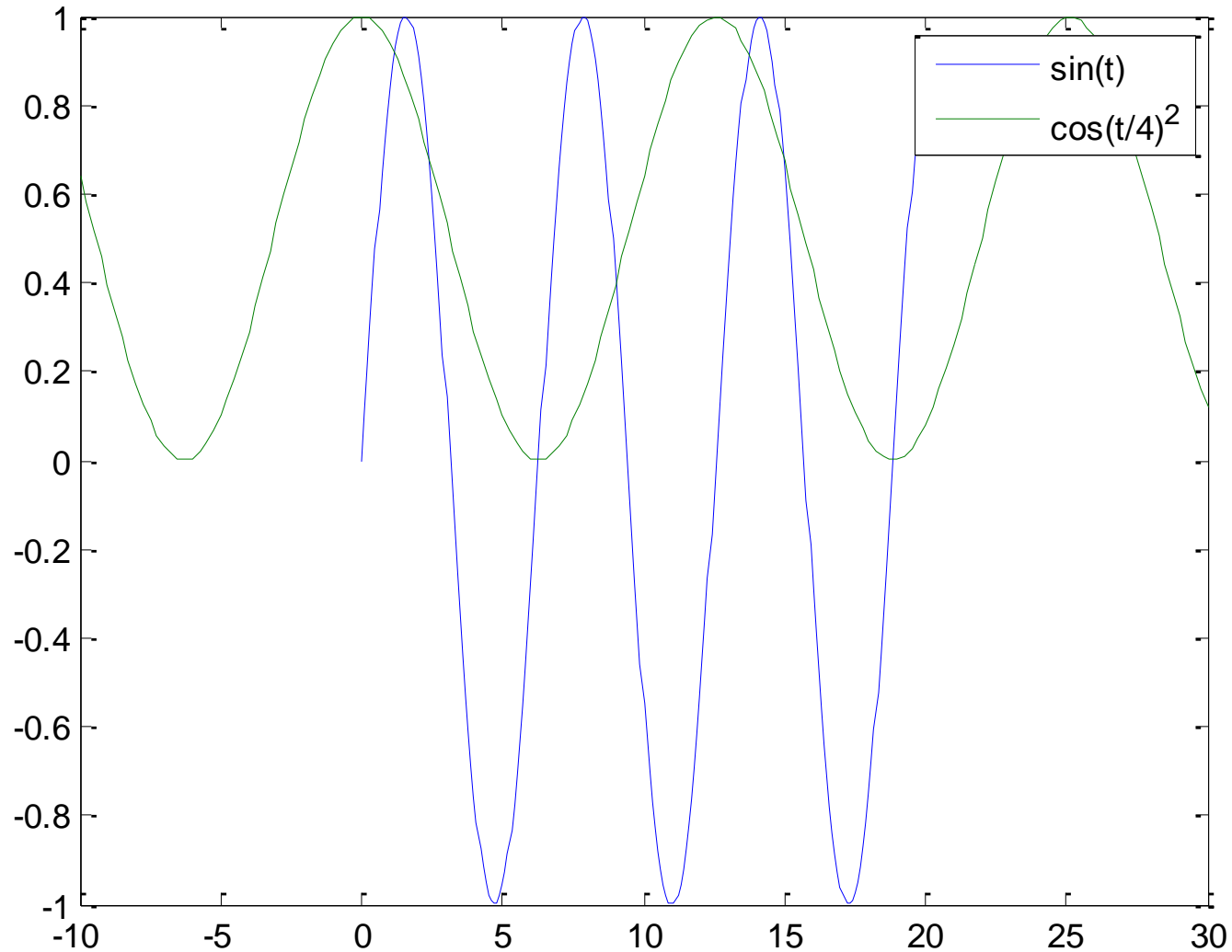
# plot

```
t=0:pi/50:2*pi; x=sin(t); y=sin(t+0.25); plot(x,y,'c-',x,y,'m+'), title('Fazni pomeraj')
```



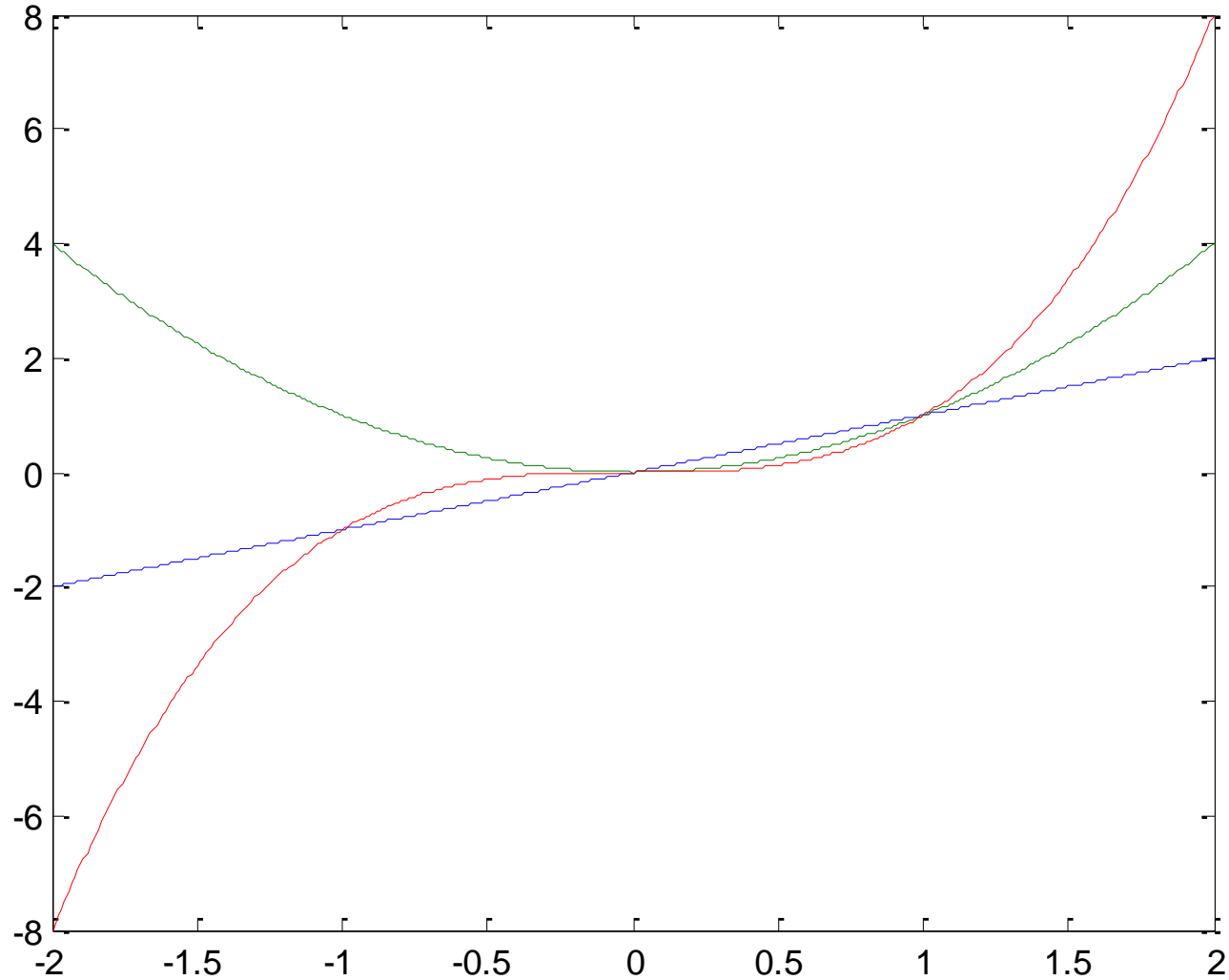
# plot: više argumenata – više krivih

```
x1=0:0.1:20; y1=sin(x1); x2=-10:0.25:30; y2=cos(x2/4); plot(x1,y1,x2,y2.*y2), legend('sin(t)', 'cos(t/4)^2')
```



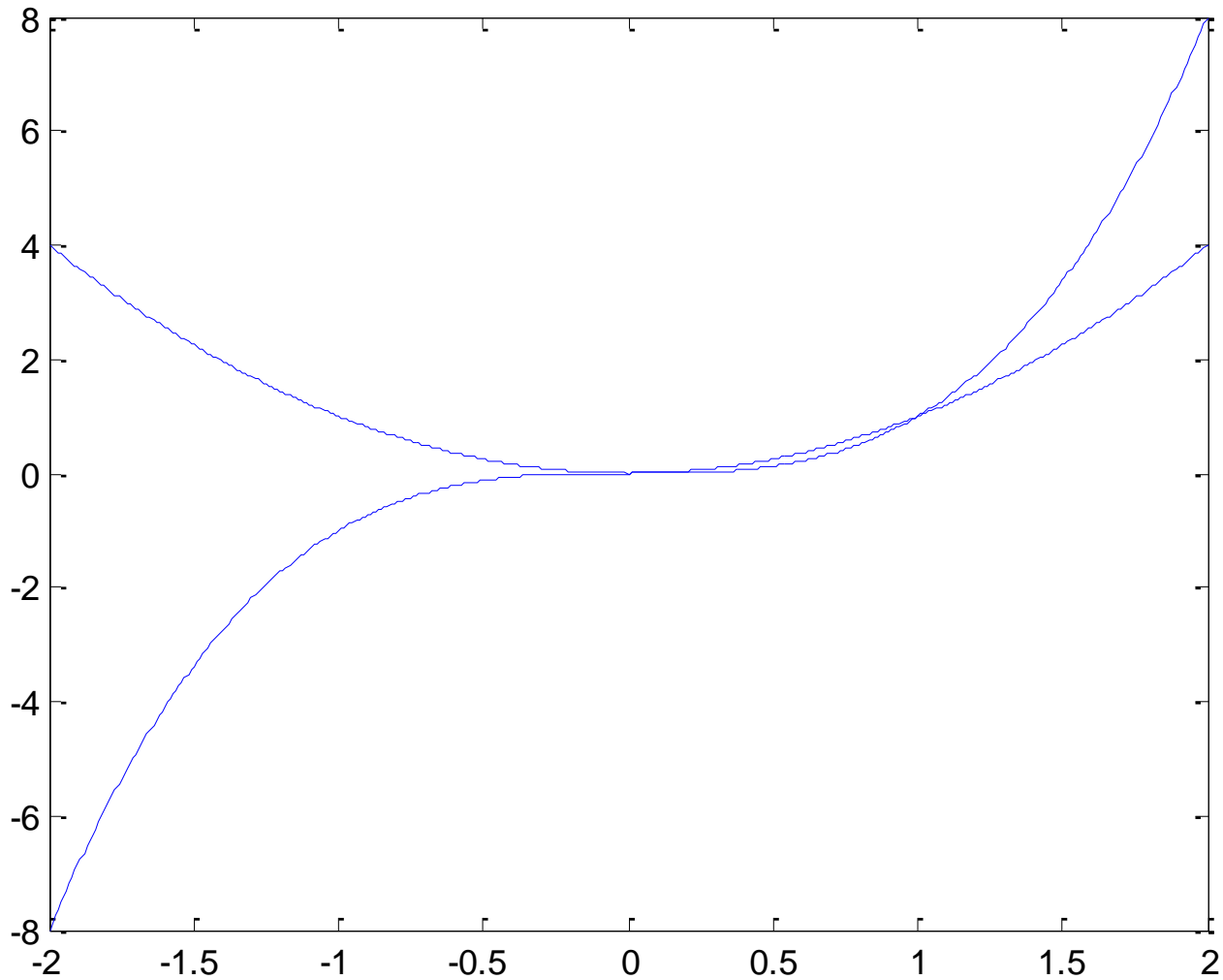
# plot: više kolona – više krivih

```
x=(-2:0.01:2)'; plot(x,[x x.*x x.^3])
```



# plot: docrtana kriva

```
plot(x,x.^2), hold on, plot(x,x.^3), hold off
```



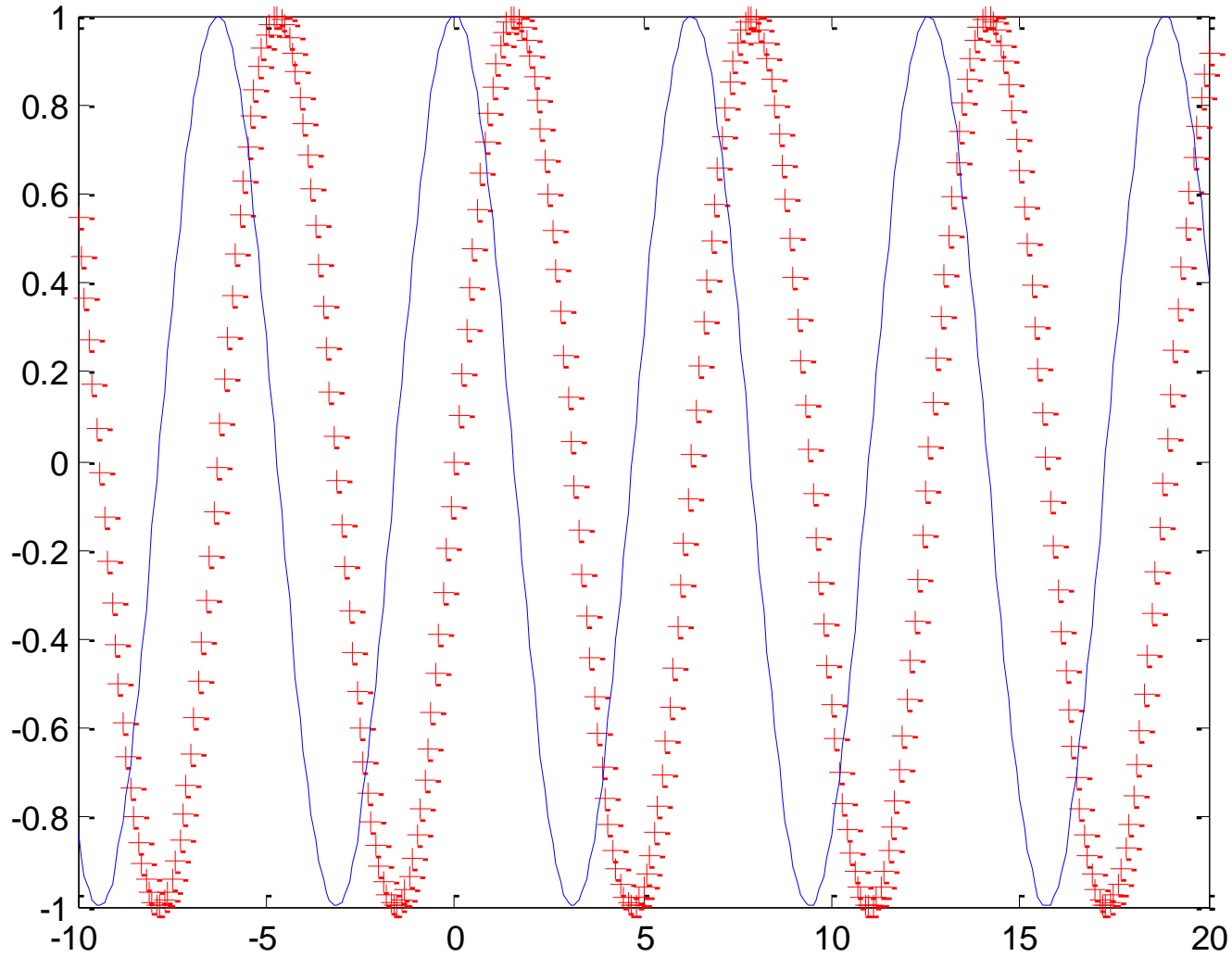
# Formati plot komande

- sa jednim argumentom - `plot(y)`  
crta ekvidistantne  $y$  u zavisnosti od rednog broja tačke 1,2,...  
`x=-10:0.1:20;`  
`plot(sin(x))`
- sa dva argumenta - `plot(x,y)`  
crta  $y$  u zavisnosti od  $x$   
`plot(x,sin(x))`
- sa tri argumenta - `plot(x,y,S)`  
predhodni način uz određivanje boje i tipa linije  
`plot(x,sin(x),'r+')`
- sa četiri i više argumenta - `plot(x1,y1,S1,x2,y2,S2,...)`  
prikaz nekoliko krivih  
`plot(x,sin(x),'r+',x,cos(x))`

*ovo je odnosi i na `loglog`, `semilogx` i `semilogy` komande*

# plot

```
x=-10:0.1:20; plot(x,sin(x),'r+',x,cos(x))
```



# Pikaz Re-Im dijagrama sa kompleksnim brojevima

- ako je argument `plot(Z)` komande vektor kompleksnih brojeva crta se kriva zavisnosti imaginarnog dela od realnog  
(*dogada se: `plot(real(Z),imag(Z))`* )

*primer:*

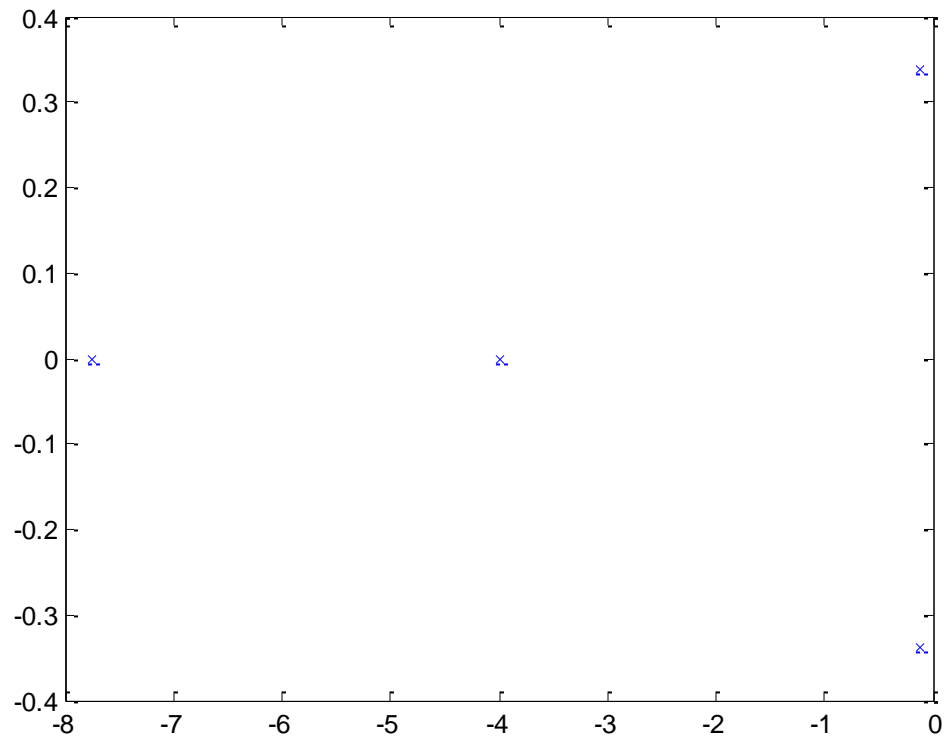
prikaz polova sistema u kompleksnoj s-ravni

```
Q=[1 12 34 9 4];
```

```
plot(roots(Q),'x')
```

# Prikaz položaja kompleksnih korena

```
Q=[1 12 34 9 4]; plot(roots(Q),'x')
```





# Prikaz više dijagrama

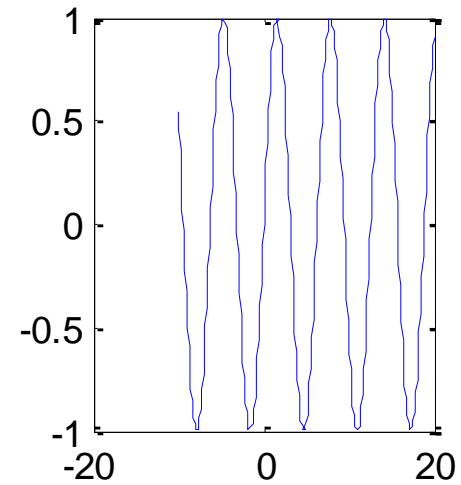
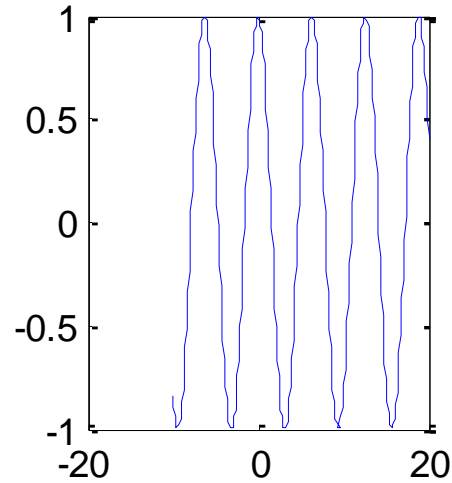
- u okviru istog “prozora”

```
subplot(2,3,6)    % 6. slika od 2 reda po 3
plot(x,sin(x))
subplot(2,3,2)
plot(x,cos(x)), ...
```
- u različitim prozorima

```
figure(1)         % izbor prozora
plot(x,sin(x))
figure(2)
plot(x,cos(x)), ...
```
- **clf** - briše tekuću sliku
- **close** - zatvara prozor slike
- **whitebg** - postavlja belu pozadinu, ...

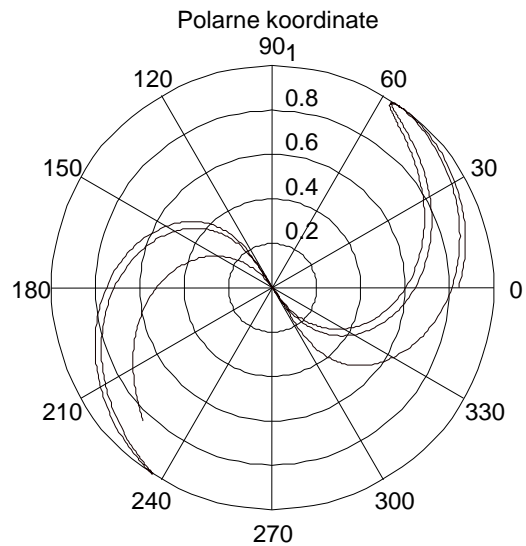
# Prikaz više dijagrama u prozoru - subplot

```
subplot(2,3,6), plot(x,sin(x)), subplot(2,3,2), plot(x,cos(x))
```

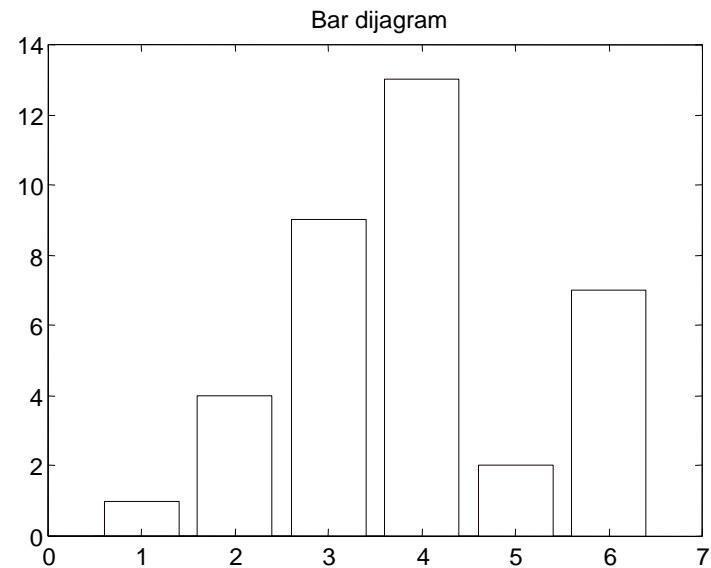


# Posebni X-Y dijagrami

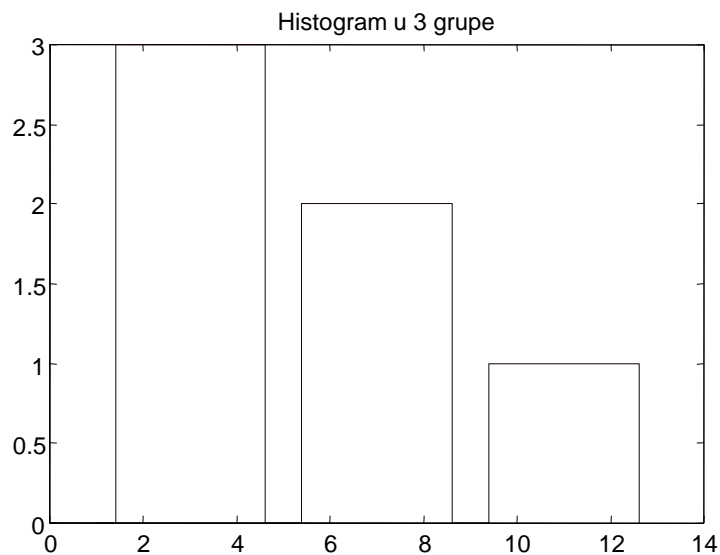
- **bar** - stubičasti dijagram  
x = [1 4 9 13 2 7];  
bar(x), title('Bar dijagram')
- **hist** - histogram  
hist(x,3), title('Histogram u 3 grupe')
- **comet** - animirani prikaz (poput kretanja komete)  
x=0:0.01:3\*pi;  
comet(sin(2.1\*x),sin(x+1))
- **polar** - prikaz u polarnim koordinatama  
polar(sin(2.1\*x),sin(x+1)), title('Polarne koordinate')
- **stairs** - “stepenice”, pogodne za odzive diskretnih sistema  
x=1:10; stairs(x,sin(y))
- **fplot** - crta analitički poznatu funkciju (automatski korak)  
fplot('abs(exp(-j\*x\*(0:9))\*ones(10,1))',[0 2\*pi],'-o')
- ostalo: **compass, errorbar, feather, rose, stem, fill**



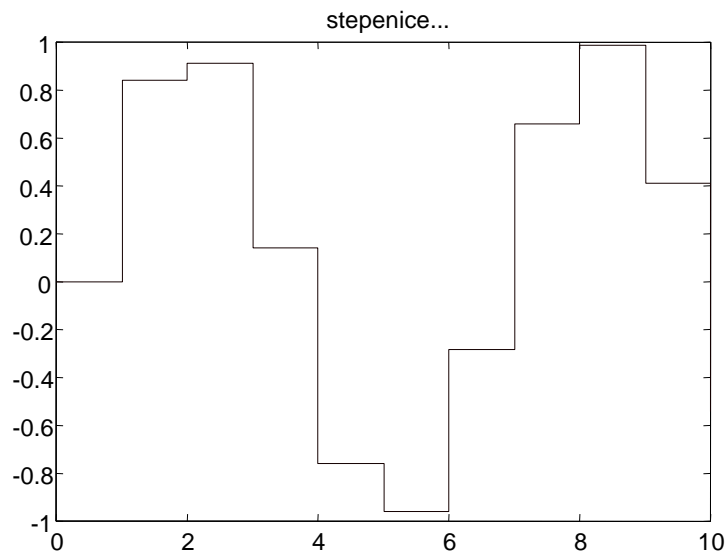
```
polar(sin(2.1*x),sin(x+1)), title('Polarne koordinate')
```



```
x = [1 4 9 13 2 7]; bar(x), title('Bar dijagram')
```



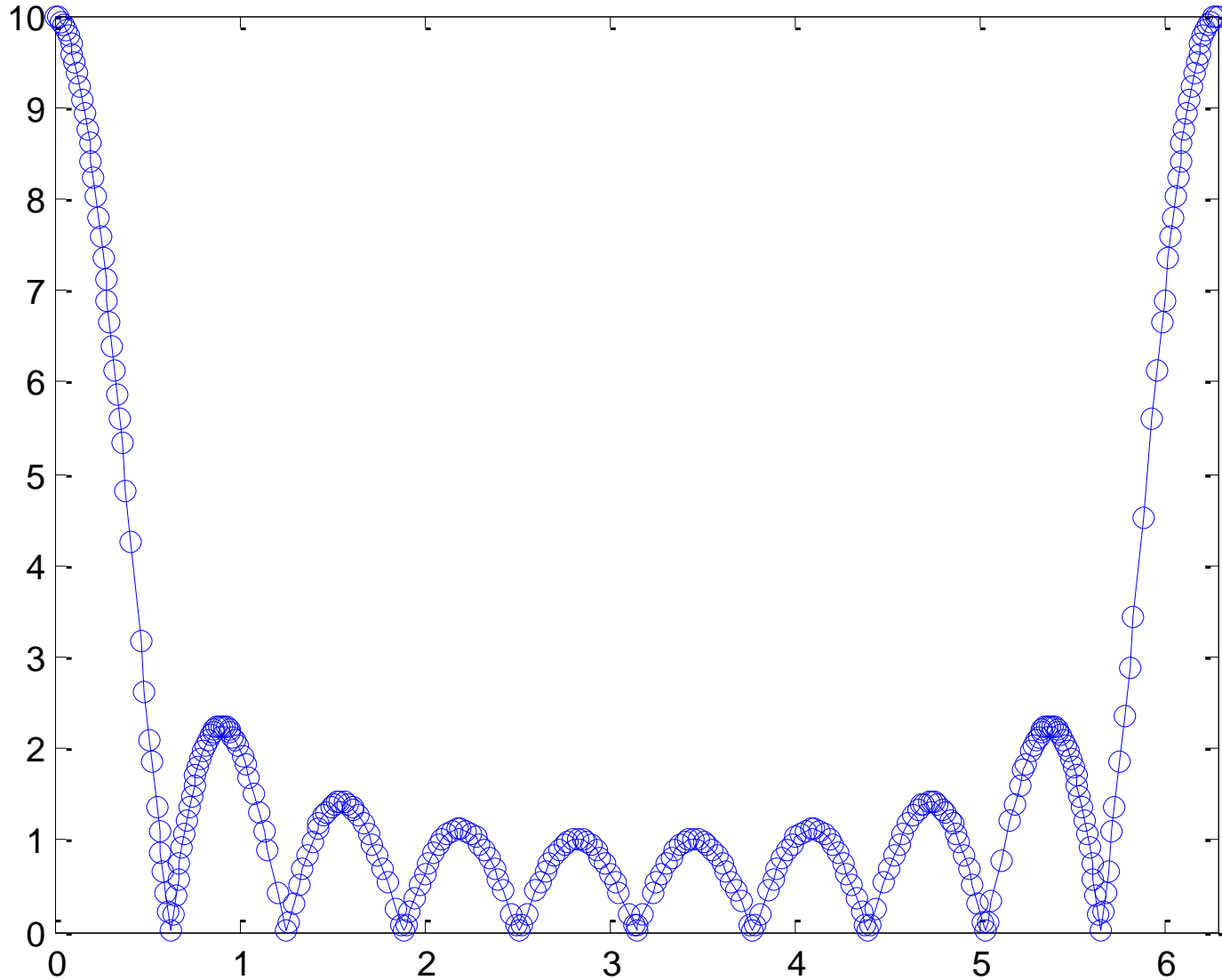
```
hist(x,3), title('Histogram u 3 grupe')
```



```
x=1:10; stairs(x,sin(y))
```

# Crtanje funkcije - fplot

```
fplot('abs(exp(-j*x*(0:9))*ones(10,1))',[0 2*pi],'-o')
```



# 3-D grafika

- postoji mnoštvo komandi i svaka od njih ima više modifikacija
- komande **plot3**, **fill3** i **comet3** predstavljaju proširenje 2-D komandi dodavanjem treće koordinate

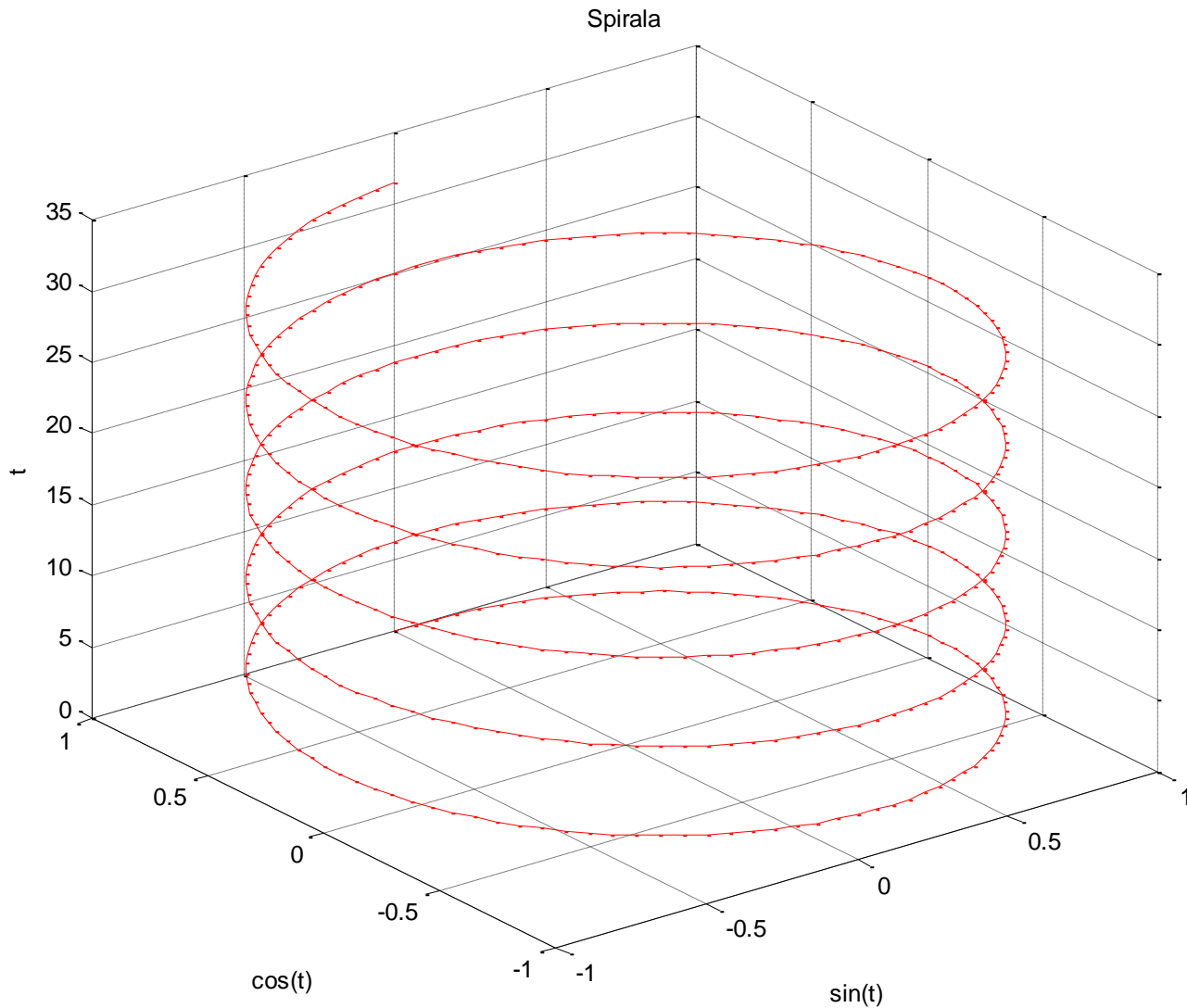
```
t = 0:pi/50:10*pi;  
plot3(sin(t),cos(t),t,'r'); title('Spirala')  
xlabel('sin(t)'), ylabel('cos(t)'), zlabel('t'), grid
```

- crtanje površi:
  - **mesh** - “žičani model”
  - **surf** - “solid model”  
*primer crtanja f-je dve promenljive:*  

```
[X,Y] = meshgrid(-2:.2:2, -2:.2:2);  
Z = X .* exp(-X.^2 - Y.^2);  
surf(X,Y,Z)
```
- varijacije: **meshc**, **meshz**, **surfc**, **surfl**, **slice**, **waterfall**

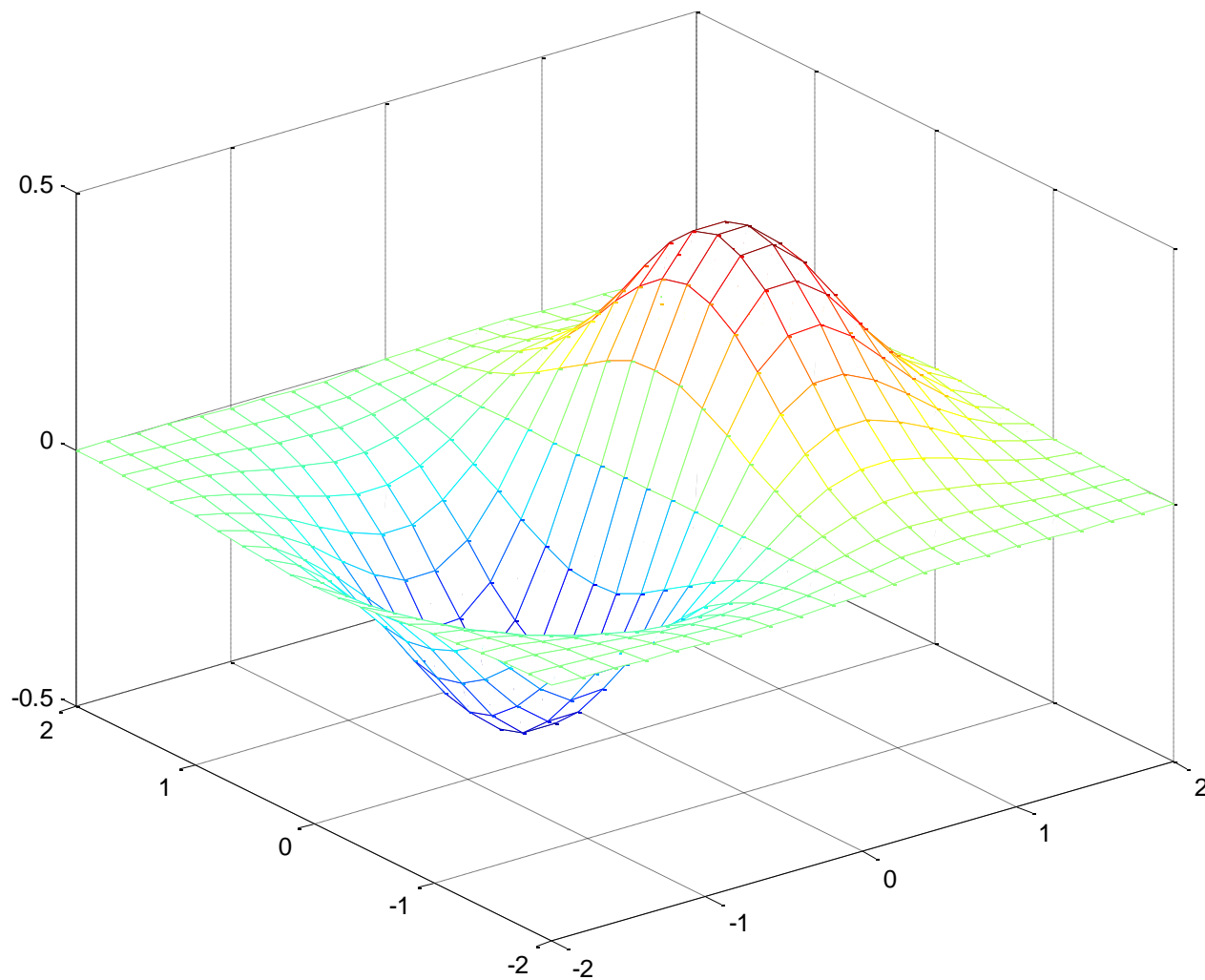
# Kriva u 3D - plot3

```
t = 0:pi/50:10*pi; plot3(sin(t),cos(t),t,'r'); title('Spirala'), xlabel('sin(t)'), ylabel('cos(t)'), zlabel('t'), grid
```



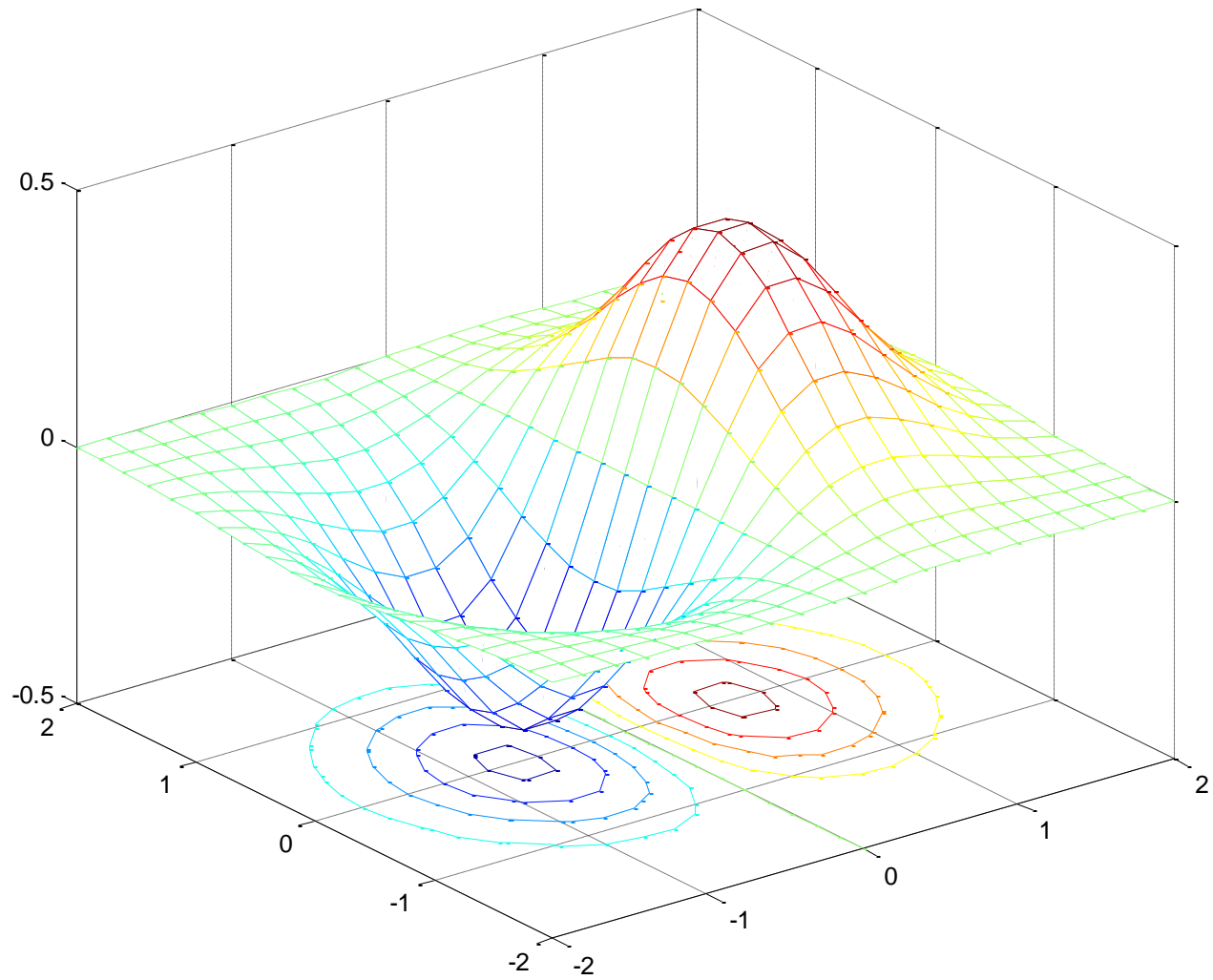
# Žičani model površi - mesh

```
[X,Y] = meshgrid(-2:.2:2, -2:.2:2); Z = X .* exp(-X.^2 - Y.^2); mesh(X,Y,Z)
```

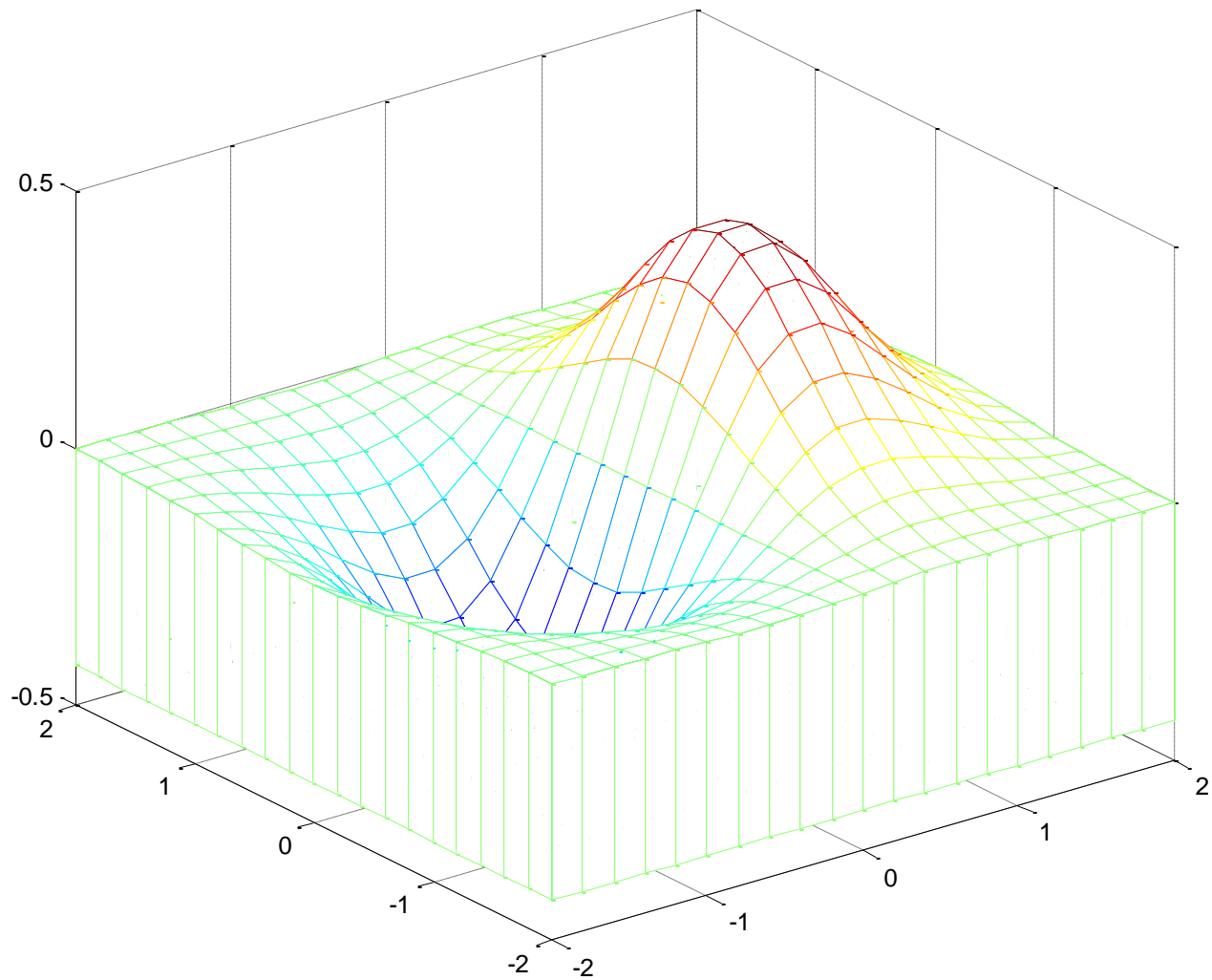




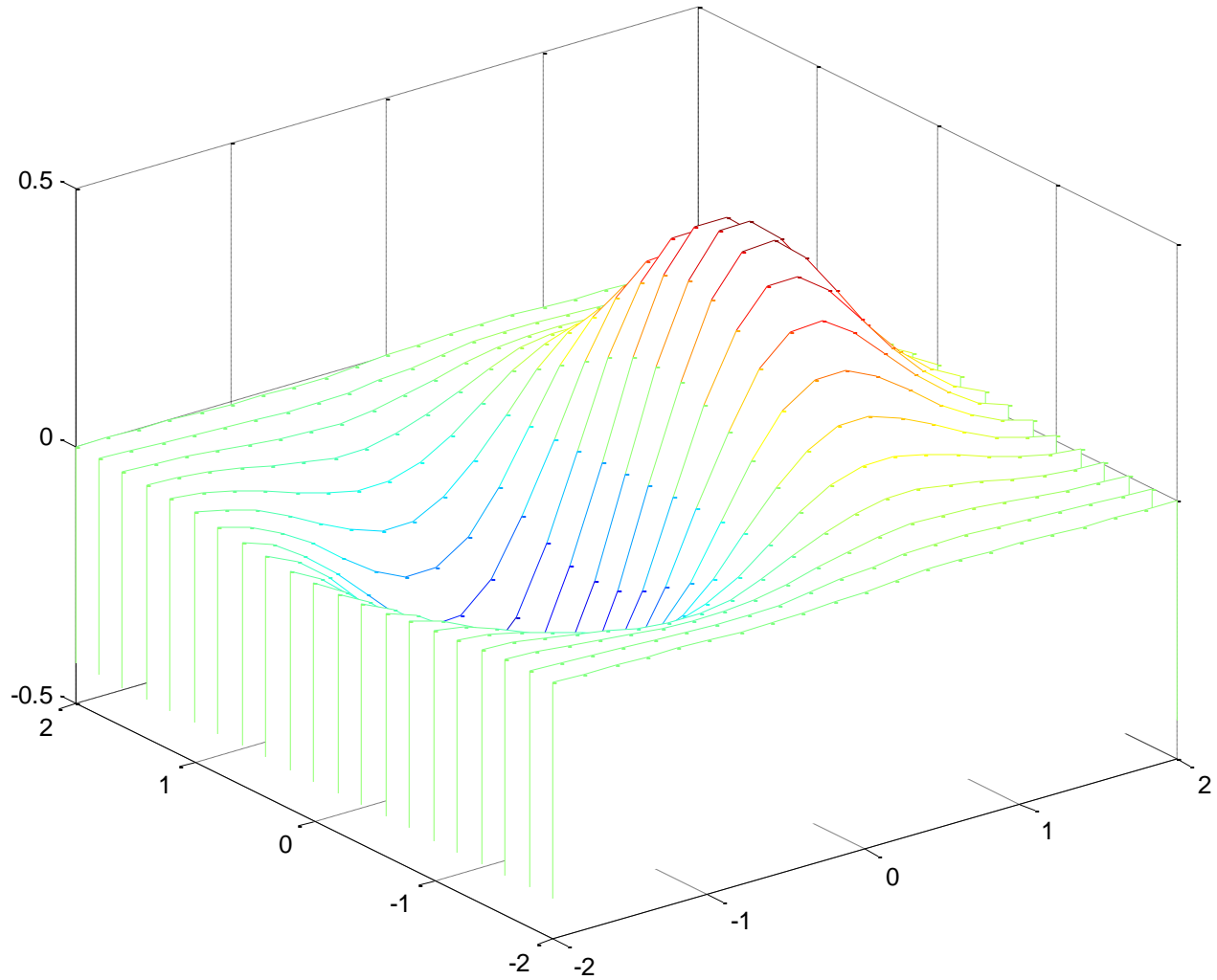
# mesh sa konturama - meshc



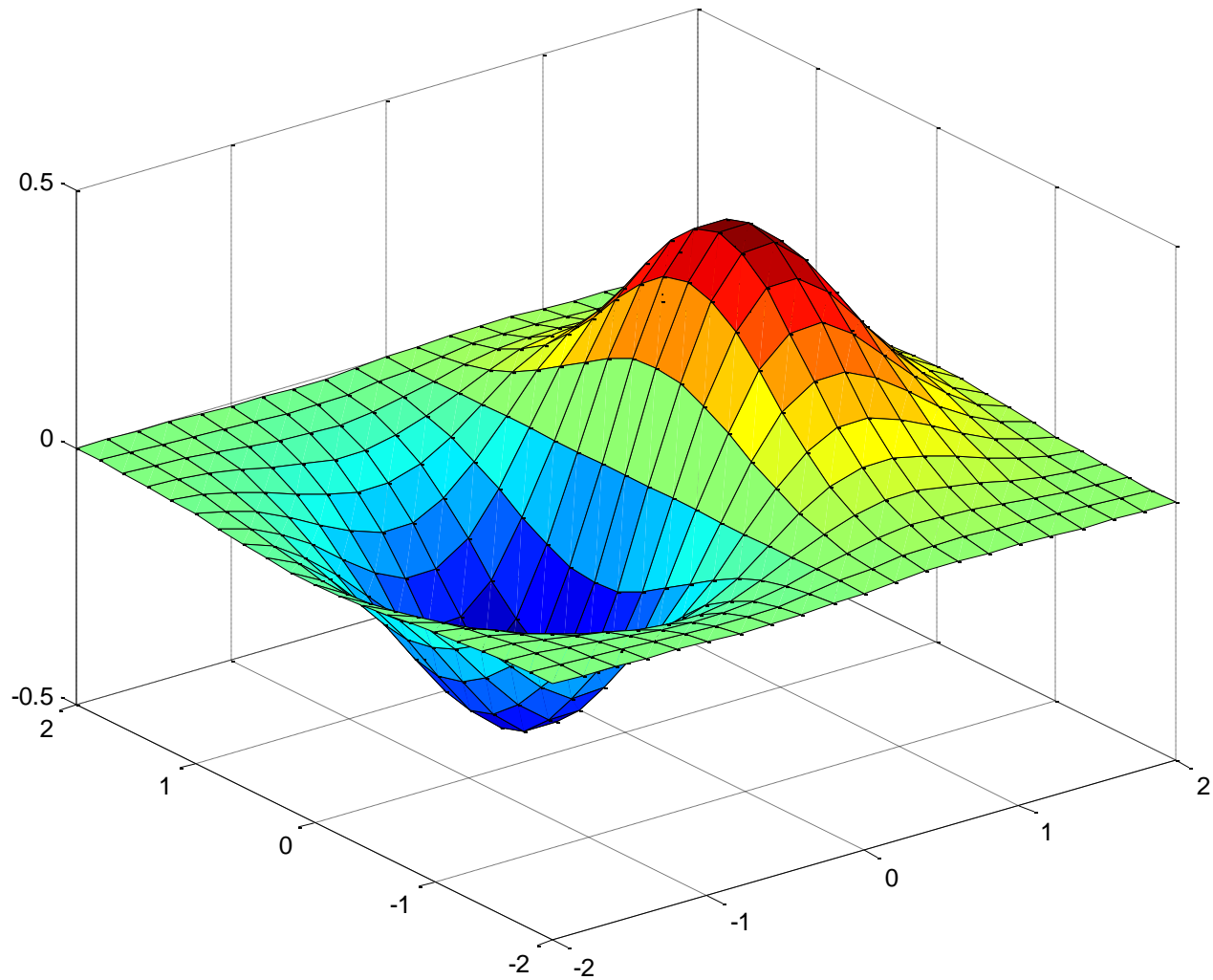
# meshz – naznačena 0-ta ravan



# waterfall

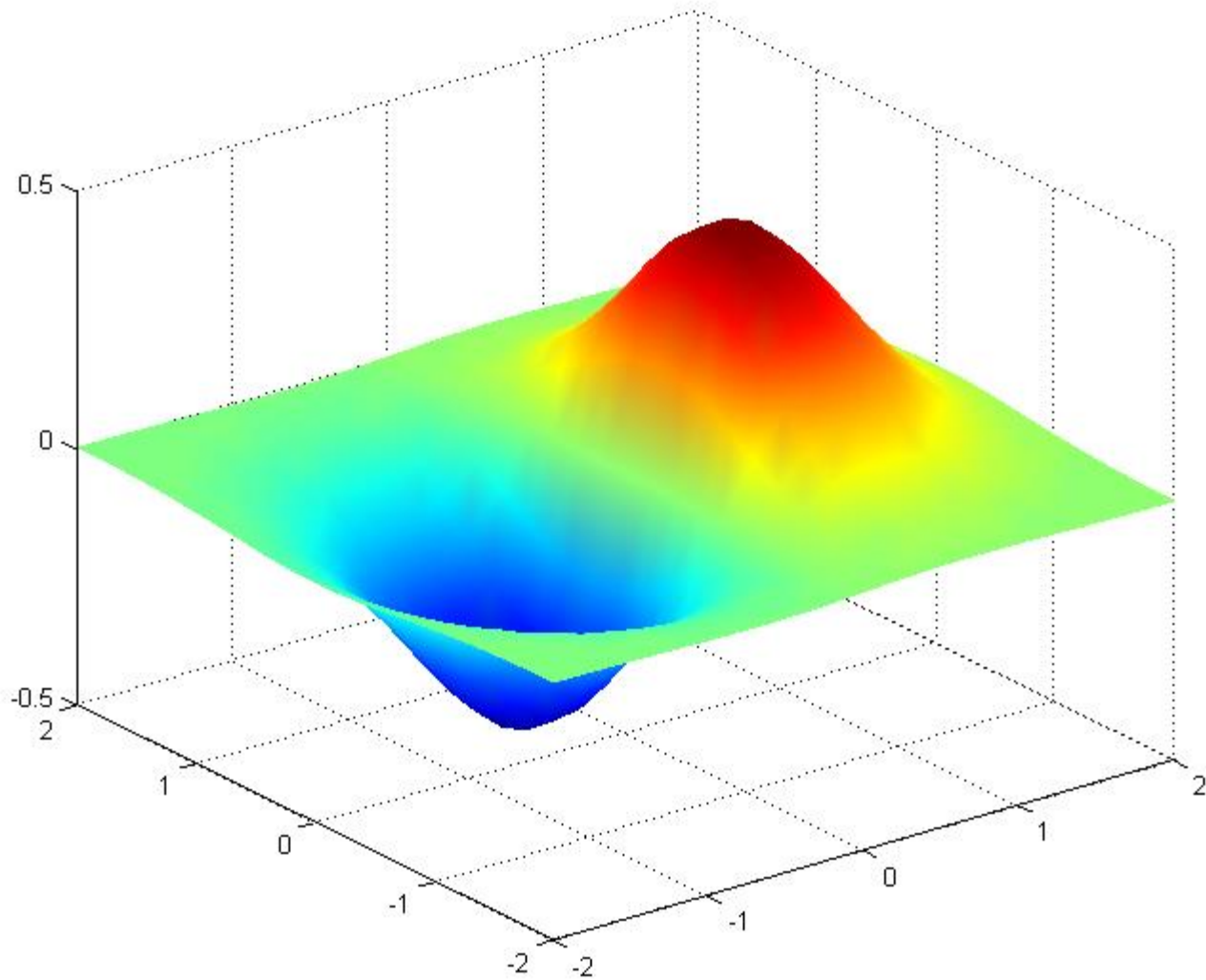


# Crtanje površi - surf



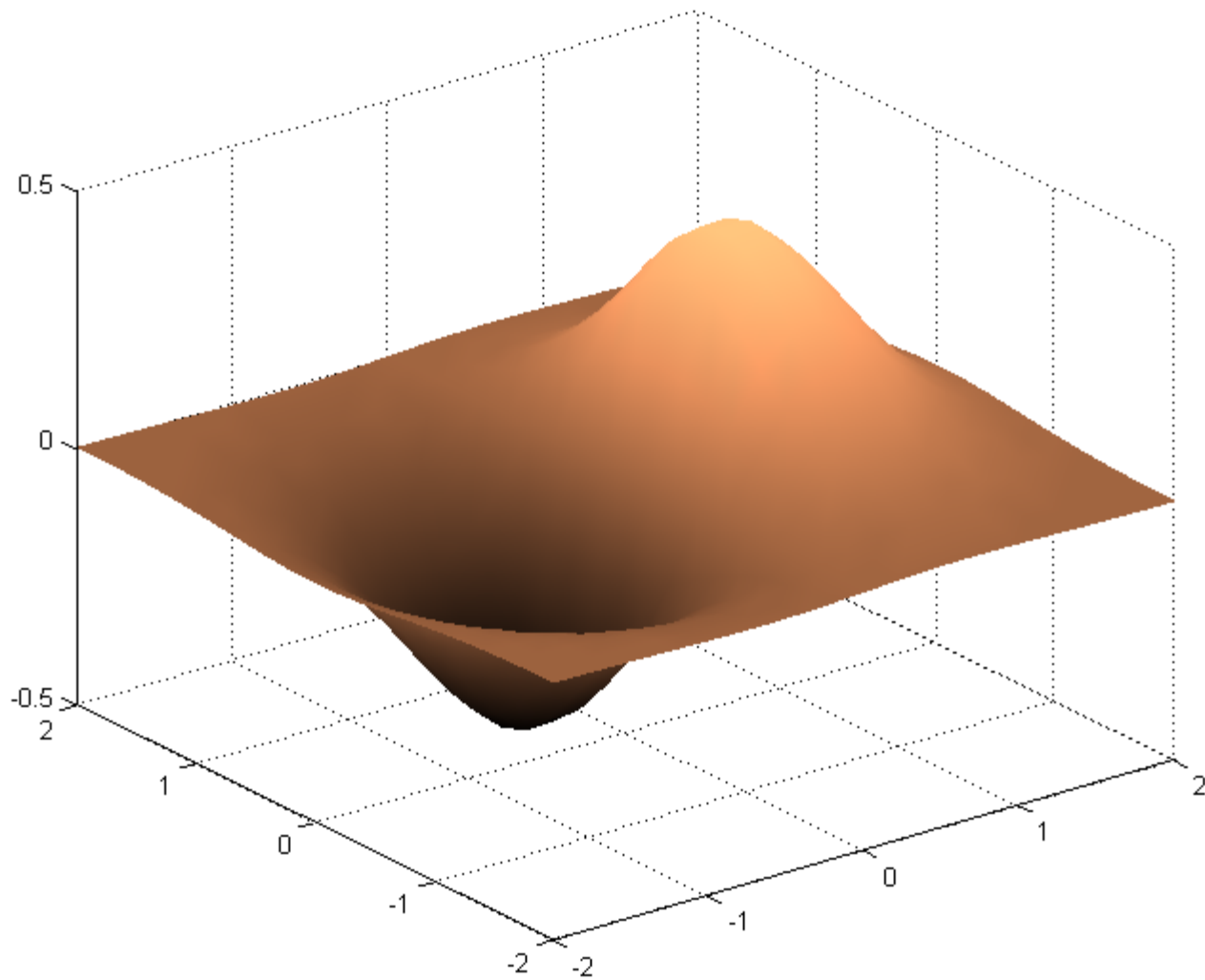
# Toniranje nacrtane površi - shading

```
shading interp; surf(X,Y,Z)
```



# Izbor boja nacrtane površi - colormap

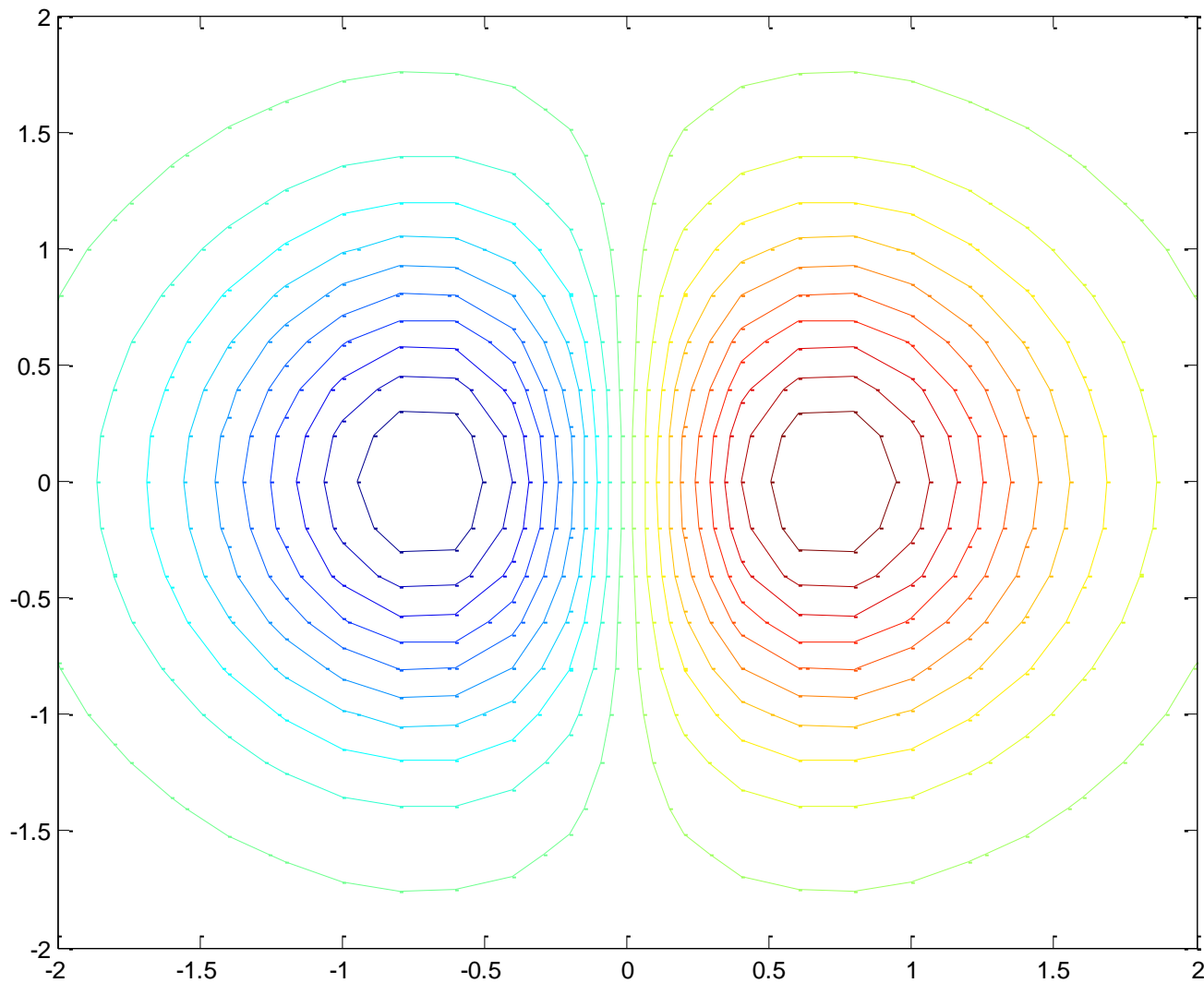
```
colormap(copper); surf(X,Y,Z)
```



# Konturni dijagrami

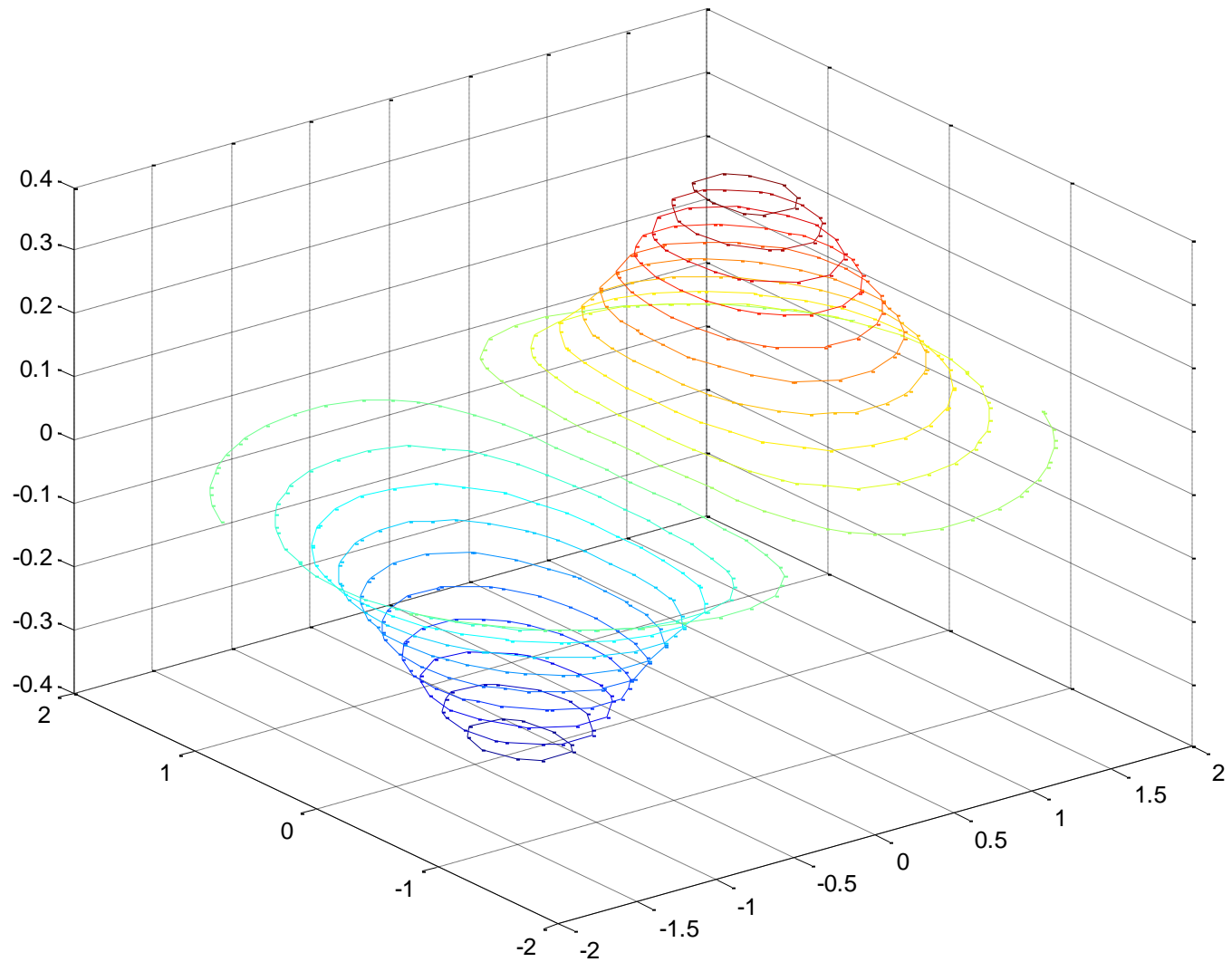
- **contour** - crta konturni dijagram u X-Y ravni  
`contour(X,Y,Z,20)`
- **contour3** - crta konturni dijagram u prostoru  
`contour3(X,Y,Z,20)`
- **contourc** - računa konturni dijagram
- **clabel** - dodaje labele nivoa na dijagram  
`c=contour(X,Y,Z,10); clabel(c)`
- **quiver** - dekoracija oznakama gradijenta  
`[px,py] = gradient(Z,0.2,0.2);`  
`contour(X,Y,Z), hold on`  
`quiver(X,Y,px,py), hold off`

# X-Y konturni dijagram u 20 nivoa

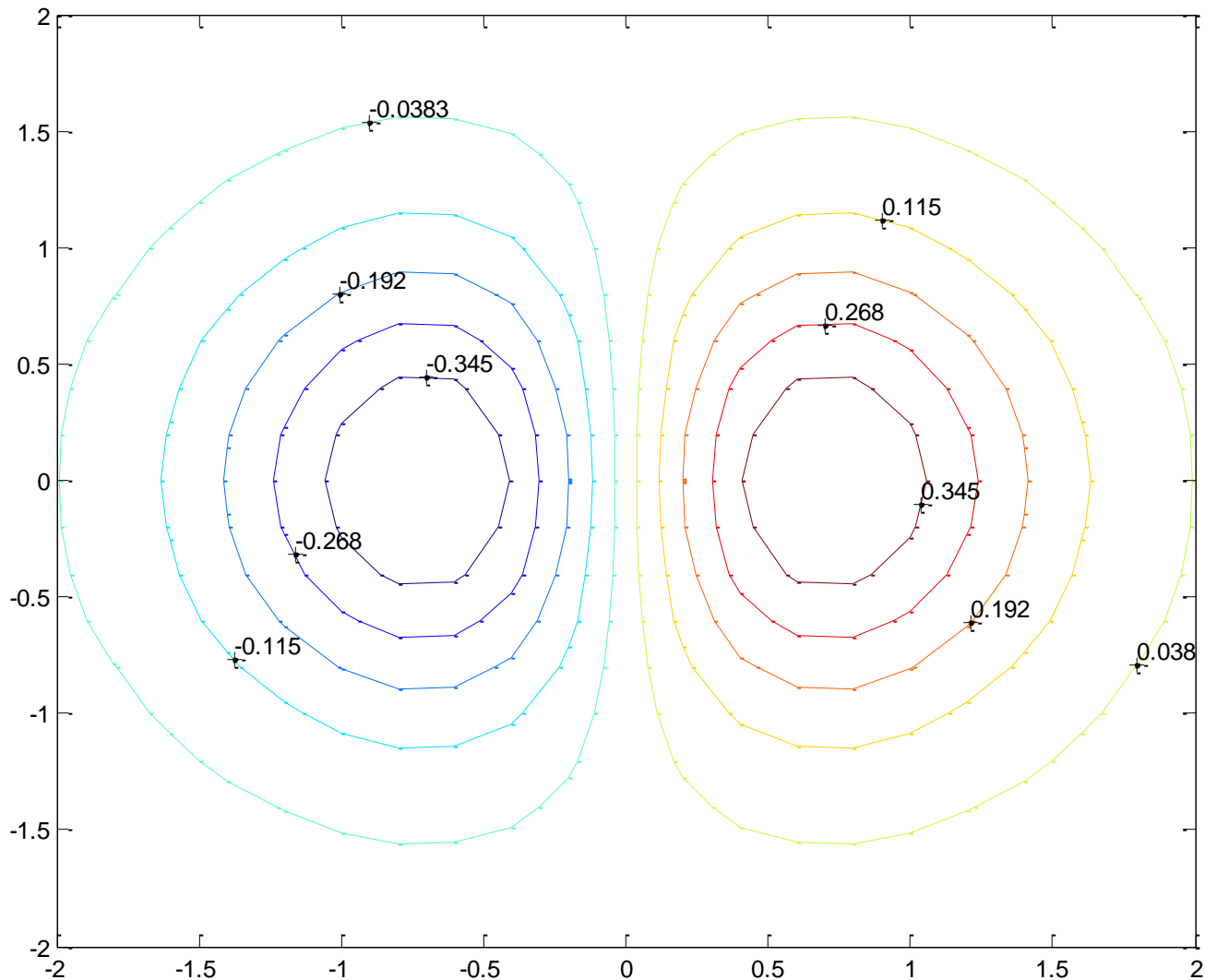




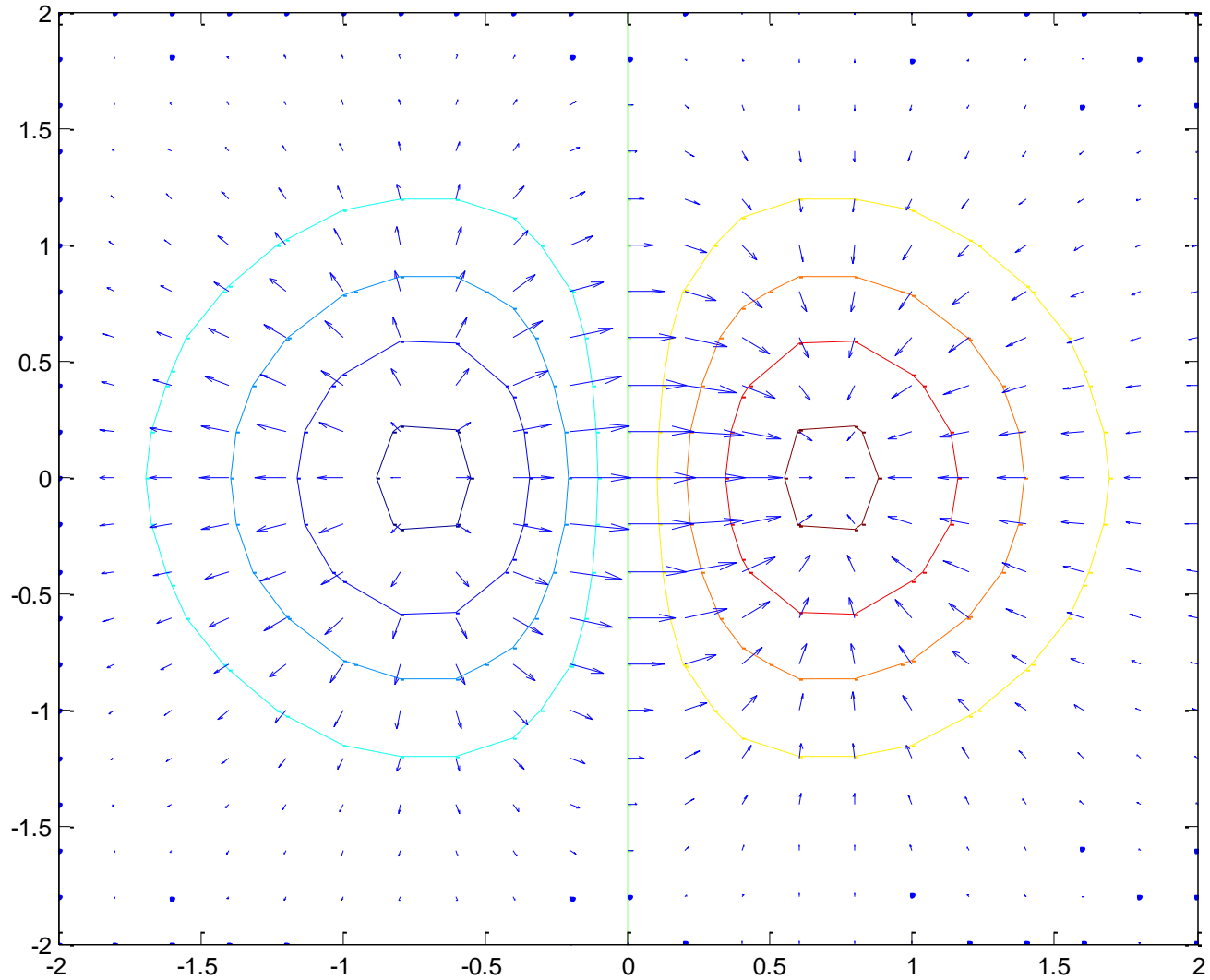
# Konturni dijagram u prostoru



# Prikaz labela nivoa - clabel



# Prikaz gradijenta - quiver

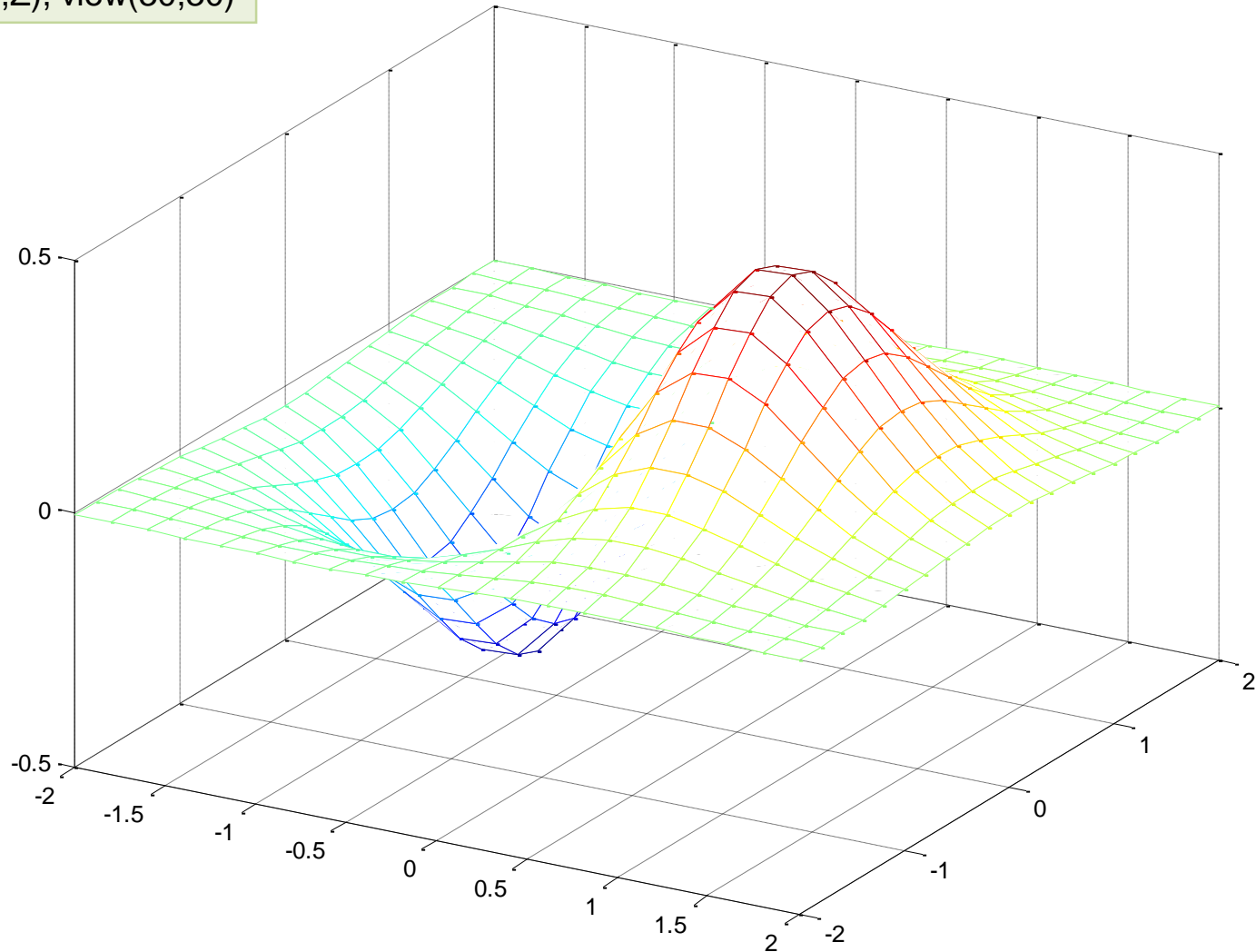


# Prilagođavanje 3-D prikaza

- **axis** - postavljanje razmere na osama (može i za 2-D)
- **caxis** - postavljanje ose pseudo-boja
- **colormap** - rad sa paletom boja
  - **colorbar** - prikazuje paletu pored dijagrama
  - **brighten** - posvetljava sliku
  - **spinmap** - animirano rotira boje u paleti
- **hidden** - prikaz skrivenih linija
- **shading** - način senčenja
- **view** - izmena ugla gledanja
- **viewmtx** - matrice transformacija kod izmene ugla gledanja
- **diffuse, specular, surfl, surfnorm** - rad sa osvetljenjem

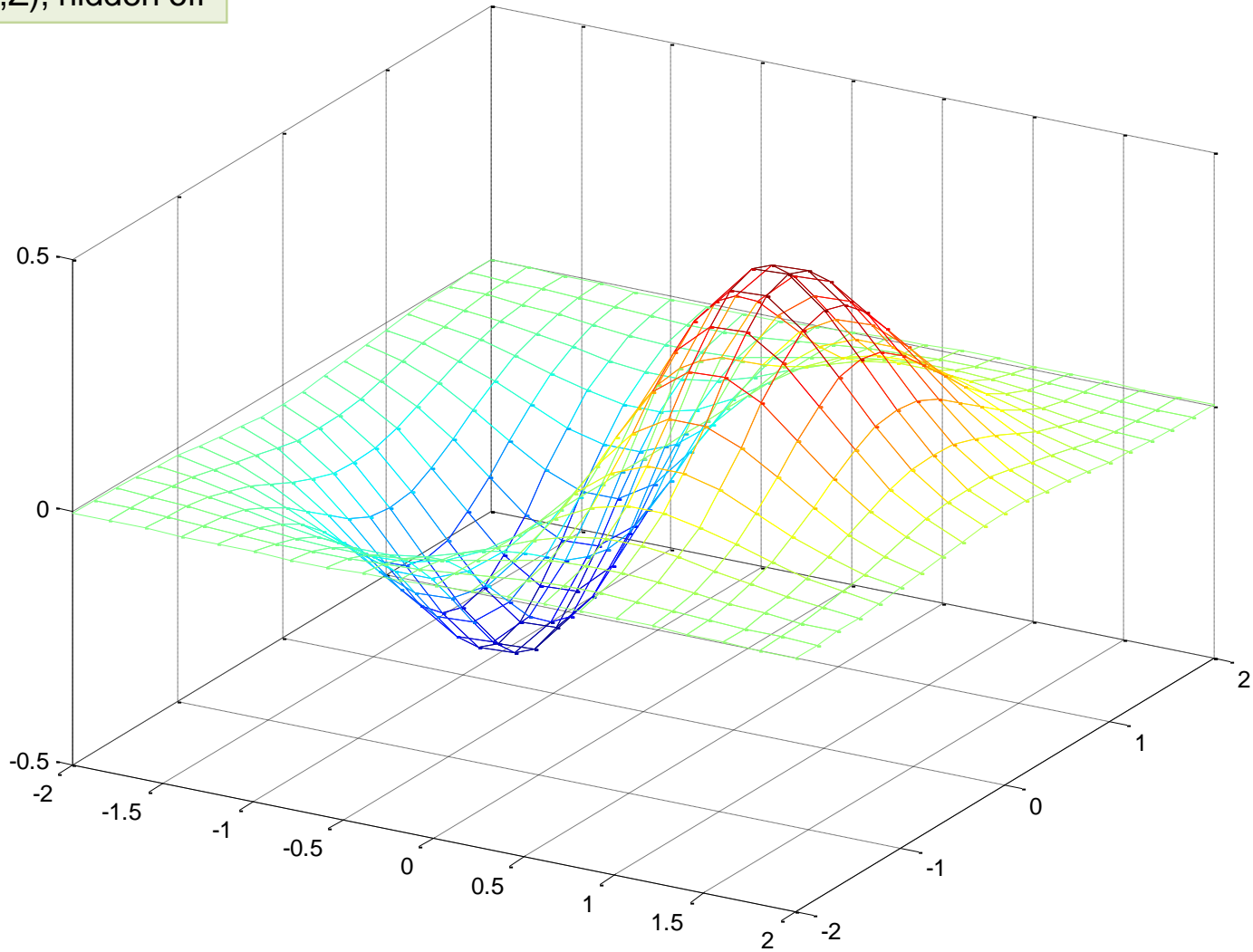
# Promena ugla gledanja

```
mesh(X,Y,Z), view(30,30)
```



# Prikaz skrivenih linija

mesh(X,Y,Z), hidden off

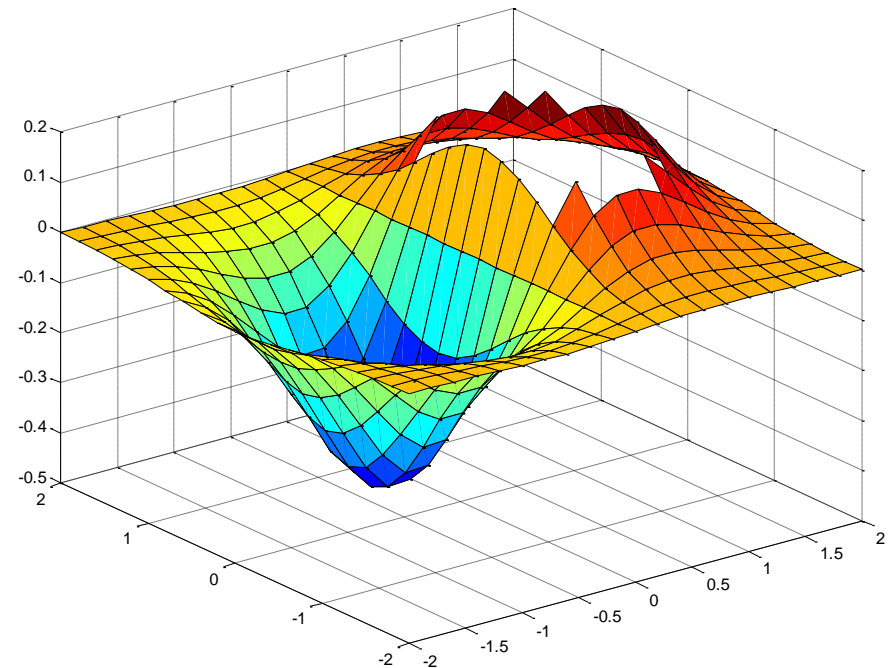


# Isecanje dela slike (dijagrama)

- dodeljivanje z-koordinati **NaN** pravi “rupe”

```
[n,m]=size(Z);  
for i=1:n  
    for j=1:m  
        if Z(i,j)>0.2  
            Z(i,j)=NaN;  
        end  
    end  
end  
end
```

- primenljivo je i na 2-D dijagrame



# Animacija grafičkog prikaza - **movie**

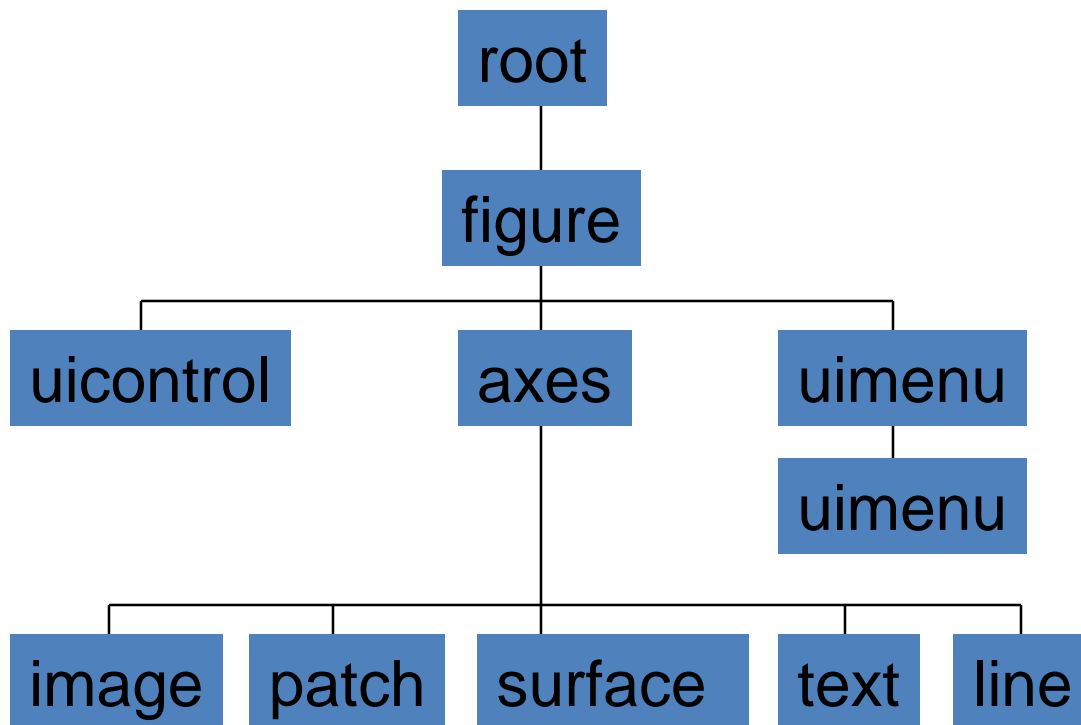
- primer formira 16 slajdova animacije  
svaki slajd se pojedinačno iscrtava i memoriše pomoću getframe

```
t=0:0.1:4*pi;  
M = moviein(16);    % rezervisanje prostora  
for j=1:16  
    plot(t,sin(t+j/3))  
    M(:,j) = getframe;  
end  
movie(M,3,5)        % 3x animacija sa po 5 slajdova/sec
```



# Rad sa grafikom na “niskom nivou”

- omogućen je rad sa linijama, površima i drugim grafičkim objektima kako bi se formirali složeni crteži
- grafički objekti - hijerarhija



**root** - ekran

**figure** - prozor

**axes** - oblast dijagrama

**uicontrol** - interfejs korisnika

**uimenu** - meniji

**image,**

**patch,**

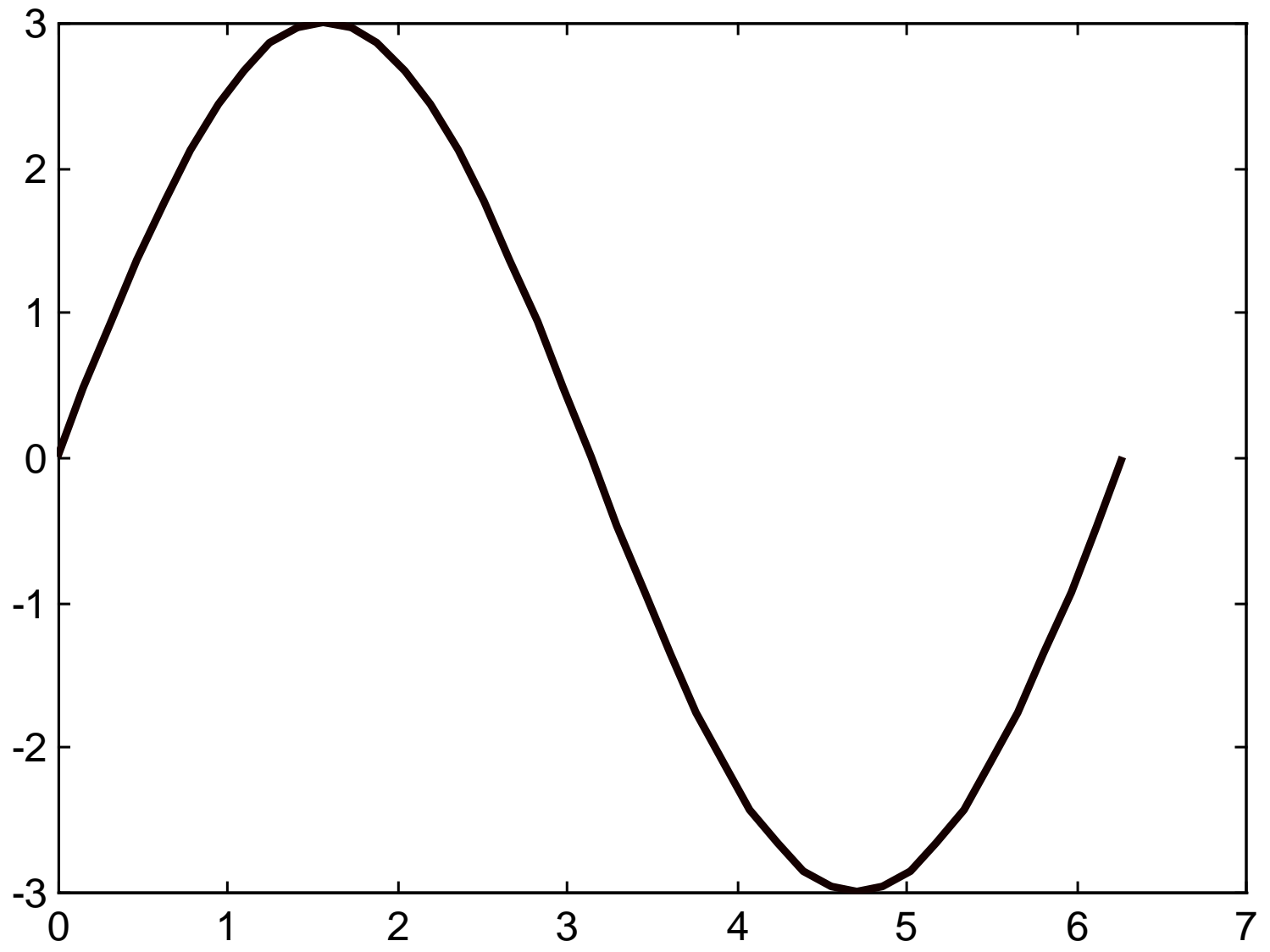
**surface,**

**text,**

**line** - elementi na dijagramu

# Osobine grafičkih objekata

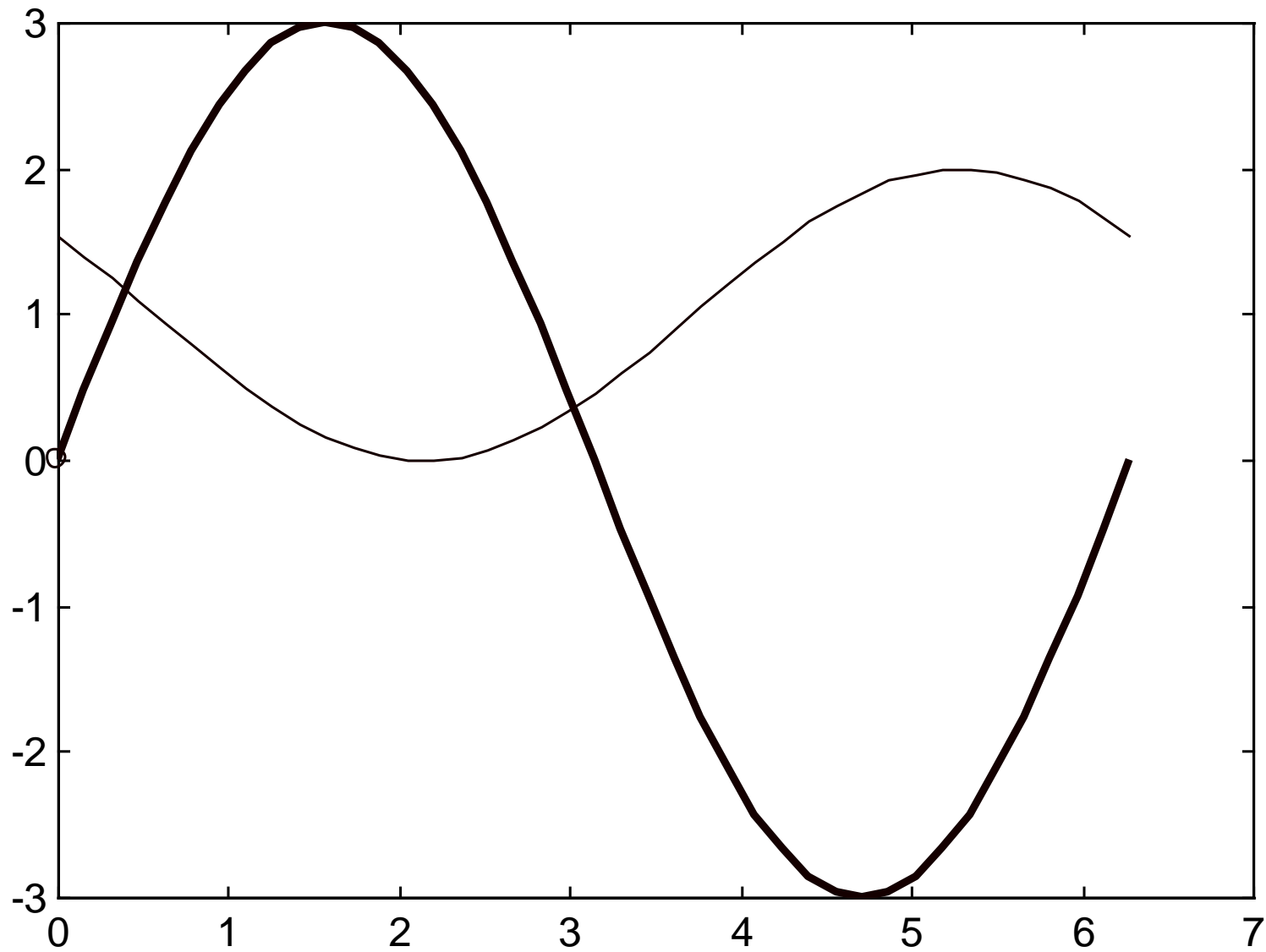
- svaki grafički objekat ima osobine koje se identifikuju imenom
- objekti se identifikuju *handle*-ovima H  
većina grafičkih komandi vraća odgovarajući H, npr  
`h=plot(x,y)`
- očitavanje osobina objekta - komanda **get**  
sve osobine: **get(H)**  
određena osobina: **get(H,'linestyle')**
- osobine se mogu postavljati (menjati) na dva načina
  - kod kreiranja objekta  
`plot(x,y,'linewidth',2)`
  - upotrebom set komande kod već formiranih objekata  
`set(h,'linewidth',3)`
- posebne komande: **gcf** i **gca** vrađaju H za tekuću sliku i ose



# Primer upotrebe **line** komande

- primer docrtava krivu pomoću **line**, pomera je dok preostali deo slike ostaje neizmenjen

```
...  
l=line(t,cos(t+1));  
set(l,'color','red')  
for n=1:10  
    set(l,'ydata',get(l,'ydata')+0.1)  
    pause(0)  
    drawnow          % da bi de videla animacija  
end
```



# Formiranje korisničkog interfejsa - uicontrol

- uicontrol se oslanja na grafičke mogućnosti okruženja u kome se izvršava MATLAB (*MS Windows*)
- postoje
  - *push-button*
  - *check-box*
  - *radio-button*
  - *slider*
  - *pop-up menu*
  - *static text*
  - *editable text*
  - *frame*
- svaka “ui-kontrola” ima mnogo parametara, koji se postavljaju komandom `set(H,'ime parametra',vrednost)`
- može se specificirati *call-back* f-ja koju sistem poziva kao odgovor na akciju kontrole. Npr. pritisnuto je dugme i sl.

# Formiranje menija - uimenu

- praktično, iscrpljene su gotovo sve mogućnosti rada sa menijima koje nudi grafičko okruženje MATLAB-a (*MS Windows*)
- mogu se:
  - formirati podmeniji
  - ubacivati linije-separatori
  - postavljati kvačice
  - onemogućavati stavke
  - dodavati prečice, ...

*primer:*

```
m1=uimenu(gcf,'label','Crtanje dijagrama');  
m2=uimenu(m1,'label','Polarne koordinate',  
  'callback','polar(t,sin(t).^3)');  
m3=uimenu(m1,'label','Dekartove koordinate',  
  'callback','plot(t,sin(t).^3)');
```