

Predavanje 4 - Elementarna matematika i grafika

Tatjana Tomović

Institut za matematiku i informatiku
Prirodno-matematički fakultet
Univerzitet u Kragujevcu

Kragujevac, 2014.

Pregled predavanja

1 Elementarna matematika

- Absolutna vrednost i znak
- Polinomi
- Linearna algebra
- Trigonometrijske funkcije
- Eksponencijalna, logaritamska i stepena funkcija

2 Grafika

- Dvodimenzionalni grafici
- Trodimenzionalni grafici

Absolutna vrednost i znak

- Funkcija *abs* računa moduo kompleksnog broja ili apsolutnu vrednost realnog broja.

Absolutna vrednost i znak

- Funkcija *abs* računa moduo kompleksnog broja ili apsolutnu vrednost realnog broja.
- Funkcija *sign* određuje znak realnog broja, vraća vrednost 1 ako je argument pozitivan, -1 ako je argument negativan i vrednost nula ako je argument jednak nula.

Polinomi

- Polinomi su predstavljeni pomoću vektora vrsta.

Polinomi

- Polinomi su predstavljeni pomoću vektora vrsta.
- Polinom $p(x) = x^4 + 2x^2 + 3x - 4$ se predstavlja kao vektor $[1 \ 0 \ 2 \ 3 \ -5]$.

Polinomi

- Polinomi su predstavljeni pomoću vektora vrsta.
- Polinom $p(x) = x^4 + 2x^2 + 3x - 4$ se predstavlja kao vektor $[1 \ 0 \ 2 \ 3 \ -5]$.
- Koeficijenti polinoma se navode počev od koeficijenta uz najviši stepen promenljive do najnižeg, u opadajućem poretку.

Polinomi

- Polinomi su predstavljeni pomoću vektora vrsta.
- Polinom $p(x) = x^4 + 2x^2 + 3x - 4$ se predstavlja kao vektor $[1 \ 0 \ 2 \ 3 \ -5]$.
- Koeficijenti polinoma se navode počev od koeficijenta uz najviši stepen promenljive do najnižeg, u opadajućem poretku.
- Vrednost polinoma u nekoj tački se izračunava koristeći funkciju *polyval*.

Polinomi

- Polinomi su predstavljeni pomoću vektora vrsta.
- Polinom $p(x) = x^4 + 2x^2 + 3x - 4$ se predstavlja kao vektor $[1 \ 0 \ 2 \ 3 \ -5]$.
- Koeficijenti polinoma se navode počev od koeficijenta uz najviši stepen promenljive do najnižeg, u opadajućem poretku.
- Vrednost polinoma u nekoj tački se izračunava koristeći funkciju *polyval*.

```
>> p = [1 0 2 3 -5];  
>> polyval(p, 1)      >> a(1, 3)  
>> A = [1 2; 0 1];    >> polyval(p, A)
```

Polinomi

- Nule polinoma možemo odrediti koristeći funkciju *roots*.

Polinomi

- Nule polinoma možemo odrediti koristeći funkciju *roots*.
- Izračunavanje koeficijenata polinoma na osnovu vrednosti nula polinoma, može se izvršiti koristeći funkciju *poly*.

Polinomi

- Nule polinoma možemo odrediti koristeći funkciju *roots*.
- Izračunavanje koeficijenata polinoma na osnovu vrednosti nula polinoma, može se izvršiti koristeći funkciju *poly*.
- Funkcija *polyint* izračunava vrednost neodređenog integrala polinoma.

Polinomi

- Nule polinoma možemo odrediti koristeći funkciju *roots*.
- Izračunavanje koeficijenata polinoma na osnovu vrednosti nula polinoma, može se izvršiti koristeći funkciju *poly*.
- Funkcija *polyint* izračunava vrednost neodređenog integrala polinoma.

```
>> nule = roots(p)    >> poly(nule)
>> p = [1 1];        >> q = [1 0 1];
>> polyint(q)         >> polyint(p, 5)
```

Polinomi

- Nule polinoma možemo odrediti koristeći funkciju *roots*.
- Izračunavanje koeficijenata polinoma na osnovu vrednosti nula polinoma, može se izvršiti koristeći funkciju *poly*.
- Funkcija *polyint* izračunava vrednost neodređenog integrala polinoma.

```
>> nule = roots(p)    >> poly(nule)
>> p = [1 1];        >> q = [1 0 1];
>> polyint(q)         >> polyint(p, 5)
```

Kada funkciju pozivamo sa dva argumenta Drugi argument je konstanta koja se javlja prilikom izračunavanja neodređenog integrala (podrazumevana vrednost je nula).

Polinomi

- Funkcija *polyder* računa izvod polinoma, proizvoda polinoma ili količnika polinoma.

Polinomi

- Funkcija *polyder* računa izvod polinoma, proizvoda polinoma ili količnika polinoma.
- Množenje dva polinoma može se postići koristeći funkciju *conv*.

Polinomi

- Funkcija *polyder* računa izvod polinoma, proizvoda polinoma ili količnika polinoma.
- Množenje dva polinoma može se postići koristeći funkciju *conv*.
- Količnik dva polinoma, kao i ostatak pri deljenju se izračunavaju koristeći funkciju *deconv*.

Polinomi

- Funkcija *polyder* računa izvod polinoma, proizvoda polinoma ili količnika polinoma.
- Množenje dva polinoma može se postići koristeći funkciju *conv*.
- Količnik dva polinoma, kao i ostatak pri deljenju se izračunavaju koristeći funkciju *deconv*.

```
>> p = [1 1];  
>> polyder(q)  
>> [num, den] = polyder(p, q)  
>> [a, b] = deconv(p, q)
```

```
>> q = [1 0 1];  
>> proizvod = polyder(p, q)  
>> conv(p, q)
```

Polinomi

- Fitovanje podataka polinomima vrši se naredbom *polyfit*.

Polinomi

- Fitovanje podataka polinomima vrši se naredbom *polyfit*.
- *polyfit*(x, y, n) nalazi koeficijente polinoma $p(x)$ stepena n koji fituje podatke $p(x(i))$ sa $y(i)$.

Polinomi

- Fitovanje podataka polinomima vrši se naredbom *polyfit*.
- *polyfit*(x, y, n) nalazi koeficijente polinoma $p(x)$ stepena n koji fituje podatke $p(x(i))$ sa $y(i)$.

```
x = (0 : 0.1 : 2.5)';  
y = erf(x);  
p = polyfit(x, y, 6)  
f = polyval(p, x);
```

Linearna algebra

- Funkcije *det* i *inv* se koje za izračunavanje determinante i inverzne matrice date matrice.

Linearna algebra

- Funkcije *det* i *inv* se koje za izračunavanje determinante i inverzne matrice date matrice.
- Funkcija *eig* računa sopstvene vrednosti i sopstvene vektore matrice.

Linearna algebra

- Funkcije *det* i *inv* se koje za izračunavanje determinante i inverzne matrice date matrice.
- Funkcija *eig* računa sopstvene vrednosti i sopstvene vektore matrice.
- Funkcija *eig*, kad je pozvana sa dva izlazna argumenta, vraća sopstvene vektore i dijagonalnu matricu koja na glavnoj dijagonali ima sopstvene vrednosti. Ako je pozvana sa jednim izlaznim argumentom vraća vektor sopstvenih vrednosti.

Linearna algebra

- Funkcije *det* i *inv* se koje za izračunavanje determinante i inverzne matrice date matrice.
- Funkcija *eig* računa sopstvene vrednosti i sopstvene vektore matrice.
- Funkcija *eig*, kad je pozvana sa dva izlazna argumenta, vraća sopstvene vektore i dijagonalnu matricu koja na glavnoj dijagonali ima sopstvene vrednosti. Ako je pozvana sa jednim izlaznim argumentom vraća vektor sopstvenih vrednosti.

```
>> A = [1 2 3; 2 3 1; 3 1 2];
>> det(A)                                >> inv(A)
>> [V, d] = eig(A)                       >> d = eig(A)
```

Trigonometrijske funkcije

sin, cos, tan, cot

sind, cosd, tand, cotd

asin, acos, atan, acot

asind, acosd, atand, acotd

sin, cos, tan, cot

sa argumentom u radijanima

sin, cos, tan, cot

sa argumentom u stepenima

arcsin, arccos, arctan, arccot

sa argumentom u radijanima

arcsin, arccos, arctan, arccot

sa argumentom u stepenima

Trigonometrijske funkcije

sin, cos, tan, cot

sind, cosd, tand, cotd

asin, acos, atan, acot

asind, acosd, atand, acotd

sin, cos, tan, cot

sa argumentom u radijanima

sin, cos, tan, cot

sa argumentom u stepenima

arcsin, arccos, arctan, arccot

sa argumentom u radijanima

arcsin, arccos, arctan, arccot

sa argumentom u stepenima

Funkcije mogu da deluju nad matrichnim argumentnom, i u tom slučaju, funkcije deluju nad svakim pojedinačnim elementom matrice.

Eksponencijalna, logaritamska i stepena funkcija

<i>exp</i>	<i>eksponencijalna funkcija</i>
<i>log</i>	<i>prirodni logaritam</i>
<i>log10</i>	<i>logaritam za osnovu 10</i>
<i>log2</i>	<i>logaritam za osnovu 2</i>
<i>sqrt</i>	<i>kvadratni koren</i>
<i>nthroot</i>	<i>realni n-ti koren realnog argumenta</i>
<i>reallog</i>	<i>logaritam sa realnim i pozitivnim argumentom</i>
<i>realpow</i>	<i>stepenovanje sa realnim izlazom</i>
<i>realsqrt</i>	<i>koren sa realnim i pozitivnim argumentom</i>

Eksponencijalna, logaritamska i stepena funkcija

```
>> log(2)           >> log10(2)
>> sqrt(-1)        >> nthroot(-2, 3)
>> nthroot(2, 3)    >> nthroot(-2, 4)
>> realpow(i, 2)
>> sqrt([149])
```

Eksponencijalna, logaritamska i stepena funkcija

```
>> log(2)           >> log10(2)
>> sqrt(-1)         >> nthroot(-2, 3)
>> nthroot(2, 3)     >> nthroot(-2, 4)
>> realpow(i, 2)
>> sqrt([149])
```

Funkcije *reallog* i *realsqrt* prihvataju za argumente samo nizove pozitivnih realnih brojeva. Funkcija *realpow* ne ograničava mogući skup ulaznih vrednosti, ali zato redukuje skup mogućih izlaznih vrednosti na skup realnih brojeva.

Eksponencijalna, logaritamska i stepena funkcija

```
>> log(2)           >> log10(2)
>> sqrt(-1)         >> nthroot(-2, 3)
>> nthroot(2, 3)     >> nthroot(-2, 4)
>> realpow(i, 2)
>> sqrt([149])
```

Funkcije *reallog* i *realsqrt* prihvataju za argumente samo nizove pozitivnih realnih brojeva. Funkcija *realpow* ne ograničava mogući skup ulaznih vrednosti, ali zato redukuje skup mogućih izlaznih vrednosti na skup realnih brojeva.

Pomenute funkcije prihvataju i matrice kao argumente i deluju nad svakim pojedinačnim elementom matrice.

Dvodimenzionalni grafici

Grafik funkcije se može dobiti korišćenjem funkcije *plot*.

Dvodimenzionalni grafici

Grafik funkcije se može dobiti korišćenjem funkcije *plot*.

```
>> x = -10 : .1 : 10;           >> y = sin(x);  
>> plot(x, y);  
>> title('Grafik funkcije sin');  
>> xlabel('x');                 >> ylabel('sin(x)');  
>> gridon;
```

Dvodimenzionalni grafici

Grafik funkcije se može dobiti korišćenjem funkcije *plot*.

```
>> x = -10 : .1 : 10;           >> y = sin(x);  
>> plot(x, y);  
>> title('Grafik funkcije sin');  
>> xlabel('x');                 >> ylabel('sin(x)');  
>> gridon;
```

Prozor za iscrtavanje grafika se otvara iznad ostalih prozora i ima svoju naslovnu liniju, svoj meni i svoju radnu površinu.

Kreiranje prozora za crtanje

Drugi prozor za crtanje možemo obezbediti koristeći funkciju *figure*.

Kreiranje prozora za crtanje

Drugi prozor za crtanje možemo obezbediti koristeći funkciju *figure*.

```
>> x = -10 : .1 : 10;  
>> z = cos(x); >> y = sin(x);  
>> plot(x, y), title('Grafik funkcije sin'), xlabel('x');  
>> ylabel('sin(x)'), grid on;  
>> figure(2);  
>> plot(x, z), title('Grafik funkcije cos'), xlabel('x');  
>> ylabel('cos(x)'), grid on;
```

Kreiranje prozora za crtanje

Drugi prozor za crtanje možemo obezbediti koristeći funkciju *figure*.

```
>> x = -10 : .1 : 10;  
>> z = cos(x);    >> y = sin(x);  
>> plot(x, y),    title('Grafik funkcije sin'),    xlabel('x');  
>> ylabel('sin(x)'),    grid on;  
>> figure(2);  
>> plot(x, z),    title('Grafik funkcije cos'),    xlabel('x');  
>> ylabel('cos(x)'), grid on;
```

Naredba *figure(i)* omogućava vidljivost prozora za crtanje sa indeksom *i*, i u isto vreme, prozor za crtanje sa indeksom *i* ima fokus, i menja se pri delovanju komandi koje se odnose na crtanje.

Crtanje više grafika u jednom prozoru

Naredba *hold* sa argumentom *on* omogućava da se stari grafik zadrži, iako se crta novi. Ovo je moguće promeniti pozivom funkcije *hold* sa argumentom *off*.

Crtanje više grafika u jednom prozoru

Naredba *hold* sa argumentom *on* omogućava da se stari grafik zadrži, iako se crta novi. Ovo je moguće promeniti pozivom funkcije *hold* sa argumentom *off*.

```
>> x = -10 : .1 : 10;  
>> z = cos(x); >> y = sin(x);  
>> plot(x,y), title('Grafici funkcija sin i cos'), xlabel('x');  
>> ylabel('sin(x), cos(x)'), grid on;  
>> hold on  
>> plot(x,z)
```

Crtanje više grafika u jednom prozoru

Naredba *hold* sa argumentom *on* omogućava da se stari grafik zadrži, iako se crta novi. Ovo je moguće promeniti pozivom funkcije *hold* sa argumentom *off*.

```
>> x = -10 : .1 : 10;  
>> z = cos(x); >> y = sin(x);  
>> plot(x, y), title('Grafici funkcija sin i cos'), xlabel('x');  
>> ylabel('sin(x), cos(x)'), grid on;  
>> hold on  
>> plot(x, z)
```

Grafici su obojeni istom bojom.

Crtanje više grafika u jednom prozoru

Crteže koji sadrže više od jednog grafika moguće je dobiti i navođenjem više argumenata funkcije *plot*.

Crtanje više grafika u jednom prozoru

Crteže koji sadrže više od jednog grafika moguće je dobiti i navođenjem više argumenata funkcije *plot*.

```
>> hold off;  
>> x = -10 : .1 : 10;  
>> z = cos(x); >> y = sin(x);  
>> plot(x, y, x, z), >> title('Grafici funkcija sin i cos'),  
>> xlabel('x');, >> ylabel('sin(x), cos(x)'),  
>> grid on;
```

Crtanje više grafika u jednom prozoru

Crteže koji sadrže više od jednog grafika moguće je dobiti i navođenjem više argumenata funkcije *plot*.

```
>> hold off;  
>> x = -10 : .1 : 10;  
>> z = cos(x); >> y = sin(x);  
>> plot(x, y, x, z), >> title('Grafici funkcija sin i cos'),  
>> xlabel('x');, >> ylabel('sin(x), cos(x)'),  
>> grid on;
```

Grafici su obojeni različiti bojama.

Crtanje više grafika u jednom prozoru

Crteže koji sadrže više od jednog grafika moguće je dobiti i navođenjem matrice kao argumenta funkcije *plot*.

Crtanje više grafika u jednom prozoru

Crteže koji sadrže više od jednog grafika moguće je dobiti i navođenjem matrice kao argumenta funkcije *plot*.

```
>> hold off;  
>> x = -10 : .1 : 10;  
>> u = [cos(x); y = sin(x)];  
>> plot(x, u)  
>> title('Grafici funkcija sin i cos'), >> xlabel('x');  
>> ylabel('sin(x), cos(x)'), >> grid on;
```

Još argumenata funkcije *plot*

- Argumenti koji su povezani sa formatom i bojom linije se zadaju u obliku stringa.

Još argumenata funkcije *plot*

- Argumenti koji su povezani sa formatom i bojom linije se zadaju u obliku stringa.
- Različite vrste linija: `'-'`, `'.'`, `'-.'`, `'--'`.

Još argumenata funkcije *plot*

- Argumenti koji su povezani sa formatom i bojom linije se zadaju u obliku stringa.
- Različite vrste linija: `'—'`, `'.'`, `'-.'`, `'--'`.
- Tačke iz kojih se sastoji linija prilikom iscrtavanja grafika mogu biti prikazane kao tačka, kružić, krstić, plus, zvezda, kvadrat, dijamant, obrnuti trougao, trougao, levi trougao, desni trougao, pentagram i heksagram, korišćenjem `'.'`, `'o'`, `'x'`, `'+'`, `'*'`, `'s'`, `'d'`, `'v'`, `'^'`, `'<'`, `'>'`, `'p'`, `'h'`, redom.

Još argumenata funkcije *plot*

- Argumenti koji su povezani sa formatom i bojom linije se zadaju u obliku stringa.
- Različite vrste linija: `'—'`, `' : '`, `' - . '`, `' - - '`.
- Tačke iz kojih se sastoji linija prilikom iscrtavanja grafika mogu biti prikazane kao tačka, kružić, krstić, plus, zvezda, kvadrat, dijamant, obrnuti trougao, trougao, levi trougao, desni trougao, pentagram i heksagram, korišćenjem `'.'`, `'o'`, `'x'`, `'+'`, `'*'`, `'s'`, `'d'`, `'v'`, `' '`, `'<'`, `'>'`, `'p'`, `'h'`, redom.
- Boja linija može biti plava, zelena, crvena, cijan, ljubičasta, žuta i crna, korišćenjem argumenata funkcije `plot`: `'b'`, `'g'`, `'r'`, `'c'`, `'m'`, `'y'`, `'k'`, redom.

Još argumenata funkcije *plot*

Još argumenata funkcije *plot*

```
>> x = -10 : .1 : 10;  
>> z = cos(x); >> y = sin(x);  
>> plot(x, y, ' - - + c', x, z, ' : ''');  
>> title('Grafici funkcija sin i cos crtani stilovima' ...  
' - - + c' i ' : ''', 'redom');  
>> xlabel('x'), ylabel('sin(x), cos(x)'), grid on;
```

Još argumenata funkcije *plot*

Na grafiku se može prikazati i legenda, korišćenjem funkcije *legend*.

Još argumenata funkcije *plot*

Na grafiku se može prikazati i legenda, korišćenjem funkcije *legend*.
Funkcija *text* ubacuje tekst na određenom mestu u grafiku.

Još argumenata funkcije *plot*

Na grafiku se može prikazati i legenda, korišćenjem funkcije *legend*. Funkcija *text* ubacuje tekst na određenom mestu u grafiku.

```
>> x = -10 : .1 : 10;  
>> y = sin(x); >> y1 = sin(2 * x); >> y2 = sin(3 * x);  
>> plot(x, y, '—', x, y1, '—', x, y2, '—');  
>> title('Grafik funkcije sin(ax), za a = 1, a = 2 i a = 3');  
>> xlabel('x'), ylabel('sin(a * x)');  
>> legend('a = 1', 'a = 2', 'a = 3');  
>> text(0, 0, 'ovo je tekst');
```

Razmera

Naredba *axis* ima dva formata jedan bez argumenata koji za svako naredno iscrtavanje zadržava trenutnu razmeru i *axis([xmin, xmax, ymin, ymax])* koja eksplicitno postavlja razmeru prozora za iscrtavanje.

Razmera

Naredba *axis* ima dva formata jedan bez argumenata koji za svako naredno iscrtavanje zadržava trenutnu razmeru i *axis([xmin, xmax, ymin, ymax])* koja eksplicitno postavlja razmeru prozora za iscrtavanje.

```
>> x = -10 : .1 : 10;  
>> y = sin(x);  
>> axis([-100, 100, -5, 5]);  
>> hold on;  
>> plot(x, y);
```

Podgrafici

- Prikazivanje više dijagrama na jedinstvenom prozoru za slike vrši se naredbom *subplot*.

Podgrafici

- Prikazivanje više dijagrama na jedinstvenom prozoru za slike vrši se naredbom *subplot*.
- Funkcija *subplot* ima tri argumenta. Prvi i drugi argument sadrže tip podele prozora za crtanje, tako što prvi argument predstavlja broj vrsta u dijagramu, a drugi broj kolona u dijagramu.

Podgrafici

- Prikazivanje više dijagrama na jedinstvenom prozoru za slike vrši se naredbom *subplot*.
- Funkcija *subplot* ima tri argumenta. Prvi i drugi argument sadrže tip podele prozora za crtanje, tako što prvi argument predstavlja broj vrsta u dijagramu, a drugi broj kolona u dijagramu.
- Treći argument predstavlja poziciju dijagrama koji prima fokus. Pozicije dijagrama se računaju tako što prvi levi dijagram ima vrednost 1, a indeksi rastu sa leva na desno i odozgo na dole.

Podgrafici

- Prikazivanje više dijagrama na jedinstvenom prozoru za slike vrši se naredbom *subplot*.
- Funkcija *subplot* ima tri argumenta. Prvi i drugi argument sadrže tip podele prozora za crtanje, tako što prvi argument predstavlja broj vrsta u dijagramu, a drugi broj kolona u dijagramu.
- Treći argument predstavlja poziciju dijagrama koji prima fokus. Pozicije dijagrama se računaju tako što prvi levi dijagram ima vrednost 1, a indeksi rastu sa leva na desno i odozgo na dole.
- Ako smo koristili funkciju *subplot*, prilikom sledećeg crtanja Matlab podrazumeva da treba da zadrži podelu prozora za crtanje, zato je, prilikom sledećeg crtanja, neophodno pozvati *subplot(1,1,1)* da bi prozor za crtanje prikazivao samo jedan grafik.

Podgrafici

```
>> x = -10 : .1. : 10;  
>> y = x;  
>> plot(x, y);  
>> y = x2;  
>> plot(x, y);  
>> y = x3;  
>> plot(x, y);  
>> y = x4;  
>> plot(x, y);  
  
>> subplot(2, 2, 1);  
>> title('Funkcija x');  
>> subplot(2, 2, 2);  
>> title('Funkcija x{2}');  
>> subplot(2, 2, 3);  
>> title('Funkcija x{3}');  
>> subplot(2, 2, 4);  
>> title('Funkcija x{4}');
```

Podgrafici

```
>> x = -10 : .1 : 10;  
>> y = x;  
>> plot(x, y);  
>> y = x2;  
>> plot(x, y);  
>> y = x3;  
>> plot(x, y);  
>> y = x4;  
>> plot(x, y);  
  
>> subplot(2, 2, 1);  
>> title('Funkcija x');  
>> subplot(2, 2, 2);  
>> title('Funkcija x2');  
>> subplot(2, 2, 3);  
>> title('Funkcija x3');  
>> subplot(2, 2, 4);  
>> title('Funkcija x4');
```

Za eksponent se koriste vitičaste zagrade.

Razne vrste grafika

- Dve funkcije crtaju prugaste dijagrame, i to su funkcije *bar* i *barh*. Prva funkcija crta prugasti dijagram tako što nanosi vrednosti po y-osi, a druga tako što nanosi vrednosti po x-osi.

Razne vrste grafika

- Dve funkcije crtaju prugaste dijagrame, i to su funkcije *bar* i *barh*. Prva funkcija crta prugasti dijagram tako što nanosi vrednosti po *y*-osi, a druga tako što nanosi vrednosti po *x*-osi.
- Funkcija *pie* crta grafik u obliku kruga, pri čemu se vrednosti pojedinih koordinata vektora argumenta tumače kao vrednosti dela površine kruga, naravno, u relativnom odnosu.

Razne vrste grafika

- Dve funkcije crtaju prugaste dijagrame, i to su funkcije *bar* i *barh*. Prva funkcija crta prugasti dijagram tako što nanosi vrednosti po y-osi, a druga tako što nanosi vrednosti po x-osi.
- Funkcija *pie* crta grafik u obliku kruga, pri čemu se vrednosti pojedinih koordinata vektora argumenta tumače kao vrednosti dela površine kruga, naravno, u relativnom odnosu.
- Histogrami su posebna vrsta grafika koja u stvari predstavlja zavisnost broja ponavljanja elementa u vektoru u funkciji od vrednosti elementa. Histogrami se crtaju funkcijom *hist*.

Razne vrste grafika

```
>> x = [1 : 10; 2 : 2 : 20];  
>> subplot(2,2,1);  
>> grid on;  
>> subplot(2,2,2);  
>> grid on;  
>> subplot(2,2,3);  
>> grid on;  
>> subplot(2,2,4);  
>> x = [1 1 1 1 1 2 2 2 2 2 2 2 2];  
>> hist(x);  
  
>> bar(x);  
>> title('bar');  
>> barh(x);  
>> title('barh');  
>> pie(x(1,:));  
>> title('pie');  
  
>> title('hist');
```

Crtanje funkcija

Umjesto konkretnih x i y vrednosti u MATLABu je moguće upotrebiti i funkciju u *plot* komandi, koristeći *fplot* instrukciju.

Crtanje funkcija

Umjesto konkretnih x i y vrednosti u MATLABu je moguće upotrebiti i funkciju u *plot* komandi, koristeći *fplot* instrukciju.

```
fplot('sin(x)', [-2 * pi, 2 * pi])
```

Crtanje linija

- Prilikom crtanja trodimenzionalnih grafika možemo crtati linije, mrežaste strukture i grafike površi.

Crtanje linija

- Prilikom crtanja trodimenzionalnih grafika možemo crtati linije, mrežaste strukture i grafike površi.
- Za crtanje linija se koristi funkcija *plot3*.

Crtanje linija

- Prilikom crtanja trodimenzionalnih grafika možemo crtati linije, mrežaste strukture i grafike površi.
- Za crtanje linija se koristi funkcija *plot3*.
- Ova funkcija prihvata tri argumenta (vektori): prvi argument su x koordinate, drugi y koordinate a treći z koordinate tačaka koje se iscrtavaju.

Crtanje linija

- Prilikom crtanja trodimenzionalnih grafika možemo crtati linije, mrežaste strukture i grafike površi.
- Za crtanje linija se koristi funkcija *plot3*.
- Ova funkcija prihvata tri argumenta (vektori): prvi argument su x koordinate, drugi y koordinate a treći z koordinate tačaka koje se iscrtavaju.
- Elipsu

$$y = -x, \quad \frac{3x^2}{2} + \frac{z^2}{3} = 1,$$

napišemo u parametarskom obliku

$$x = \sqrt{\frac{2}{3}} \cos t, \quad y = -\sqrt{\frac{2}{3}} \cos t, \quad z = \sqrt{3} \sin t, \quad t \in [0, 2\pi].$$

Crtanje linija

```
>> t = 0 : .1 : 2 * pi;  
>> x = sqrt(2/3) * cos(t);  
>> y = -x;  
>> z = sqrt(3) * sin(t);  
>> plot3(x, y, z);  
>> grid on;  
>> xlabel('x'), ylabel('y'), zlabel('z');  
>> title('Grafik ellipse y = -x, 3/2x^2 + z^2/3 = 1');
```

Crtanje linija

```
>> t = 0 : .1 : 2 * pi;  
>> x = sqrt(2/3) * cos(t);  
>> y = -x;  
>> z = sqrt(3) * sin(t);  
>> plot3(x, y, z);  
>> grid on;  
>> xlabel('x'), ylabel('y'), zlabel('z');  
>> title('Grafik elipse  $y = -x$ ,  $3/2x^2 + z^2/3 = 1$ ');
```

Sve opcije koje smo koristili pri crtanju funkcijom *plot*, važe i prilikom crtanja funkcijom *plot3*.

Crtanje linija

- Funkcija *linspace* ima tri argumenta.

Crtanje linija

- Funkcija *linspace* ima tri argumenta.
- Prvi je polazna tačka, drugi je krajnja tačka, a treći broj tačaka koje treba postaviti između.

Crtanje linija

- Funkcija *linspace* ima tri argumenta.
- Prvi je polazna tačka, drugi je krajnja tačka, a treći broj tačaka koje treba postaviti između.
- U slučaju da treći argument nije naveden koristi se 100 kao podrazumevana vrednost.

Crtanje mrežastih grafika

- Koristi se za crtanje grafika funkcije $z = f(x, y)$.

Crtanje mrežastih grafika

- Koristi se za crtanje grafika funkcije $z = f(x, y)$.
- Neophodno je zadati tri matrice: jednu koja predstavlja x koordinate svih tačaka u kojima su vrednosti funkcije z poznate, drugu matricu svih y koordinata pomenutih tačaka, i na kraju, same vrednosti funkcije z u ovim tačkama.

Crtanje mrežastih grafika

- Koristi se za crtanje grafika funkcije $z = f(x, y)$.
- Neophodno je zadati tri matrice: jednu koja predstavlja x koordinate svih tačaka u kojima su vrednosti funkcije z poznate, drugu matricu svih y koordinata pomenutih tačaka, i na kraju, same vrednosti funkcije z u ovim tačkama.

```
>> x = [1 2 3; 1 2 3; 1 2 3];  
>> y = [1 1 1; 2 2 2; 3 3 3];  
>> z = y;  
>> mesh(x, y, z);  
>> xlabel('x'), ylabel('y'), zlabel('z');  
>> title('grafik ravni z = y');
```

Crtanje mrežastih grafika

- Grafik iste ravni, ali sa drugačijim matricama.

Crtanje mrežastih grafika

- Grafik iste ravni, ali sa drugačijim matricama.

```
>> x = [1 2 3; 1 2 3; 1 2 3];  
>> y = [1 3 1; 2 2 2; 3 1 3];  
>> z = y;  
>> mesh(x, y, z);  
>> xlabel('x'), ylabel('y'), zlabel('z');  
>> title('grafik ravni z = y, los raspored tacaka');
```

Crtanje mrežastih grafika

- Prilikom crtanja grafika na uniformnim podelama koordinatnih osa, može se koristiti funkcija *linspace* i funkcija *meshgrid*.

Crtanje mrežastih grafika

- Prilikom crtanja grafika na uniformnim podelama koordinatnih osa, može se koristiti funkcija *linspace* i funkcija *meshgrid*.
- Funkcija *meshgrid* od dva vektora, x i y , tipa $1 \times n$ i $1 \times m$, redom, formira dve matrice. Obe matrice su tipa $m \times n$, a predstavljaju odgovarajuće matrice za crtanje grafika funkcije nad podelom koja je Dekartov proizvod tačaka u vektorima x i y .

Crtanje mrežastih grafika

- Prilikom crtanja grafika na uniformnim podelama koordinatnih osa, može se koristiti funkcija *linspace* i funkcija *meshgrid*.
- Funkcija *meshgrid* od dva vektora, x i y , tipa $1 \times n$ i $1 \times m$, redom, formira dve matrice. Obe matrice su tipa $m \times n$, a predstavljaju odgovarajuće matrice za crtanje grafika funkcije nad podelom koja je Dekartov proizvod tačaka u vektorima x i y .

```
>> x = linspace(1, 10, 10);  
>> y = linspace(11, 30, 20);  
>> [x y] = meshgrid(x, y)  
>> z = x.^2 + y.^2;  
>> mesh(x, y, z);  
>> xlabel('x'), ylabel('y'), zlabel('z');
```

Crtanje mrežastih grafika

- Funkcija *view* ima više formata.

Crtanje mrežastih grafika

- Funkcija *view* ima više formata.
- Jedan format prihvata vektor tipa 1×3 , koji po izvršenju funkcije *view* predstavlja novi pravac iz koga se posmatra grafik.

Crtanje mrežastih grafika

- Funkcija *view* ima više formata.
- Jedan format prihvata vektor tipa 1×3 , koji po izvršenju funkcije *view* predstavlja novi pravac iz koga se posmatra grafik.
- Primer poziva funkcije `>> view([1 1 0]);`.

Crtanje površi

- Kada je potrebno prikazati grafike funkcija kao jasne površi.

Crtanje površi

- Kada je potrebno prikazati grafike funkcija kao jasne površi.
- Koristimo funkciju *surf*.

Crtanje površi

- Kada je potrebno prikazati grafike funkcija kao jasne površi.
- Koristimo funkciju *surf*.
- Funkcija *surf* očekuje iste podatke kao funkcija *mesh*.

Crtanje površi

- Kada je potrebno prikazati grafike funkcija kao jasne površi.
- Koristimo funkciju *surf*.
- Funkcija *surf* očekuje iste podatke kao funkcija *mesh*.

```
>> x = [1 2 3; 1 2 3; 1 2 3];  
>> y = [1 1 1; 2 2 2; 3 3 3];  
>> z = y; >> surf(x, y, z);  
>> xlabel('x'), ylabel('y'), zlabel('z');  
>> figure(2);  
>> y = [1 3 1; 2 2 2; 3 1 3];  
>> z = y; >> surf(x, y, z);  
>> xlabel('x'), ylabel('y'), zlabel('z');
```

Senčenje i boja

- Za način senčenja koristimo funkciju *shading*.

Senčenje i boja

- Za način senčenja koristimo funkciju *shading*.
- Predefinisana vrednost senčenja je *faceted*.

Senčenje i boja

- Za način senčenja koristimo funkciju *shading*.
- Predefinisana vrednost senčenja je *faceted*.
- Drugi način senčenja se dobija naredbom *shading interp*.

Senčenje i boja

- Za način senčenja koristimo funkciju *shading*.
- Predefinisana vrednost senčenja je *faceted*.
- Drugi način senčenja se dobija naredbom *shading interp*.
- Treći način senčenja je ravno sečenje, koje se dobija naredbom *shading flat*.

Senčenje i boja

- Za način senčenja koristimo funkciju *shading*.
- Predefinisana vrednost senčenja je *faceted*.
- Drugi način senčenja se dobija naredbom *shading interp*.
- Treći način senčenja je ravno sečenje, koje se dobija naredbom *shading flat*.
- Naredba *colormap* određuje boje kojima se iscrtava grafik.

Senčenje i boja

- Za način senčenja koristimo funkciju *shading*.
- Predefinisana vrednost senčenja je *faceted*.
- Drugi način senčenja se dobija naredbom *shading interp*.
- Treći način senčenja je ravno sečenje, koje se dobija naredbom *shading flat*.
- Naredba *colormap* određuje boje kojima se iscrtava grafik.
- Predefinisane vrednosti argumenta funkcije *colormap* su *autumn*, *spring*, *summer*, *winter*, *jet*, *bone*, *colorcube*, *cool*, *copper*, *flag*, *hot*, *hsv*, *pink*, *prism*, *white* i *gray*.