

- Primjena računara
- **TEMA:** Radno okruženje Matlab-a
- Marina Lukić, prof.

MATLAB, Mathematica, Maple i dr.

- Moćni alati za matematička izračunavanja
- Poseduju moćne matematičke naredbe
- Pružaju velike mogućnosti za generisanje grafika
- Kombinacija moćnih funkcija za izračunavanja i moćnih funkcija vizuelizacije čini ih posebno korisnim alatima za inženjere

Ako inženjerski problem može da se reši korišćenjem softverskog alata, obično je efikasnije koristiti softverski alat nego pisati program korišćenjem programskog jezika

Međutim, mnogi problemi ne mogu da se reše korišćenjem softverskih alata

Imajući ovo u vidu nameće se potreba da znamo da pišemo programme korišćenjem programskih jezika

MATrična LABoratoriјa

MATLAB je softverski alat, ali ima i sopstveni programski jezik.

Startovanje Matlab-a

Na Windows platformi MATLAB startujemo izabiranjem Windows Start > Programs > MATLAB 7.1 > MATLAB 7.1, ili dvostrukim klikom na ikonu na radnoj površini Windows-a.



Nakon toga se otvara **radna površina Matlab-a (MATLAB Desktop)**.

Kada se MATAB staruje, radna površina MATLAB-a se pojavljuje sa nekoliko podprozora, menijem i radnom linijom. Svi prozori su prikačeni pa površinu MATLAB-a.

Mogu da se otkače sa izborom stavke menija  Desktop > Undock ili klikom na dugme .

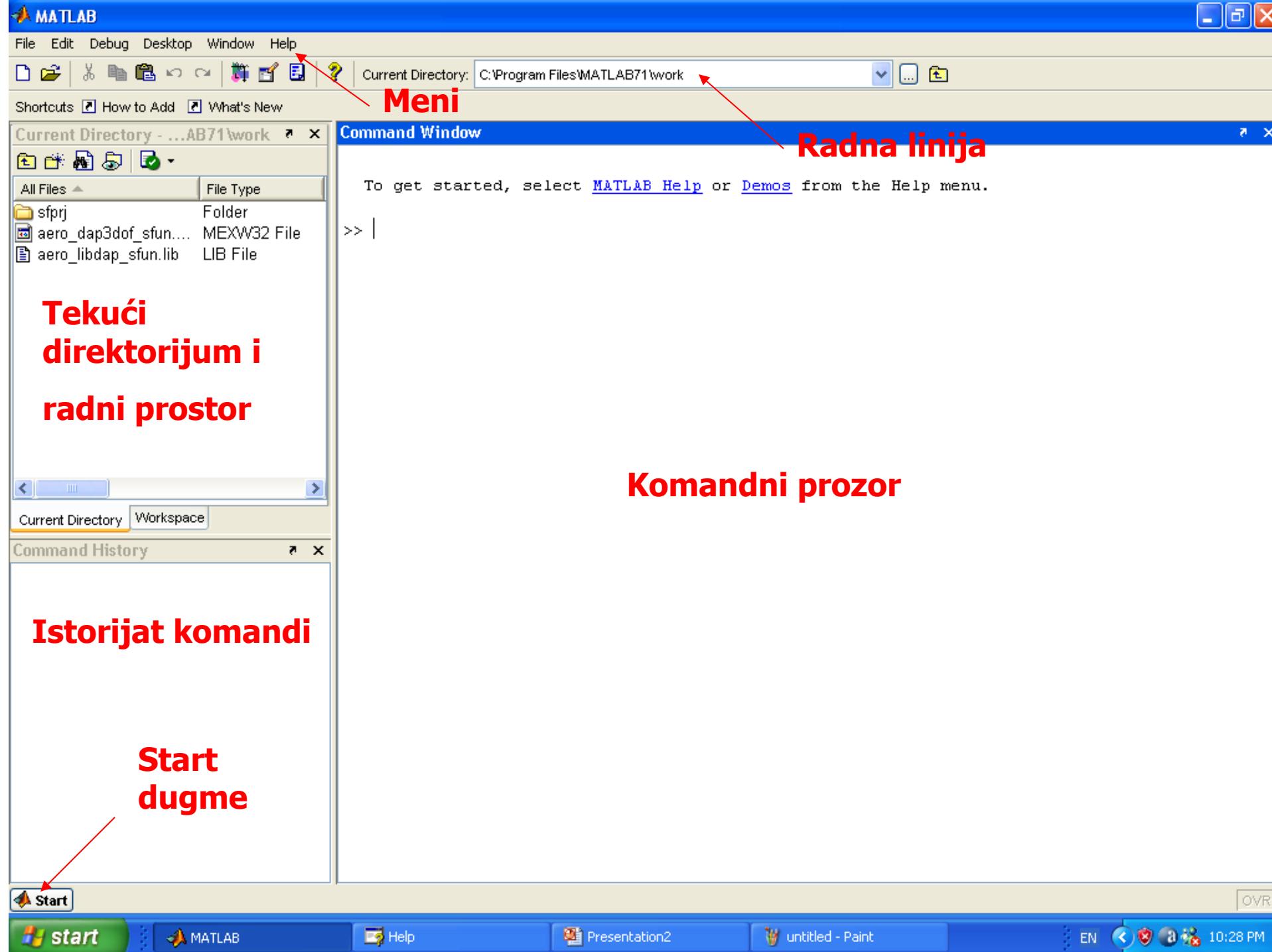
Prozor se ponovo može prikačiti za radnu površinu klikom nadugme .



Prozori se gase klikom na dugme .

Radna površina Matlab-a
sastoji se iz

- MATLAB Start dugmeta
 - Komandnog prozora (Command Window)
 - Istorijata komandi (Command History)
 - Radnog prostora (Workspace)
 - Radnog direktorijuma (Current Directory)
- kao i menija (Menu), radne linije (Toolbar) i prečica (Shortcuts).



MATLAB Start Dugme

Ovo dugme omogućava jednostavan pristup svim alatima, demonstracionim programima i celokupnoj dokumentaciji.

Moguće je napraviti i izvršavati MATLAB prečice koje predstavljaju grupu od više MATLAB naredbi.

Vežba: Pokrenite MATLAB Demos klikom na **Start>MATLAB>Demos**

Komandni prozor

COMMAND WINDOW

Komandni prozor je jedan od glavnih alata za unos podataka, izvršavanje MATLAB funkcija i M-datoteka, kao i za prikaz rezultata.

Prompt komandnog prozora, `>>`, je mesto gde se unose naredbe. Moguće je uneti MATLAB funkcije sa argumentima ili dodeliti vrednosti promenljivama.

Komandni prozor

MATLAB izrazi i naredbe se proveravaju dok ih kucate u komandnom prozoru, a rezultati se izračunavaju i prikazuju. Najčešće su u sledećem obliku:

varijabla=izraz ili samo **izraz**.

Izrazi se sastoje od operatora, funkcija i imena promenljivih. Izvršavanje izraza za rezultat daje matricu (ili drugi tip podatka), koji se nakon toga prikazuje na ekranu ili dodeljuje promenljivoj za buduću upotrebu. Ukoliko se ime promenljive ili znak = izostave promenljiva **ans** (answer-odgovor) se automatski stvara i dodeljuje joj se rezultat.

Komandni prozor

U komandnom prozoru izračunavanje se vrši na sličan način kao na kalkulatoru.

Da bismo izračunali 10^2 , unosimo naredbu

```
>>10^2
```

Prikazaće se sledeći rezultat

ans =

100

Da bi smo izračunali vrednost $\cos()$ unosimo

```
>>cos(pi)
```

Prikazaće se sledeći rezultat

ans =

-1

Komandni prozor

Kliknite na jezičak Workspace (Radni prostor) da ga prikažete iznad Current Directory (Tekućeg direktorijuma) da bi videli promenljive dok ih kreirate.

Ukucajte ovu naredbu u Komandni prozor:

```
A=[1 2 3 ; 4 5 6 ; -1 7 9]
```

ili

```
A=[1 2 3  
    4 5 6  
    -1 7 9]
```

I jedna i druga naredba kreiraju matricu dimenzija 3x3 i dodeljuju je promenljivoj A. Trebalo bi da vidite matricu A u prozoru radnog prostora.

Komandni prozor sadrži spisak svih naredbi koje ste uneli i te naredbe ne mozemo poništiti.

Međutim, možemo da unesemo određenu naredbu iz spiska naredbi i da izvršimo novu verziju.

To se može uraditi na sledeći način:

1. Da koristimo cursorske tastere (strelica na goire i strelica na dole) da se krećemo kroz listu naredbi koje smo izvršili. Kada pronađemo odgovarajuću naredbu možemo je promeniti i onda izvršiti novu verziju.

2. Da ponovo unesemo naredbu.

Komandna linija u MATLAB-u može lako da se menja u komandnom prozoru. Kursor može da se pozicionira korišćenjem leve ili desne strelice ili Backspace (ili Delete) dugmeta na tastaturi.

Komandni prozor

MATLAB vodi računa o tome da li su komande, funkcije ili varijable pisane velikim ili malim slovima, tako da su **A** i **a** dve različite promenljive. Zapeta ili blanko razdvajaju elemente unutar reda matrice (nekada je zapeta neophodna da bi se razdvojili izrazi zato što blanko znak može da bude dvosmislen). Tačka-zapeta završava red matrice. Kada se pišu brojevi u eksponencijalnom obliku (npr. 2.34e-3) blanko znaci ne smeju da se unose (recimo pre e).

Komandni prozor

Komandni prozor može da se sačuva sa **diary** naredbom na sledeći način:

diary ime_datoteke

Na ovaj način sve ono što se unese u komandni prozor se beleži u imenovanoj datoteci. Ukoliko se ime datoteke ne navede podaci se beleže u datoteku **diary**. Unos u datoteku se prekida ukucavanjem naredbe **diary off**. Komanda **diary on** se koristi da se nastavi sa upisom u datoteku.

Sadržaj komandnog prozora možemo obrisati unosom naredbe **clc** ili izborom stavke menija **Edit > Clear Command Window**.

Prozor radnog prostora WORKSPACE

Prozor radnog prostora prati (lista) promenljive koje su unete ili izračunate u toku MATLAB sesije.

Kada izvršavamo naredbu, radni prostor treba da prikaže jednu promenljivu

`ans`

Prozor radnog prostora

Naredba **who** (ili **whos**) lista promenljive koje se nalaze u radnom prostoru.

Promenjiva ili funkcija može da bude očišćena sa radnog prostora komandom

clear ime_promenljive

ili desnim klikom na promenljivu u prozoru radnog prostora i odabiranjem stavke konteksnog menija **Delete**. Ukoliko unesemo naredbu clear bez imena promenljive ona će obrisati sve promenljive iz radnog prostora.

Prozor hronologije komandi

COMMAND HISTORY

Prozor sa hronologijom sadrži komande koje su do sada kucane. Ove komande mogu ponovo da se izvrše dvostrukim klikom na komandu ili prevlačenjem naredbe u komandni prozor.

Vežba: Kliknite dva puta na naredbu

A=A+1

prikazanoj u komandnom prozoru.

Ukoliko želite više opcija kliknite desnim klikom na naredbu u prozoru hronologije komandi.

PROZOR RADNOG DIREKTORIJUMA

CURRENT DIRECTORY

Kada MATLAB pristupa datotekama ili čuva informacije na računaru, on koristi radni direktorijum.

Radni direktorijum je prikazan na vrhu glavnog prozora.

On se može promeniti biranjem drugog direktorijuma iz padajuće liste koja se nalazi pored liste direktorijuma

Prozor radnog direktorijuma

MATLAB u radnom (tekućem) direktoriju pretražuje M-datoteke, kao i .mat binarne datoteke radnog prostora. Matrice se, takođe, mogu sačuvati i učitati u ASCII formatu. One moraju da budu zadate u vidu kvadratnog niza sa numeričkim vrednostima.

pwd naredba vraća putanju tekućeg direktorijuma, dok sa **cd** naredbom se menja tekući direktorijum. **dir** lista sadržaj tekućeg direktorijuma, dok komanda **what** lista samo datoteke specifične za MATLAB grupisane po tipu podataka. Sa **delete** naredbom je moguće obrisati datoteku, dok **type** naredba prikazuje sadržaj datoteke u komandnom prozoru. Sve ove operacije je moguće uraditi korišćenjem radne linije i stavki glavnog i konteksnog menija.

Prozor editora niza

ARRAY EDITOR

Jedanput kada niz postoji on može da bude modifikovan prozorom editora niza. Editor se ponaša kao neki spreadsheet program (recimo Excel) samo za matrice.

Dvostrukim klikom na promenljivu koja se nalazi u radnom prostoru automatski se pokreće prozor editora niza koji sadrži program za uređivanje nizova. Vrednosti koje su smeštene u promenljivoj prokazuju se u tabelarnom formatu. Te vrednosti možemo menjati ili dodavati nove.

Prozor za pomoć

Ovaj prozor se otvara izborom stavke glavnog menija Help > MATLAB Help ili ukucavanjem naredbe **doc**.

Takođe je moguće koristiti naredbu **help** koja prikazuje pomoć u komandnom prozoru.

TEMA: Promenljive, operacije sa skalarima,
operacije sa nizovima

PROMENLJIVE

U MATLAB-u dodeljujemo imena:

- skalarima
- vektorima i matricama

Pravila imenovanja su:

- imena promenljivih moraju početa slovom
- Postoji razlika između mala i velika slova
(vreme i VREME su različite promenljive)
- ostali simboli imena mogu da budu
 - slova
 - cifre
 - donja crta (_).

Test za ispravnost imena može se izvršiti naredbom **isvarname**. Ako je odgovor 1 znači da je ispravno. Ako je odgovor 0 znači da nije ispravno ime.

Na primer:

```
>>isvarname vreme  
ans =  
1           Znači da je ispravno.
```

- Za označavanje imena možemo koristiti bilo koji broj znakova. Međutim, MATLAB koristi samo prvih nekoliko znakova (zavisno od verzije).
 - da bismo videli broj značajnih znakova za datumveriju možemo da koristimo naredbu `namelengtmax`.
- Službene reči se ne mogu koristiti kao imena promenljivih.
(Da bismo videli spisak službenih reči koristimo naredbu `iskeyword`)
- Nije dozvoljeno korišćenje imena ugrađenih funkcija (npr. sin, cos, idr)
(Provera
 - `>>wich sin`
 - vraća
 - `sin is a built in function`

OPERACIJE SA SKALARIMA

Aritmetičke operacije sa dva skalara su:

Operacija	Algebarski oblik	MATLAB oblik
sabiranje	$a+b$	$a+b$
oduzimanje	$a-b$	$a-b$
množenje	$a*b$	$a*b$
deljenje	$\frac{a}{b}$	a/b
stepenovanje	a^b	a^b

$$x = a+b$$

znak jednakosti predstavlja operator dodeljivanja:

$$a+b \rightarrow x$$

$$x = x+1 \quad (x+1 \rightarrow x)$$

Prioritet aritmetičkih operatora:

Prioritet	Operacija
1	zagrade
2	stepenovanje (sleva udesno)
3	množenje, deljenje (sleva udesno)
4	sabiranje, oduzimanje (sleba udesno)

$$y = 0.5 * r * (a + b)$$

Unosimo vrednosti:

$$r=5;$$

$$a=12;$$

$$b=6;$$

Izračunavanje izraza vrši se prema sledećem redosledu:

- 1) $a+b = 12+6=18$
- 2) $0.5 * r = 0.5 * 5 = 2.5$
- 3) $2.5 * 18 = 45$

Rezultat je $y = 45$.

Ako izostavimo zagrade

$$y=0.5 * r * a + b$$

Izračunavanje izraza vrši se prema sledećem redosledu:

- 1) $0.5 * r = 0.5 * 5 = 2.5$
- 2) $2.5 * a = 2.5 * 12 = 30$
- 3) $30 + b = 30 + 6 = 36.$

Rezultat je $y=36$.

Ako je izraz složeniji, poželjno je razdvojiti izraz na nekoliko celina, tako izraz

$$f = \frac{x^3 - 2x^2 - x + 10.5}{x^2 + 0.75x - 20.6}$$

možemo podeliti na sledeće celine:

$$i = x^3 - 2x^2 - x + 10.5;$$

$$k = x^2 + 0.75x - 20.6;$$

$$f = i/k;$$

Napomena: praznine ispred i iza operatora su dozvoljene).

OPERACIJE SA NIZOVIMA

Vektor se može ovako definisati:

Naredbea

$$x = [1 \ 2 \ 3 \ 4 \ 5]$$

vraća

$$x =$$

$$1 \ 2 \ 3 \ 4 \ 5$$

Vektor kolona unosi se na sledeći način:

$$Y =]1;2;3 \ 4;5]$$

ili

$$y = [1
2
3
4
5]$$

Matrica koja sadrži vrste i kolona, može se definisati ovako:

Nardba

$$A = [1 \ 2 \ 3 \ 4; 2 \ 3 \ 4 \ 5; 3 \ 4 \ 5 \ 6]$$

vraća

$$A =$$

$$1 \ 2 \ 3 \ 4$$

$$2 \ 3 \ 4 \ 5$$

$$3 \ 4 \ 5 \ 6$$

Matrice se mogu unositi po vrstama:

$$A = [\quad 1 \ 2 \ 3 \ 4;
2 \ 3 \ 4 \ 5;
3 \ 4 \ 5 \ 6]$$

Jednako raspoređene matrice mogu da se unose na sledeći način:

$$B = 1:5 \text{ ili } B = [1:5] \text{ (zagrad nisu obavezne)}$$

Ova naredba vraća matricu:

$$\begin{aligned} B = \\ 1 & 2 & 3 & 4 & 5 \end{aligned}$$

Podrazumevani inkrement je 1.

Međutim, ako želimo da inkrement bude drgačiji, onda je potrebo da unesemo inkrement između prve i poslednje vrednosti. Na primer

$$C = 1:3:10$$

imkrementirana vrednost biće 3, i vraća se vrednost:

$$\begin{aligned} C = \\ 1 & 4 & 7 & 10 \end{aligned}$$

Matrice mogu da se koriste i sa skalarima.

Ako napišemo

$$D = B + 3$$

tada se vraća vrednost

$$\begin{matrix} D = \\ 4 & 5 & 6 & 7 & 8 \end{matrix}$$

Prethodno definisanu matricu možemo da upotrebimo za definisanje druge matrice. Na primer, naredbe

$$\begin{matrix} A = [1, 3, 5]; \\ C = [8, A]; \end{matrix}$$

vraćaju

$$\begin{matrix} C = \\ 8 & 1 & 3 & 5 \end{matrix}$$

Naredba

$$S = [2 4 6; A]$$

vraća

$$\begin{matrix} S = \\ 2 & 4 & 6 \\ 1 & 3 & 5 \end{matrix}$$

Vrednost u matrici možemo da promenimo, ili da dodamo nove, pozivanjem određene pozicije u matrici.

Naredba

$A(3) = 4;$

menja vrednost trećeg elementa matrice A iz 5 u 4.

Ako u komandnom prozoru ukucamo ime matrice A

`>>A`

MATLAB vraća

$A =$
1 3 4

Matricu možemo da uvećamo definisanjem novih elemenata.

Naredba

$A(4) = 10;$

matricu A proširuje sa tri na četiri elementa.

Ako unesemo naredbu

`>>A(7) = 6;`

Matrica A imaće sedam vrednosti pri čemu će elementi A(5), A(6) imati vrednost 0.

Ako unesemo ime matrice A

`>>A`

vraćaju se vrednosti

1 3 4 5 10 0 0 6

Korišćenje operatora dvotačke

Operator dvotačka : se može koristiti za definisanje novih, kao i za modifikovanje postojećih matrica.

Možemo definisati matrice sa jednako raspoređenim vrednostima. Naredbe

$A = 1:5$

vraća

```
>>A =  
1 2 3 4 5
```

Operator dvotačke možemo da koristimo i za uzimanje podataka iz matrice. Ukoliko pri pozivanju na matricu umesto određenog indeksa koristimo dvotačku, tada ona predstavlja jednu celu vrstu ili kolonu.

Za matricu:

$$A = [1 \ 2 \ 3 \ 4 \ 5; 2 \ 3 \ 4 \ 5; 3 \ 4 \ 5 \ 6 \ 7];$$

Drugu kolonu možemo da izdvojimo iz matrice A naredbom

$$x = A(:,2)$$

koja vraća

$$x =$$

$$\begin{matrix} 2 \\ 3 \\ 4 \end{matrix}$$

Takođe, možemo da izdvojimo vrstu koristeći naredbu

$$y = A(3,:)$$

koja vraća

$$y =$$

$$\begin{matrix} 3 \ 4 \ 5 \ 6 \ 7 \end{matrix}$$

Operator dvotačke možemo da koristimo i u značenju “od vrste_do vrste_”, ili “od kolone_ do kolone_”.

Da bismo izdvojili dve posledje vrste matrice A, unosimo naredbu

$z = A(2:3,:)$ (vraća drugu i treću vrstu)

koja će vratiti

$z =$

2 3 4 5 6
3 4 5 6 7

Takođe, možemo da izdvojimo samo četiri brojne vrednosti u donjem desnom uglu matrice A.

Naredba

$w = A(2:3,4:5)$ (vraća deo 2. i 3. vrste; deo 4. i 5. kolone)

vraća

$w =$

5 6
6 7

Možemo da definišemo praznu matricu

$x = [];$

Ako koristimo ime matrice sa jednom dvotačnkom, npr.
naredba

$A(:)$ ili $x = A(:)$

transformiše matricu A u jednu matričnu kolonu, koja je
sastavljena od svih kolona matrice

ans =	ILI	x =
1		1
2		2
3		3
2		2
3		3
4		4
3		3
4		4
5		5
4		4
5		5
6		6
5		5
6		6
7		7

Pojedinačne vrednosti možemo izdvojiti iz niza na dva načina.

1) Korišćenjem notacije za vrstu i kolonu.

>>A

A =

1	2	3	4	5
2	3	4	5	6
3	4	5	6	7

A(3;4)

ans =

6

2) Možemo koristiti jedan indeks; npr. vrednost u 2. vrsti i 4. koloni matrice A predstavlja element pod indeksom 11. Njega možemo izdvojiti tako što odbrojimo indekse duž odgovarajućih kolona.

A(11)

ans =

5

Množenje i deljenje

Kada je u pitanju množenje i deljenje onda te operacije imaju određeno značenje:

Za množenje svakog elementa posebno, koristimo operator
.*

Na primer

$$A . * B$$

daje sledeće dejstvo:

element broj 1 matrice A množi se sa elementom broj 1 matrice B.
element broj 2 matrice A množi se sa elementom broj 2 matrice B
element broj 3 matrice A množi se sa elementom broj 3 matrice B,
i tako dalje.

Na primer, ako je

$$A = [1 \ 2 \ 3] \text{ i } B = [6 \ 7 \ 8]$$

tada

>>A .* B

vraća

ans =

$$6 \ 14 \ 24$$

Na primer,
pretpostavimo da želimo da pretvorimo uglove u radijane.
Potrebno je najpre da unesemo vrednosti uglova u vidu
matrice (niza)

$$E = [10 \ 25 \ 80 \ 90];$$

Da bismo promenili vrednost uglova u radijane,
potrebno je napisati naredbu:

$$R = E * pi / 180;$$

vraća vrednosti u radijanima.

Operacija transponovanja

Operator transponovanja (\top) zamenjuje vrste kolonama i obratno:

Na primer,

E \top

vraća

ans =

10

25

80

90

Ovaj operator se može korisno upotrebiti za formiranje tabelarnih prikaza. Na primer, ako želimo da napravimo tabelu sa stepenima i radijanima, potrebno je uneti

tabela = [E \top , R \top]

MATLAB će oformiti tabelu sa nazivom tabela u kojoj je kolona broj 1 E \top , a kolona broj 2 je R \top .

tabela =

10.000	0.1745
25.000	0.4363
80.000	1.3963
90.000	1.5708

Formati za prikazivanje brojeva

U MATLAB-u elementi matrice koji su celi brojevi uvek se prikazuju kao celi brojevi.

Vrednosti sa decimalnom tačkom prikazuju se u vidu podrazumevanog formata sa četiri decimale.

Na primer,

$$A = 10$$

vraća

$$A =$$

$$10$$

Na primer,

$$A = 8.5$$

vraća

$$A =$$

$$8.5000$$

$$A = 75.6$$

vraća

$$A =$$

$$75.6000$$

U MATLAB-u se mogu koristiti i drugi formati za prikazivanje više značajnih cifara.

Tako, za prikazivanje vrednosti sa 14 decimala koristimo naredbu

format long

koja važi za sve naredne unose.

Na primer vrednost A=75.6 sada postaje:

>>A

A =
75.5999999999999

Ako hoćemo da vratimo format na četiri decimale potrebno je da aktiviramo naredbu

format short

>>A

A =
75.6000

Naučna notacija

Kada broj postane suvše velik ili premali da bi ga MATLAB prikazao korišćenjem podrazumevanog formata, program ga automatski prikazuje u naučnoj notaciji.

U ovoj notaciji brojevi se prikazuju u vidu mantise i eksponenta.

$$m10^e \quad (1 \leq m \leq 10)$$

Na primer konstantu

$$a = 1560000000000000000000000000$$

program vraća

$$\begin{aligned} a = \\ 1.5600e+23 \end{aligned}$$

Na primer.

`>>1.5e-10`
prikazuje kao

$$\begin{aligned} A = \\ 1.5000e-010 \end{aligned}$$

Ova notacija je veoma pogodan način za predstavljanje veoma velikih ili veoma malih brojeva.

Možemo primorati MATLAB da prikazuje sve brojeve u naučnoj notaciji

sa naredbom

format short e (sa 5 značajnih cifara i 4 decimalne)

ili sa naredbom

format long e (sa 16 značajnih cifara i 15 decimala)

Na primer

format short e

x = 15.356789

vraća

x =

1.5357e+001 (5 značajnih cifara i 4 decimalne)

TEMA: Učitavanje i čuvanje MAT datoteka

ČUVANJE PODATAKA

Kada radimo u komandnom prozoru, pri napuštanju MATLAB-a, gubi se sve što smo prethodno radili.

Postoji mogućnost da sačuvamo vrednosti promenljivih koje smo prethodno definisali u komandnom prozoru, kao i onih koje su prikazane u radnom prozoru.

Memorisanje promenljivih

Da bismo sačuvali promenljive koje smo uneli u komandnom prozoru (proveriti RADNI PROZOR sa leve strane MATLAB-ovog ekrana, gde se nalazi spisak promenljivih) između sesija, moramo da upišemo sadržaj RADNOG PROZORA u datoteku.

Podrazumevani format je binarna MAT datoteka.

Da bismo sačuvali **radni prostor** (treba naglasiti da je to samo skup promenljivih a ne i spisak naredbi u komandnom prozoru) u datoteci, potrebno je uneti

`save ime_datoteke`

`save` je naredba MATLAB-a

`ime_datoteke` je ime datoteke koje korisnik definiše.

Ukoliko se ne navede ime datoteke MATLAB daje ime

`matlab.mat`.

i upisuje datoteku u direktorijum sa putokazom:

`D:\Program Files\MATLAB7\work`

Takođe, možemo koristiti sledeću opciju, iz linije menija:

`File`

`Save Workspace As`

pri čemu MORAMO da unesemo ime datoteke.

Možemo da upisujemo pojedinačne veličine iz spiska promenljivih u radni direktorijum, korišćenjem naredbe

save ime_datoteke lista_promenljivih

gde je:

ime_datoteke

 – ime koje definiše korisnik gde želi da upiše veličine

lista_promenljivih

 – spisak promenljivih koje se upisuju u datoteku.

Na primer, naredbom

save podaci x y z

upisujemo u datoteku čije je ime **podaci** samo veličine x, y, z.

Prekid rada sa MATLAB-om

1. Za prekid rada sa MATLAB-om koristimo sledeće dve naredbe koje unosimo u komandnoj liniji:

`>>exit`

ili

`>>quit.`

2. Takođe, možemo da završimo rad u MATLAB-u ako iz File menija odaberemo opciju

`EXIT MATLAB`

3. Prekid rada u MATLAB-u možemo ostvariti ako aktiviramo ikonu za zatvaranje prozora (označenu sa X) u gornjem desnom uglu ekrana.

4. Prekid rada, ali bez izlaska iz MATLAB-a može se ostvariti naredbom

`<ctrl> + C.`

Kada se ponovo uključujemo u MATLAB, radni prostor se može popuniti ranije sačuvanim podacima. Ovo se postiže naredbom

load

koja predstavlja inverznu funkciju od naredbe save.

Tako naredba,

>>load

unosi u radni prostor podatke iz datoteke matlab.mat.

Isto dejstvo imala bi i naredba

>>load matlab.mat

Naredba

>>load ime_datoteke

puni radni prostor iz datoteke ime_datoteke.mat.

Na primer.

>>load podaci x y z (unos: x z y nije obavezan)

ili

>>load podaci (ima isto dejstvo kao i gornja naredba)

Naredbu load možemo koristiti za učitavanje podataka iz ASCII datoteke.

Primer:

Uneti vektore $x = [1 \ -2 \ 3]$ $y = [-3 \ 4 \ -6]$

i matricu $z = \begin{bmatrix} -1 & 2 & 4 \\ 2 & 4 & 6 \end{bmatrix}$

Potrebno je sačuvati podatke u datoteku sa imenom
podaci

```
>>x=[1,-2,3];
>>y=[-3,4,-6];
>>z=[-1,2,4;2,4,6];
>>save podaci
```

Izlazimo iz MATLAB-a naredbom

```
>>quit.
```

Da bismo uneli u radni prostor veličine x i z,
potrebno je ponovo da stratujemo MATLAB i da
unesemo naredbu

```
>>load podaci x z  (y nije uneto u radni prostor)
```

Naredba

>>clear

briše radni prostor

Naredba

>>clc

briše komandni prozor

Naredbe, kada se koriste zajedno

>>clear,clc

brišu sve (i radni prostor i komandni prozor).

Ako se sačuvani podaci koriste u nekom programu koji nije MATLAB (na primer C ili C++, format “.mat“ nije pogodan zato što su datoteke sa sufiksom “.mat“ jedinstvene za MATLAB).

ASCII format je standardni format za razmenu između računarskih platformi i pogodniji je ako imamo potrebu da razmenjujemo podatke.

MATLAB dopušta da sačuvamo datoteku kao ASCII datoteku korišćenjem naredbe

```
>>save ime_datoteke spisak_promenljivih -ascii
```

Naredba ascii ukazuje da će MATLAB upisati podatke u standardni format terksta sa 8 cifara. ASCII datoteke treba da se sačuvaju u datoteci sa sufiksom “.dat”, pri čemu se mora pored imena uneti sufiks “.dat”.

Ako je potrebna veća tačnost, podaci se mogu memorisati u formati sa 16 cifara, korišćenjem sledeće komande:

```
>>save ime_datoteke spisak_promenljivih -ascii -double
```

Moguće je elemente (brojeve) ograničiti tabulatorima:

```
>>save ime_datoteke spisak_promenljivih -ascii -double -tabs
```

Naredbom load možemo preuzeti vrednosti iz radnog direktorijuma.

>>load ime_datoteke

Na primer, da bismo formirali matricu A i sačuvali je u datoteci sa nazivom ime2.dat u formati sa 8 cifara, koristimo sledeće naredbe:

```
>>A=[2 8 10; 7 5 4];  
>>save ime2.dat -ascii
```

Na osnovu dejstva ove naredbe svaki red matrice A biće isписан у posebnom redu u datoteci.

Da bismo videli datoteku ime2.dat potrebno je kliknuti dvaput на име datoteke у прозору radnog direktorijuma.

Podatke možemo unositi iz drugih programa, kao i Excel-ove tabele koristeći istu tehniku

load ime_datoteke

ili možemo odfabri opciju

File

Import Data...

iz linije menija.

Naredba za komentar je znak procenta (%)

Ceo red može da bude komentar ako na početku linje unesemo znak %.

Možemo uneti komentar posle naredbe, ali to mora biti uneto u istom redu.

>>a=10 %Promenljiva A ima vrednost 10

Elementarne matematičke funkcije:

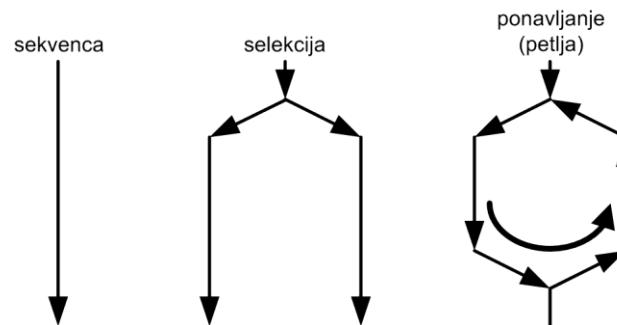
abs(x)	vraća apsolutnu vrednost broja x (abs(-5) ans = 5)
sqrt(x)	vraća kvadratni koren broja x (sqrt(5))
round(x)	zaokružuje broj x na najbliži ceo broj (round(5,7) ans=6)
fix(x)	otseca broj x na najbliži ceo broj (ne zaokružuje) (fix(7.7) ans=7)
floor(x)	zaokružuje broj x na najbliži ceo broj ka negativnoj beskonačnoj vrednosti (floor(-9.6) ans=-10)
ceil(x)	zaokružuje broj x na najbliži ceo broj ka pozitivnoj beskonačnoj vrednosti (ceil(9.6) ans=10)
sign(x)	znak funkcije $\begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$ sign(-5) ans=-1
rem(x,y)	daje ostatak deljenja x/y rem(2,3) ans=2
exp(x)	vraća vrednost e^x exp(10) ans=3.2026e+004
log(x)	vraća $\ln x$ log(10) ans=2.3026
log10(x)	vraća $\log_{10}(x)$ log10(10) ans=1

TEMA: Upravljačke strukture

UPRAVLJČKE STRUKTURE

Delovi računarskiog koda mogu da sdrže neku od sledeće tri strukture:

- sekvenca
- selekcija
- ponavljanje



slika 1. Programske strukture koje se koriste u MATLAB

Sekvence

sekvence su niz naredbi koje se izvršavaju jedna za drugom (sekvencijalno).

Selekcija

Struktura selekcije dozvoljava programeru da izvrši jednu naredbu (ili grupu naredbi) ako je neki kriterijum tačan, i drugi skup naredbi ako kriterijum nije tačan. Izraz selekcije daje mogućnost izbora između putanja u zavisnosti od LOGIČKOG USLOVA. Uslovi koji se procenjuju često sadrže RELACIJE i LOGIČKE NAREDBE ili FUNKCIJE.

Ponavljanje (petja)

Struktura ponavljanja, ili petlja, uslovjava da se grupa naredbi izvršava:

- nijedanput
- jadanput
- više puta.

Broj ponavljanja petlje zavisi ili od brojača ili procene logičkog uslova.

RELACIJSKI ILI LOGIČKI OPERATORI

MATLAB podržava šest relacijskih operatora za poređenje dve matrice jednakih veličina.

Relacijski operatori

Relacijski operator	Značenje
<	manje od
<=	manje ili jednako
>	veće od
>=	veće ili jednako
= =	jednako
~=	nije jednako

Poređenja su ili tačna ili netačna, i u većini programskih jezika (uključujući i MATLAB) koriste se sledeće vrednosti:

- za tačno vrednost 1
- za netačno vrednost 0.

Ako definišemo dva skalara

`>>x = 10;`

`>>y = 5;`

a ako napišemo

`>>x < y`

rezultat je netačan i MATLAB vraća vrednost 0.

`ans =
0`

Na osnovu ovakvih odgovora MATLAB odlučuje kod odgovora u:

- strukturama selekcije i
- strukturama ponavljanja.

Promenljive u MATLAB-u obično predstavljaju cele matrice. Ukoliko ponovo definišemo x i y, možemo videti da MATLAB obrađuje poređenja između matrica:

```
>>x = 1:5;  
>>y = x - 5;  
>>x < y
```

x je veće od y za svaki element u izrazu, tako da je svako poređenje netačno i odgovor je niz

vraća

```
ans =  
0 0 0 0
```

MATLAB poredi odgovarajuće elemente i formira matricu sa odgovorima koji su nule u jedinice.

Ako bismo imali sledeće vektore:

```
>>x = [1, 2, 3, 4, 5];  
>>y = [-2, 0, 2, 4, 6];  
>>x<y
```

Samo za poslednji (peti)
element poređenje je tačno

```
ans =  
0 0 0 0 1
```

Da bi MATLAB odlučio da li je poređenje tačno za celu matricu, ono mora da bude tačno za svaki pojedinačni element u matrici. To jest svi rezultati moraju biti jedinice.

LOGIČKI OPERATORI

Postoje sleđći logički operatori:

and (logičko I)

or (logičko ILI)

not (logička Ne operacija)

Logički operator	Značenje
&	and (logičko I)
	or (logičko ILI)
~	not (logička Ne operacija)

Primer:

```
>>x = [1,2,3,4,5];  
>>y = [-2,0,2,4,6];  
>>z = [8,8,8,8,8];  
>>z >x & z>y
```

vraća

```
ans =  
1 1 1 1 1
```

Ovde je $z > x$ i $z > y$ za svaki element i kao odgovor vraćaju se 1

Izraz

```
>>x>y | x>z
```

Uslov je tačan za prva tri elementa a netačan za poslednja dva

vraća

```
1 1 1 0 0
```

Negacija (logička NE operacija)

```
>>a = 4;  
>>b = 10;  
>>a < b
```

```
ans  
1
```

```
>>~(a<b)
```

```
ans  
0
```

Relacijski i logički operatori koriste se u strukturama selekcije i petljama (ciklusima) u cilju određivanja koje naredbe treba da se izvše.

STRUKTURE SELEKCIJE

Naredba `find` je jedinstvena za MATLAB i može često da se koristi umesto struktura `if` i `loop`.

Naredba `find` vraća vektor koji je sastavljen od primeraka (uzoraka) nenuultih članova vektora .

Ovi primerci (uzorci) mogu da budu korišćeni u predstojećim naredbama.

Pretpostavimo da imamo niz vrednosti temperature izmerenih u toku procesa proizvodnje. Ukoliko je temperatura manja od 95°F tada uslovi procesa proizvodnje neće biti dobri.

```
>>t = [100, 98, 94, 101, 93];
```

Da bismo utvrdili koji su uzorci loši možemo koristiti funkciju `find`.

```
>>find(t < 95)
```

vraća vektor sačinjen od brojeva članova čiji su primerci pogrešni.

ans =

3 5

MATLAB najpre izračunava $t < 95$ i daje vektor nula i jedinica; to se može videti ako unesemo:

```
>>t < 95
```

vraća vektor koji sadrži rezultate poređenja kada je poređenje bilo tačno (1) i kada je poređenje bilo netačno (0):

```
ans =
```

```
0 0 1 0 1
```

Naredba `find` vrši pregled ovog vektora i daje brojeve elemenata vektora za koje je poređenje bilo tačno. (To su bile 1 u vektoru `ans`).

Možemo da damo ime ovim listama elemenata,
na primer,

>>pogresno = find(t<95);

i sada možemo formirati tabelu rezultata unošenjem
naredbe

>>tabela1= [pogresno',t(pogresno)']

koja vraća

tabela1 =
3 94
5 93

Kada se koristi naredba `find` sa dvodimenzionalnom matricom, vraća se jedan niz elemenata. Treba povo naglasiti da je MATLAB kolonski orjentisani jezik, pa u tom smislu dvodimenzionalne matrice smatra dugačkom listom brojeva. Naredba `find` koristi princip numerisanja članova u kolonama i takođe ide kolonu po kolonu

Na primer posmatramo matricu 5x3.

1	6	11
2	7	12
3	7	13
4	9	14
5	10	15

Naradbu `find` možemo koristiti za vraćanje određnog reda ili kolone za dati član

`>>[vrsta,kolona] = find (izraz)`

Posmatraćemo dvodimenzionalnu matricu i hoćemo da koristimo naredbu find da bismo utvrdili lokaciju svih elemenata većih od 9:

```
>>x = [1,2,3;19,5,1;3,12,1;8,3,1];
>>element = find(x>9)
>>[vrsta,kolona] = find(x>9)
```

vraća

```
x =
     1   2   3      2      2      1
     10  5   1      7      3      2
     3  12   2
     8   3   1
element =
vrsta =
kolona =
```

VEKTORI I MATRICE

>> %prikaz predstavljanja vektora reda

>> v=[1 2 3 4 5]

v =

1 2 3 4 5

>>% vektor kolone

>> s = [1;2;3;4;5]

s =

1

2

3

4

5

>> %kombinacijom vektora kolona i redova dobija se MATRICA

```
matrica = [1 2 3; 4 5 6; 7 8 9]
```

```
matrica =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
>> matrica(4)
```

```
ans =
```

```
2
```

%pristupanje elementu matrice kreće se brojanjem po kolonama (po vertikali)

```
>> praznaMatrica = []  
praznaMatrica =  
[]  
>> praznaMatrica(1) = 7  
praznaMatrica =  
7  
%punjenje prazne matrice
```

matrica =

1 2 3

4 5 6

7 8 9

>> matrica(1,3)

ans =

3

%prvi argument odgovara broju reda a drugi argument se odnosi na broj po
redu koji se trazi(tj koja
kolona)

```
>> matrica(2,[1,2])
```

```
ans =
```

```
4 5
```

```
% iz drugog reda izdvojiti 1 i 2. element
```

matrica =

1 2 3

4 5 6

7 8 9

>> matrica([1,3],[2,3])

ans =

2 3

8 9

%pravi se podmatrica, argumenti u prvi uglastim zagradama označvaju koji redovi
se biraju a drugi na koje elemente iz tih redova

```
>> matrica(end)
```

```
ans =
```

```
9
```

```
% poslednji član iz matrice (član poslednjeg reda)
```

```
>> matrica(end,:)
```

```
ans =
```

```
7 8 9
```

```
%poslednji red matrice
```

```
>> matrica(:,end)
```

```
ans =
```

```
3
```

```
6
```

```
9
```

```
%poslednja kolona matrice
```

```
>> matrica(end-1,:)
```

```
ans =
```

```
4 5 6
```

```
%pretposlednji red matrice
```

```
>> matrica = [matrica;[5 5 5]]
```

```
matrica =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
5 5 5
```

```
%proširenje matrice za novi red na kraju
```

```
>> matrica=[[4 4 4];matrica]
```

```
matrica =
```

```
4 4 4
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
5 5 5
```

```
%proširenje matrice za novi red na pocetku
```

matrica =

1 2 3

4 5 6

>> matrica(1:2, 3)

ans =

3

6

% matrica(X:Y,Z) u svakom redu od X:Y, prikazuje se Z. po redu element

```
matrica =
```

```
1 2 3
```

```
4 5 6
```

```
>> matrica(:,2)=[]
```

```
matrica =
```

```
1 3
```

```
4 6
```

```
%brisanje druge kolone matrice
```

```
>> matrica(1,:)=[]
```

```
www.puskice.org
```

```
matrica =
```

```
4 6
```

```
%brisanje prvog reda
```

matrica =

1 2 3

4 5 6

7 8 9

>> matrica(:,end+1)=[0 0 0]

matrica =

1 2 3 0

4 5 6 0

7 8 9 0

%dodavanje nove poslednje kolone

```
>> matrica(end+1,:)=[-1 -1 -1 -1]
```

```
matrica =
```

```
1 2 3 0
```

```
4 5 6 0
```

```
7 8 9 0
```

```
-1 -1 -1 -1
```

```
%dodavanje reda na kraju
```

SPECIJALNE MATRICE

```
>> k =zeros(3) %matrica nula, dimenazije 3
```

```
k =
```

```
0 0 0
```

```
0 0 0
```

```
0 0 0
```

```
>> q = ones(3) %matrica jedinica, dimenzije 3
```

```
q =
```

```
1 1 1
```

```
1 1 1
```

```
1 1 1
```

```
>> eye(4)
```

```
ans =
```

```
1 0 0 0
```

```
0 1 0 0
```

```
0 0 1 0
```

```
0 0 0 1
```

%matrica 4x4 ciji su elementi na glavnoj dijagonali jednaki 1 a ostali 0

```
rand(3,3)
```

```
ans =
```

0.157613081677548	0.485375648722841	0.421761282626275
0.970592781760616	0.800280468888800	0.915735525189067
0.957166948242946	0.141886338627215	0.792207329559554

```
% rand(x,y) matrica random slučajnih brojeva, dimenzije x sa y, po uniformnoj  
raspodeli;
```

```
randn generiše po normalnoj raspodeli
```

```
>> fix(100*rand(3,3))
```

```
ans =
```

```
95 84 75
```

```
65 93 74
```

```
3 67 39
```

%ovo je primer matrice nenegativnih celih random brojeva od 0 do 100
(množenjem se zarez pomera za dva mesta, a pomocu funkcije fix
zaokružujemo brojeve ka nuli)

```
matrica =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
>> triu(matrica)
```

```
ans =
```

```
1 2 3
```

```
0 5 6
```

```
0 0 9
```

```
%gornja trougaona matrca
```

```
>> tril(matrica)
```

```
ans =
```

```
1 0 0
```

```
4 5 0
```

```
7 8 9
```

```
%donja trougaona matrica
```

```
>> magic(3)
```

```
ans =
```

```
8 1 6
```

```
3 5 7
```

```
4 9 2
```

```
%magic generiše matricu čiji je zbir po svim kolonama i redovima  
jednak (15 u ovom slučaju)
```

```
>> matrica*matrica
```

www.puskice.org

ans =

30 36 42

66 81 96

102 126 150

% množenje po principu prvi red prvom kolonom

matrica =

1 2 3

4 5 6

7 8 9

>> matrica*2

ans =

2 4 6

8 10 12

14 16 18

%množenje svih elemenata matrice skalarom

```
>> matrica(1,[1,2,3]) = 2*matrica(1,[1,2,3])
```

```
matrica =
```

```
2 4 6
```

```
4 5 6
```

```
7 8 9
```

```
%množenje prvog reda skalarom
```

matrica =

2 4 6

4 5 6

7 8 9

>> matrica(:,2)=3*(matrica(:,2))

matrica =

2 12 6

4 15 6

7 24 9

%množenje druge kolone skalarom

matrica =

2 12 6

4 15 6

7 24 9

>> matrica.*[1 1 1;2 2 2 ;3 3 3]

ans =

2 12 6

8 30 12

21 72 27

%množenje svakog elementa redova obeju matrica odgovarajućim skalarom

matrica =

2 12 6

4 15 6

7 24 9

>> matrica.^2

ans =

4 144 36

16 225 36

49 576 81

%što je ekvivalentno da smo napisali matrica.*matrica

```
>> matrica.^[2 2 2; 3 3 3; 4 4 4]
```

```
ans =
```

```
4 144 36
```

```
64 3375 216
```

```
2401 331776 6561
```

%dizanje na stepen u uglastim zagradama, po svakom redu matrice

```
matrica =  
1 2 3  
4 5 6  
7 8 9  
>> sum(matrica)
```

```
ans =  
12 15 18
```

%suma po kolonama matrice(po defaultu)

```
>> sum(matrica,2)
```

```
ans =  
6  
15  
24
```

%za sumu po svim redovima

```
>>sum(sum(matrica)) suma cele matrice
```

```
>> l = matrica'
```

```
l =
```

```
1 4 7
```

```
2 5 8
```

```
3 6 9
```

%transponovana matrica (kolone i redovi zamene mesta)

```
>>diag(l)
```

```
ans =
```

```
1
```

```
5
```

```
9
```

% diag() izdvaja vektor glavne dijagonale matrice

I =

1 2 3

4 5 6

7 8 9

>> flipr(I)

ans =

3 2 1

6 5 4

9 8 7

% lik matrice u ogledalu

```
matrica =
```

```
1 4 7 4
```

```
2 5 8 4
```

```
3 6 9 4
```

```
5 5 5 4
```

```
>> trace(matrica)
```

```
ans =
```

```
19
```

```
%suma glavne dijagonale kvadratne matrice
```

```
matrica =  
1 4 7 4  
3 6 9 4  
5 5 5 4  
>> max(matrica)  
ans =  
5 6 9 4  
%najveci element iz svake kolone  
>>min(matrica)  
%analogno daje najmanje elemente po kolonama  
>> min(matrica)  
ans =  
1 4 5 4
```

>>`min(matrica,[],2)` i `max(matrica,[],2)`

daju najmanje i najveće elemente po redovima matrice

% `max(max(matrica))` i `min(min(matrica))`

najveći i najmanji elementi matrice

```
matrica =
```

```
1 4 4
```

```
3 6 4
```

```
5 5 4
```

```
>> length(matrica)
```

```
ans =
```

```
3
```

```
%dužina(dimenzija) matrice
```

```
>> [a,b]=size(matrica) %vraća broj kolona i redova matrice
```

```
>> any(matrica)
```

```
ans =
```

```
1 1 1
```

%barem jedan po koloni je različit od nule, kada vrati 1 to znaci da postoji, kada je 0 onda su svi u koloni

```
0
```

```
>> all(matrica)
```

```
ans =
```

```
1 1 1
```

%ispituje da li su svi po kolonama razliciti od nule

% **all(all(matrica))** i **any(any(matrica))** ispituje se da li su svi u matrici različiti od nule odnosno da li u celoj postoji makar jedan koji je različit od nule

```
matrica =  
1 2 3  
4 5 6  
7 8 9  
>> mean(matrica)  
ans =  
4 5 6  
%prosečna vrednost svake kolone  
>> mean(mean(matrica))  
ans =  
5  
%prosecna vrednost cele matrice  
>> mean(matrica')  
ans =  
2 5 8
```

% ovo moze biti efikasan nacin da pronađemo srednju vrednost po redovima matrice tako što upotrebimo funkciju mean a kao parametar prosledimo transponovanu matricu čije su kolone sada redovi matrice a mozemo i ovako:

```
>> mean(matrica,2)
```

```
ans =
```

```
2
```

```
5
```

```
8
```

```
matrica =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
>> prod(matrica)
```

```
ans =
```

```
28 80 162
```

```
%množenje elemenata međusobno po kolonama
```

```
>> prod(prod(matrica))
```

```
ans =
```

```
362880
```

```
%proizvod cele matrice
```

```
>> prod(matrica,2)
```

```
ans =
```

```
6
```

```
120
```

```
504
```

```
%proizvod po redovima
```

```
y =  
1 20 0  
34 12 90  
34 33 11  
>> sort(y)  
ans =  
1 12 0  
34 20 11  
34 33 90  
%sortiranje elemenata u kolonama matrice  
>> sort(y,'descend')  
ans =  
34 33 90  
34 20 11  
1 12 0  
%sortiranje elemenata kolona matrice od većeg ka najmanjem; suprotno je 'ascend'
```

Zadaci za vježbu

Zadatak 1:

Učitati sljedeću matricu:

$$\begin{matrix} 2 & 1 & 9 \\ 11 & 7 & 0 \\ 4 & 5 & 13 \end{matrix}$$

Pronaći:

1. kub svih elemenata matrice
2. ispisati poslednji član matrice
3. novu matricu proširenu za jedan red sa proizvoljnim vrijednostima

Zadatak 2:

Za sljedeći izraz izračunati x, ako je $x=3/4$.

$$f = \frac{x^4 + 3x^3 + 9x^2 - 18}{x^3 + 10x^2 + 8x + 5}$$

Zadatak 3:

Generisati matricu sa slučajnim elementima (jednocifreni brojevi) 4×4 . Iz matrice izdvojiti sljedeće elemente: Npr

2	1	9	8
1	7	0	3
4	5	3	5
9	2	0	1

Podmatricu izdvojiti koristeći operaciju brisanja.

Zadatak 4:

Generisati matricu sa slučajnim elementima (dvocifreni brojevi) 3x3. Npr: 22 67 90

12 15 30

45 86 64

Pronaći:

1. Najveći element matrice
2. Najmanji element u drugom redu matrice
3. prosječnu vrijednost elemenata matrice po kolonama

Zadatak 5:

Pronaći ostatak dijeljenja brojeva a i b ako je a=7 b=4.

Zadatak 6:

Učitati matricu koja ima sljedeće elemente:

- prvi red prikazuje parne elemente od 1 do 10
- drugi red prikazuje neparne elemente počev od broja 41
- treći red prikazuje elemnte kod kojeg je svaki naredni element uvećan za 4 počev od broja 12

Zadatak 7:

Ispitati da li je $x > y$ i da li je $y < z$

```
>>x = [2  0  -6  9   3];
```

```
>>y = [11  1  4   12  -2];
```

```
>>z = [8  -6  2   10  -9];
```

Zadatak 8:

Generisati matricu sa slučajnim elementima (jednocifreni brojevi) 4×4 . Iz matrice izdvojiti sljedeće elemente: Npr,

2	1	9	8
1	7	0	3
4	5	3	5
9	2	0	1

Pronaći:

- a) elemente na glavnoj dijagonali
- b) izdvojiti gornju trougaonu matricu
- c) sumu elemenata na sporednoj dijagonali
- d) prosječnu vrijednost cijele matrice

Zadatak 9:

Generisati matricu sa slučajnim elementima (dvocifreni brojevi) 5×5 . Potom sortirati elemente u metrići po opadajućem poretku.

Zadatak 10:

Iz prethodne matrice izdvojiti elemente tj njihove pozicije koji su veći od 31.

Rješenja zadataka

>> %Zadatak 1

>> A=[2 1 9;11 7 0;4 5 13]

A =

2 1 9

11 7 0

4 5 13

>> %kub elemenata matrice

>> kub=A.^3

kub =

8 1 729

1331 343 0

64 125 2197

>> %poslednji clan matrice

>> p_clan=A(end)

p_clan =

13

>> %prosirena matrica za jedan red

>> p_matrica=[A;0 0 0]

p_matrica =

2 1 9

11 7 0

4 5 13

0 0 0

%ZADATAK 2

```
>> x=3/4;  
  
>> a=x^4+3*x^3+9*x^2-18;  
  
>> b=x^3+10*x^2+8*x+5;  
  
>> f=a/b  
  
f =  
  
-0.6661
```

%Zadatak 3

```
>> Z=fix(rand(4,4)*10)
```

```
Z =
```

```
3 9 3 5
```

```
8 2 5 7
```

```
5 7 0 9
```

```
5 7 0 1
```

```
>> Z(3,2)
```

```
ans =
```

```
7
```

```
>> x=Z;  
>> x(:,[1,2])=[]  
x =  
3 5  
5 7  
0 9  
0 1  
>> x([1,4],:)=[]
```

```
x =  
5 7  
0 9
```

Zadatak 4

```
K=rand(3,3)*100
```

```
K =
```

```
60 68 8
```

```
26 74 22
```

```
65 45 91
```

```
>> %najveci element matrice
```

```
>> maks=max(max(K))
```

```
maks =
```

```
91
```

```
%najmanji element u drugom redu matrice
```

```
>> m=min(K,[],2)
```

```
m =
```

```
8
```

```
22
```

```
45
```

```
>> min(min(m))
```

```
ans =
```

```
8
```

```
>> prosjek_matrice=mean(K)
```

```
prosjek_matrice =
```

```
81.3333 24.0000 71.3333
```

>> %Zadatak 5

```
>> a=7;b=4;  
>> ostatak=rem(7,4)  
ostatak =  
    3
```

>> %Zadatak 6

```
>> A=[2:2:10;41:2:49;12:4:28]  
A =  
    2    4    6    8   10  
   41   43   45   47   49  
   12   16   20   24   28
```

%Zadatak 7

```
>> x=[2 0 -6 9 3];  
>> y=[11 1 4 12 -2];  
>> z=[8 -6 2 10 -9];  
>> x>y & y<z  
ans =  
    0    0    0    0    0
```

>> %Zadatak 8

>> X=fix(rand(4,4)*10)

X =

1 8 8 4

2 5 6 0

1 5 3 2

1 1 5 1

>> %elementi na glavnoj dijagonalni
matrice

>> g_d=diag(X)

g_d =

1

5

3

1

>> gornja_trougaona=triu(X)

gornja_trougaona =

1 8 8 4

0 5 6 0

0 0 3 2

0 0 0 1

>> %suma elemenata na
sporednoj dijagonali

>>

sporedna_dijagonala=fliplr(X)

sporedna_dijagonala =

4 8 8 1

0 6 5 2

2 3 5 1

1 5 1 1

>>

sporedna_dijagonala=trace(sporedna_
dijagonala)

sporedna_dijagonala =

16

>> prosjek_matrice=mean(mean(K))
prosjek_matrice =
58.8889

Zadatak 9:

```
>> X=fix(rand(5,5)*100)
```

X =

18	94	36	40	57
23	49	11	9	5
41	48	78	13	23
4	33	38	94	35
90	90	24	95	82

```
>> sortiranje=sort(X,'descend')
```

sortiranje =

90	94	78	95	82
41	90	38	94	57
23	49	36	40	35
18	48	24	13	23
4	33	11	9	5

Zadatak 10

```
>> find(X>31)
```

ans =

4	18
5	19
6	20
7	21
8	22
10	24
12	25
14	
15	
16	

POLINOMI

Polinomi su funkcije koje imaju sljedeći oblik:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

U Matlab-u se polinomi predstavljaju vektorom vrstom čiji su elementi koeficijenti

$$a_n, a_{n-1}, \dots, a_1, a_0$$

Prvi element je koeficijent ispred x sa najvišim stepenom.

Vektor mora da sadrži sve koeficijente, uključujući i one jednake nuli.

Primjer:

$$4x+2 \quad p=[4 \ 2]$$

$$-2x^2+3x-1 \quad p=[-2 \ 3 \ -1]$$

$$5x^3 - 3x \quad p = [5 \ 0 \ -3 \ 0]$$

$$4x^5 + 4x^2 - 3 \quad p = [4 \ 0 \ 0 \ 4 \ 0 \ -3]$$

Vrijednost polinoma

Vrijednost polinoma u tački x može se izračunati pomoću funkcije **polyval**

polyval(p,x)

p je vektor koji sadrži
koeficijente polinoma

x je broj ,promjenljiva kojoj je dodijeljena vrijednost,ili
izraz koji se može izračunati

Primjer:

1.Za polinom

$$f(x) = -2x^5 + 3x^3 - 2x + 1$$

a) Izračunaj $f(2)$

b) Nacrtaj grafik polinoma za $-2 \leq x \leq 6.5$

Rjesenje:

`>>p=[-2 0 3 0 -2 1];`

`>>f=polyval(p,2)`

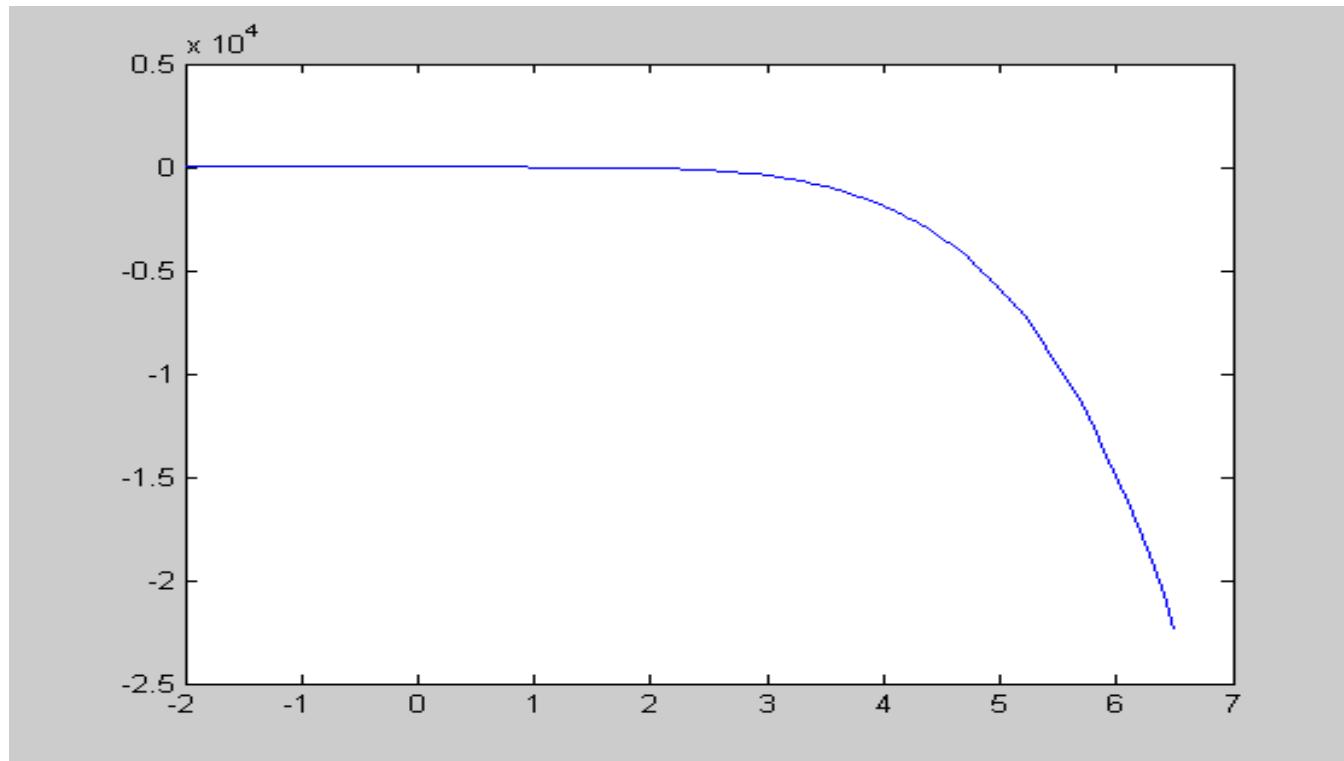
f =

-43

```
>> x=-2:0.1:6.5;
```

```
>> y=polyval(p,x);
```

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



Rješenja polinoma (nule polinoma)

Nule polinoma su vrijednosti argumenta za koje je vrijednost polinoma jednaka nuli.

Primjer:

Rješenja polinoma $f(x)=x^3-2x^2-3x$ jesu vrijednosti za koje je
 $x^3-2x^2-3x=0$, a to su

$$x=0, x=-1 \text{ i } x=3$$

U Matlab-u postoji funkcija koja izračunava rješenje, ili rješenja(tj. nule polinoma)

Funkcija je oblika

```
r = roots(p)
```

r je vektor koji sadrži
rješenje polinoma

p je vektor koji sadrži
koeficijente polinoma

Provjerimo prethodni primjer:

```
>> p=[1 -2 -3 0];  
>> r=roots(p)  
r =  
 0  
 3.0000  
 -1.0000
```

Primjer2.

Odrediti nule polinoma

$$f(x) = -x^5 + 3x^3 - 2x + 3$$

```
>> format short
```

```
>> p=[-1 0 3 0 -2 3];
```

```
>> r=roots(p)
```

r =

$$-1.4019 + 0.4696i$$

$$-1.4019 - 0.4696i$$

$$1.7078$$

$$0.5480 + 0.7095i$$

$$0.5480 - 0.7095i$$

Ako su rješenja polinoma poznata mogu se izračunati koeficijenti polinoma.

Komanda **poly** ima sljedeći oblik.

`p = poly(r)`

`p` je vektor vrsta sa
rješenjima polinoma

`r` je vektor (vrsta ili kolona) sa koeficijentima polinoma

Primjer: Sastaviti polinom čija su rješenja 3 , -4, 2, -1

```
> r=[3 -4 2 -1]
```

```
r =
```

```
3 -4 2 -1
```

```
>> p=poly(r)
```

```
p =
```

```
1 0 -15 10 24
```

%polinom $8x+5$ prvog stepena se moze zapisati kao

y = [8 5];

>> polyval(y,1)

ans =

13

%funkcija polyval vraća vrednost polinoma datog kao prvi argument u tački koja je data kao drugi argument

>> z=[1 2 1];

%što je zapravo polinom drugog stepena: x^2+2x+1 tj kvadrat binoma

>> roots(z)

ans =

-1

-1

%vraća nule polinoma

```
>> z=[3 4 1];
>> s=roots(z)

s =
-1.000000000000000
-0.333333333333333

>> poly(s)

ans =
1.000000000000000    1.333333333333333    0.333333333333333

%ako znamo nule polinoma lako dolazimo i do koeficijenata preko
funkcije poly
```

Operacija sa polinomima

OPERACIJE SA POLINOMIMA

```
>> q = [1 2 3];
```

```
>> s = [5 6 7];
```

```
>> s+q
```

ans =

6 8 10

%klasično sabiranje polinoma gdje se redovi polinoma moraju poklapati

```
>> a = [1 0 1];
```

```
>> b = [2 7];
```

```
>> c = conv(a,b);
```

>> %c je vektor koeficijenata ispred polinoma dobijenog množenjem dva polinoma u konkretnom slučaju množenjem polinoma x^2+1 i $2x+7$ dobija se polinom x^3+7x^2+2x+7 čiji su koeficijenti upravo:

c =

2 7 2 7

Pronalaženje korijena polinoma uz pomoć MatLab-a

- **Zadatak broj 1**
- Naći rješenja jednačine $x^2 - 3x + 2 = 0$.

Rj.

```
>> syms x  
>> roots([1 -3 2])
```

ans =

2

1

>>

Prema tome rješenje jednačine su 2 i 1 .

- **Zadatak broj 2**
- Pronaći korijene polinoma $P(x) = -6x^5 + 15x^4 - 30x^2 + 30x - 13$.

```
>> syms x  
>> roots([-6 15 0 -30 30 -13])  
ans =  
-1.5166  
1.5022 + 0.4650i  
1.5022 - 0.4650i  
0.5061 + 0.5671i  
0.5061 - 0.5671i  
>>  
Prema tome korijeni polinoma su:  $x_1 = -1.5166$  1  $x_2 = 1.5022 + 0.4650i$   
 $x_3 = 1.5022 - 0.4650i$  3  $x_4 = 0.5061 + 0.5671i$  4  $x_5 = 0.5061 - 0.5671i$ 
```

- **Zadatak broj 3**
- Pronaći korijene polinoma $P(k) = k^3 + k^2 - 37k + 35$.

Rj.

```
>> syms x
```

```
>> roots([1 1 -37 35])
```

ans =

-7.0000

5.0000

1.0000

```
>>
```

Prema tome korijeni polinoma su: $k_1 = -7$, $k_2 = 5$, $k_3 = 1$.

Linearne jednačine

Sistem linearih jednačina

- problem rješavanja sistema linearih jednačina jedan je od najčešćih problema linearne algebre
- $x_1+2x_2+3x_3= 366$
- $4x_1+5x_2+6x_3=804$
- $7x_1+8x_2=351$

Sistem linearnih jednačina

Matrični račun: Sistem linearnih jednačina možemo prikazati u matričnom obliku

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

A – matrica sistema

b - slobodni vektor

x – matrica rješenja

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 366 \\ 844 \\ 351 \end{pmatrix}$$

$$\begin{pmatrix} x_1 + 2x_2 + 3x_3 \\ 4x_1 + 5x_2 + 6x_3 \\ 7x_1 + 8x_2 + 0x_3 \end{pmatrix} = \begin{pmatrix} 366 \\ 844 \\ 351 \end{pmatrix}$$

Rješimo sistem pomoću inverzne matrice:

```
>> A=[1 2 3;4 5 6;7 8 0]
```

A =

1 2 3

4 5 6

7 8 0

```
>> b=[366;804;351]
```

b =

366 804 351

```
>> x=inv(A)*b
```

x =

25.0000 22.0000 99.0000

Sistem linearnih jednačina

- Pored korištenja inverzne matrice moguće je rješiti sistem primjenom operacije djeljenja s lijeve strane

```
>> x=A\b
```

x =

25.0000 22.0000 99.0000

- Ovaj drugi pristup određivanju rješenja češće se primjenjuje iz razloga:
 - ima manje operacija što ga čini bržim.
 - u slučaju velikih matrica daje tačnija rješenja.

Zadatak 1

Riješiti sistem linearnih jednačina $Ax=b$ gdje su $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 1 \end{bmatrix}$, $x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ i $b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$.

Zadatak 2

$$x_1 + 2x_2 - x_3 = -2.5$$

$$-x_1 + 3x_2 + 2x_3 = 0.5$$

$$-x_1 - x_2 + x_3 = 1.5$$

Rješenje

```
» A = [1 2 -1; -1 3 2; -1 -1 1];
» b = [-2.5; 0.5; 1.5];
» x = A\b
x =
    2.5000
   -1.0000
    3.0000
» A*x
ans =
   -2.5000
    0.5000
    1.5000
» x = inv(A)*b
x =
    2.5000
   -1.0000
    3.0000
```

MATLAB

Rad sa grafikom

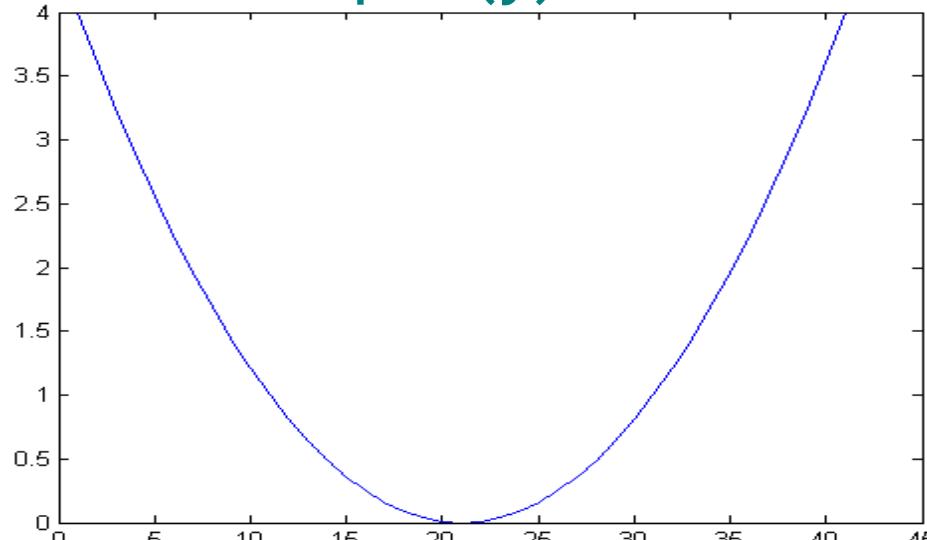
Crtanje grafika funkcije $y = f(x)$

- **plot(y)** – crtanje vektora y u zavisnosti od rednog broja elementa.
- **plot(x,y)** – crtanje funkcije y u zavisnosti od nezavisno promenljive x .
- Ukoliko grafički prozor nije otvoren, **plot** otvara novi grafički prozor.
- Primer:

```
x = -2 : 0.1 : 2;  
y = x .^ 2;
```

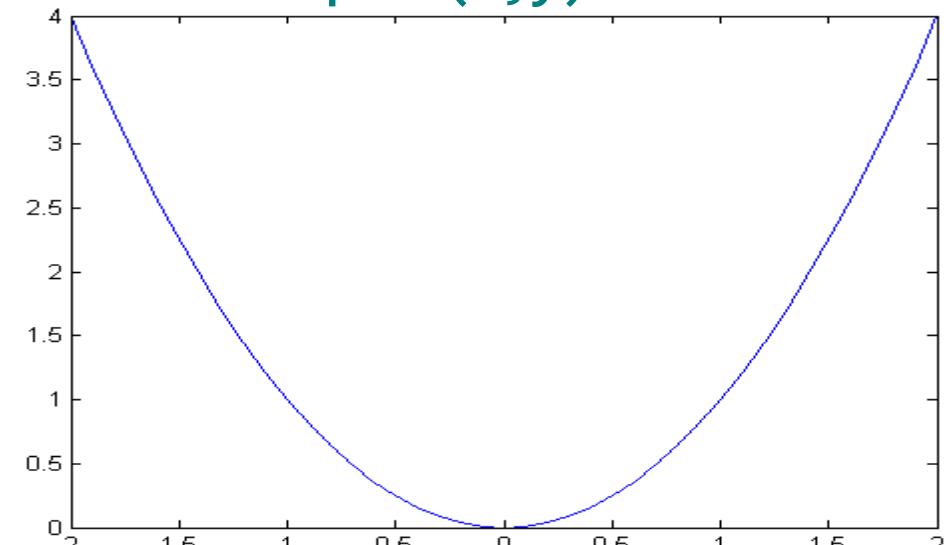
Obratiti pažnju na operator tačka, koji je neophodan za definisanje vektora funkcije.

plot(y)



Redni broj odbirka po x-osi

plot(x,y)



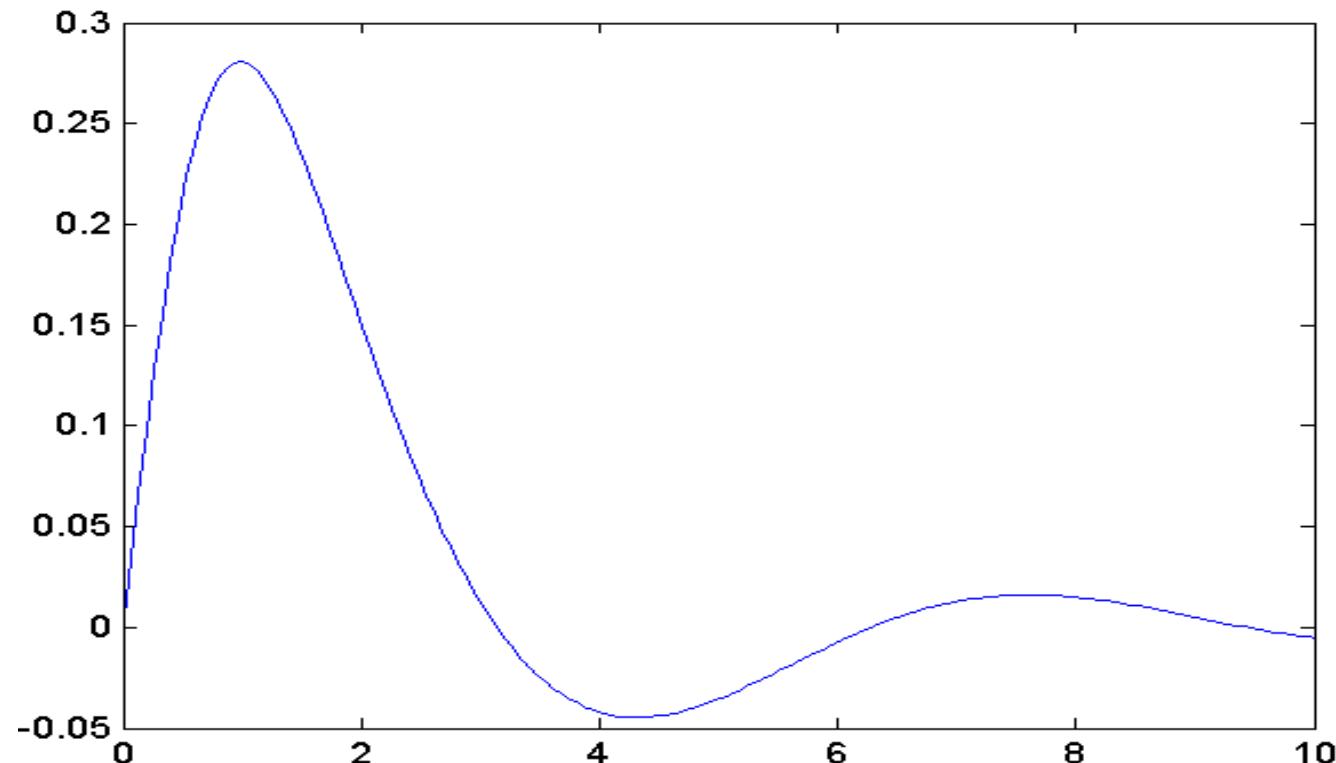
Vrednost promenljive x po x-osi

Funkcija linspace

- Funkcija **linspace** je pogodna za definisanje oblasti x-ose. Sintaksa funkcije je **linspace(x1,x2,N)**, gde **x1** i **x2** predstavljaju početnu i krajnju tačku na x-osi, a **N** broj tačaka.
- Ukoliko se **N** ne navede, podrazumevano se uzima 100 tačaka.
- Primer:

```
x = linspace(0,10,150);  
y = sin(x)./(x.^2+2);  
plot(x,y);
```

Obratiti pažnju na
operator tačka!



Crtanje grafika funkcije. Stil i boja linije.

- **plot(x,y,S)** – crtanje funkcije $y(x)$ pri čemu string S definiše stil i boju linije. String se u MATLAB-u predstavlja tekst unutar apostrofa.
- S sadrži elemente iz jedne od sledeće tri kolone (prva kolona definiše boju, druga karakter kojim se crta linija i treća stil linije):

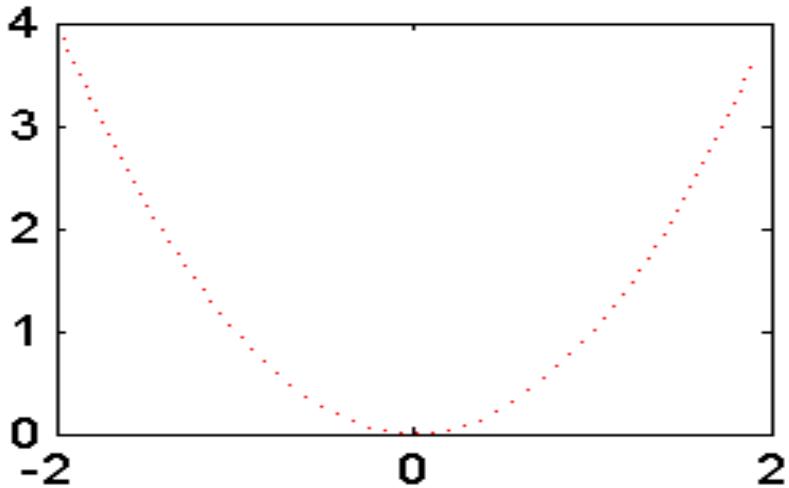
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Na primer, **plot(x,y,'g:')** crta tačkastu zelenu liniju, dok **plot(x,y,'y--o')** crta žutu isprekidanu liniju sa kružićima.

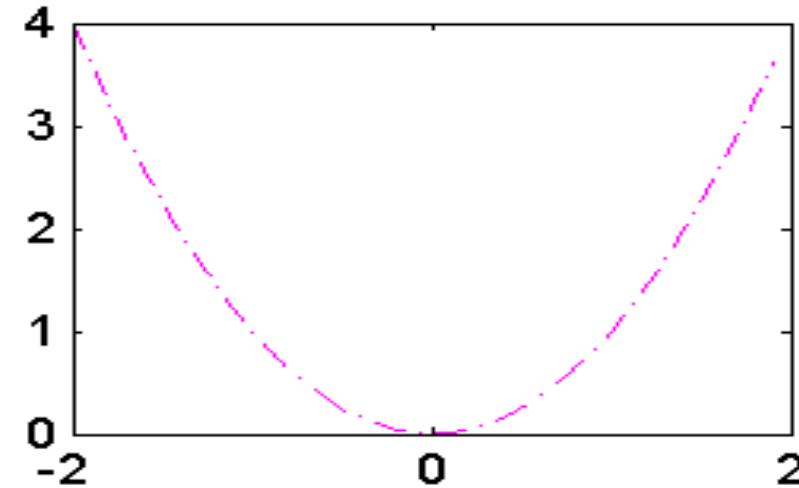
- Parametar **linewidth** u **plot** funkciji definiše debljinu linije u pt-ima.

Primeri

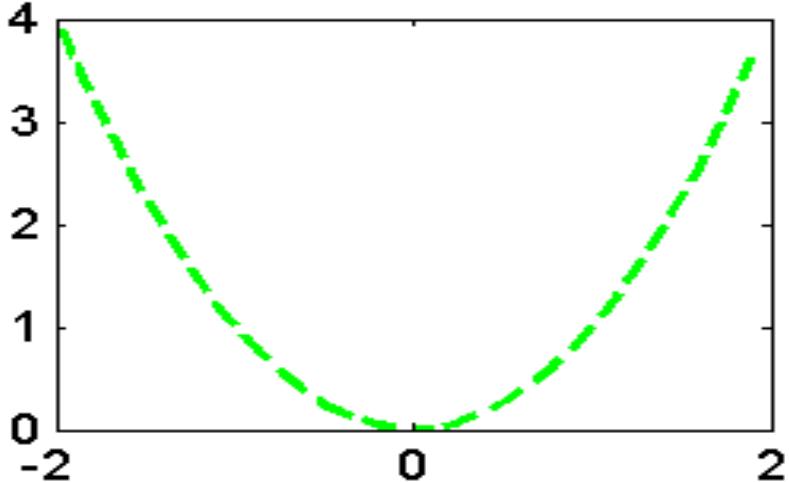
```
plot(x,y,'r:')
```



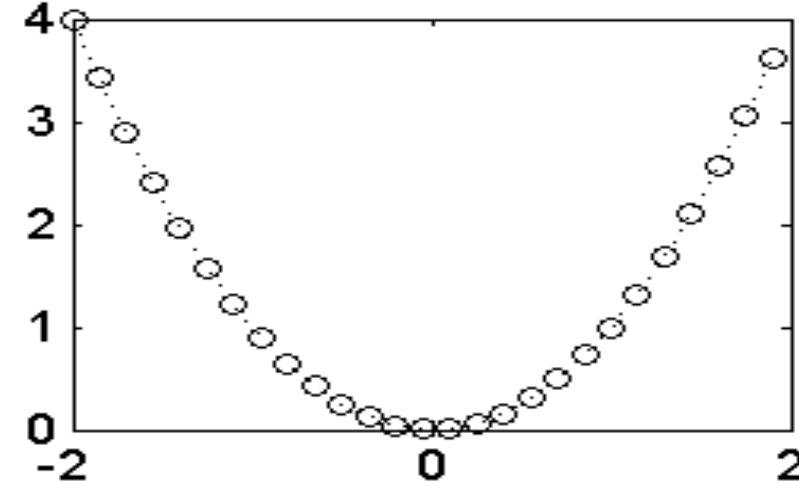
```
plot(x,y,'m-.')
```



```
plot(x,y,'g--','LineWidth',2)
```



```
plot(x,y,'k:o')
```

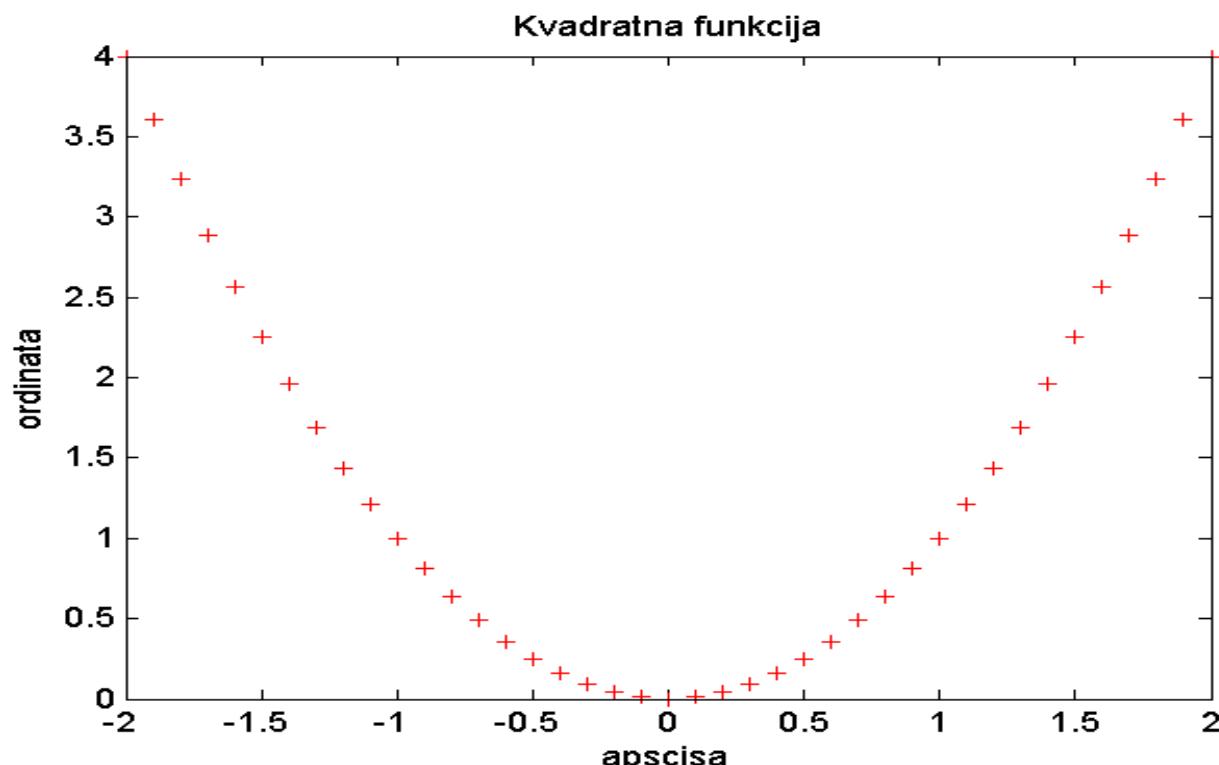


Dodavanje naslova i teksta na osama

- **xlabel('tekst')** – postavlja tekstualnu oznaku x-ose.
- **ylabel('tekst')** – postavlja tekstualnu oznaku y-ose.
- **title('tekst')** – postavlja naslov grafika.

■ Primer:

```
x = linspace(-2,2,41);  
y = x .^ 2;  
plot(x,y, 'r+')  
xlabel('apscisa')  
ylabel('ordinata')  
title('Kvadratna funkcija')
```



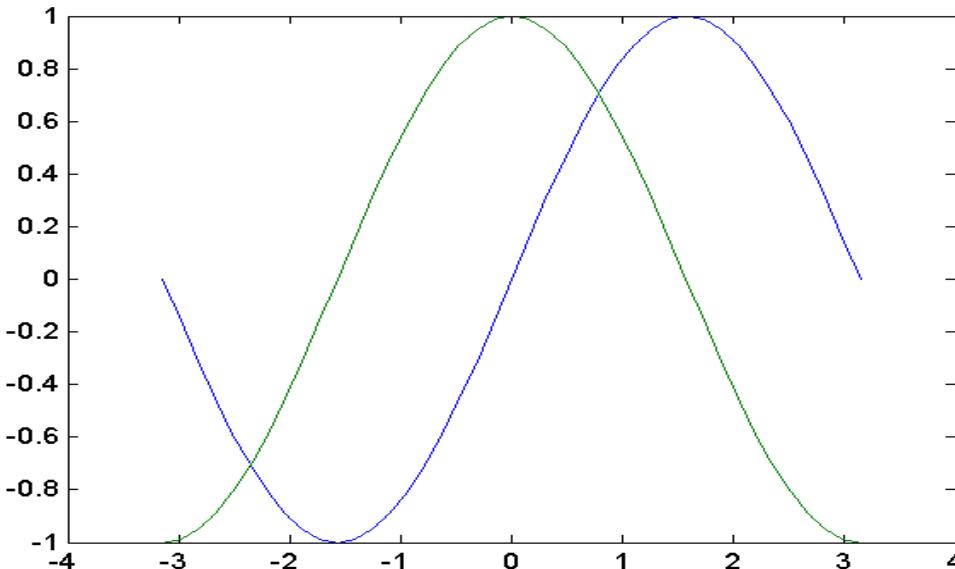
Više funkcija u istom grafičkom prozoru

- **plot(x1,y1,x2,y2,...)** – crtanje više funkcija u istom prozoru.
- **plot(x1,y1,S1,x2,y2,S2,...)** – crtanje više funkcija u istom prozoru. Stringovi S1, S2, ... definišu izgled linija.

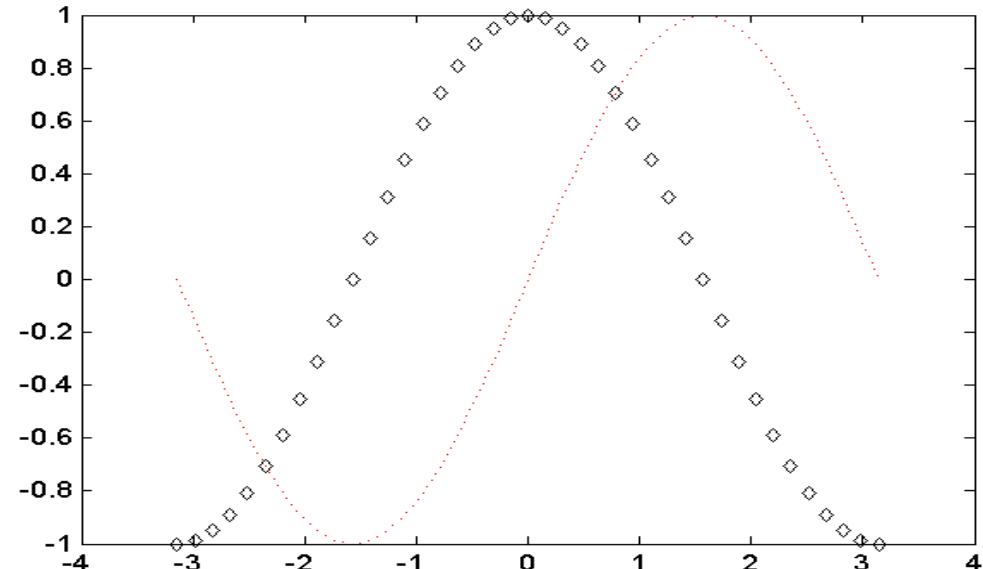
- Primer:

```
x = linspace(-pi,pi,41);
y1 = sin(x);  y2 = cos(x);
```

plot(x,y1,x,y2)



plot(x,y1, 'r:', x,y2, 'kd')



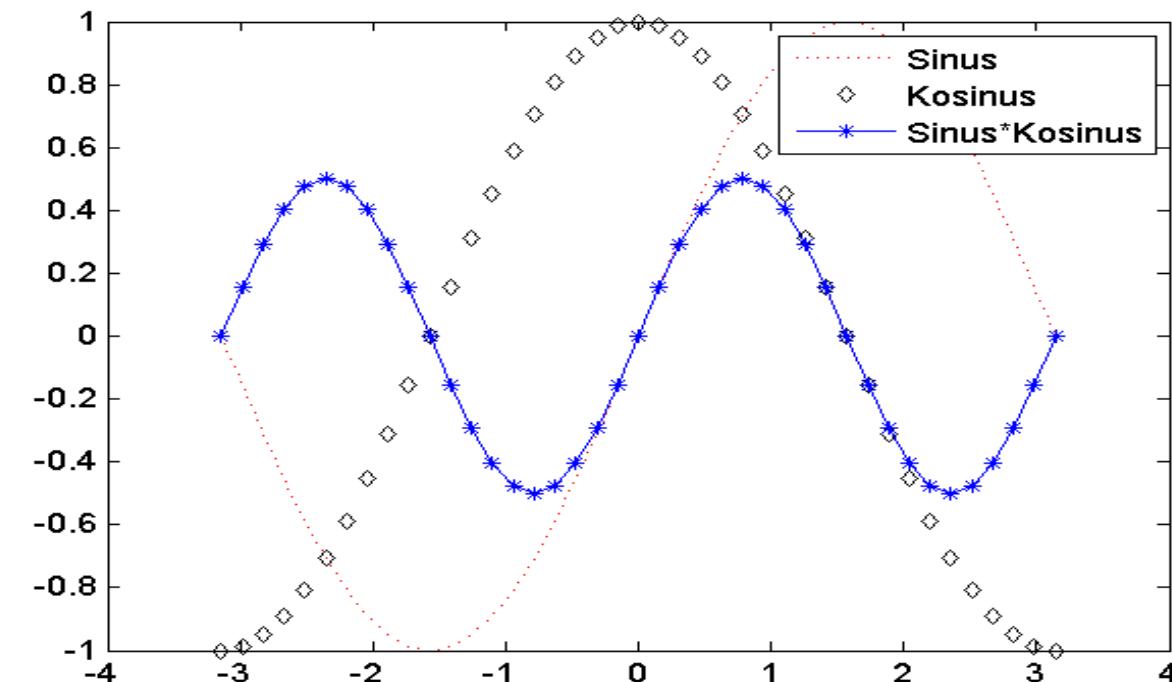
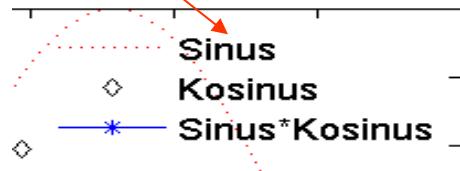
Postavljanje legende na grafik

- **legend('tekst1','tekst2','tekst3',...)** – postavljanje legende na grafik (po redosledu navođenja u plot funkciji).

- Primer:

```
x = linspace(-pi,pi,41);
y1 = sin(x); y2 = cos(x); y3 = y1.*y2;
plot(x,y1,'r:',x,y2,'kd',x,y3,'b-*')
legend('Sinus','Kosinus','Sinus*Kosinus')
```

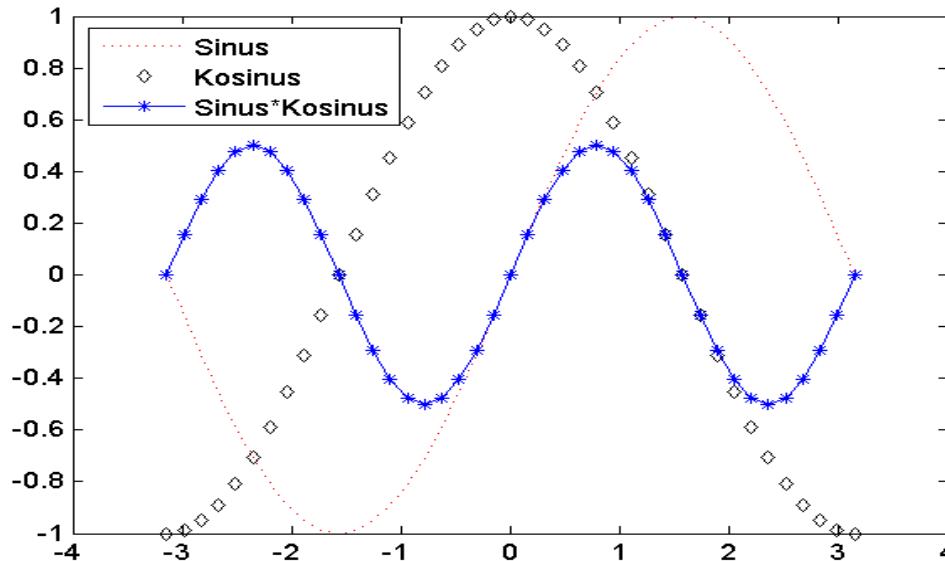
- Naredbom **legend boxoff** se uklanja okvir oko legende.



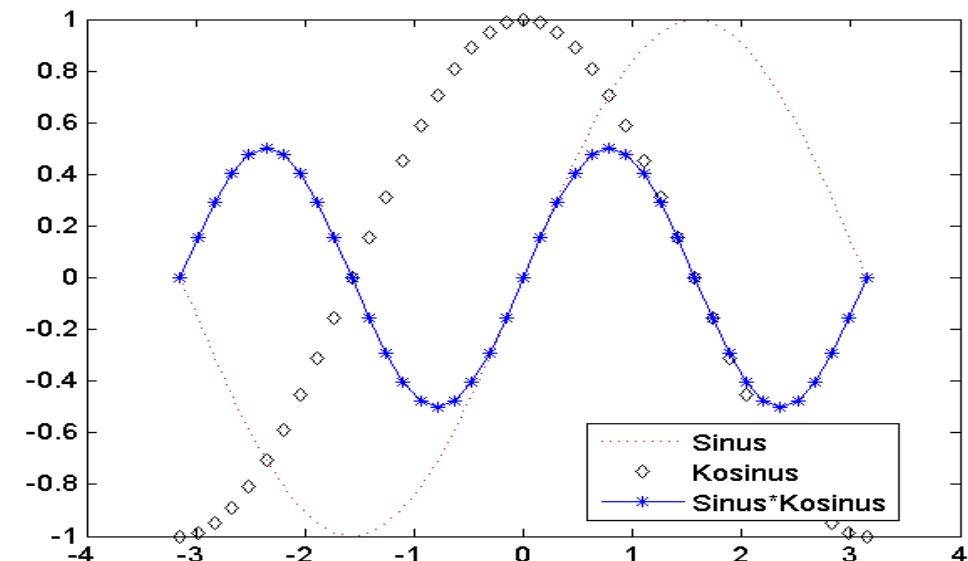
Pozicioniranje legende

- Podrazumevano se legenda pozicionira u gornji desni ugao grafičkog prozora. Pošto na taj način legenda ponekad prekriva deo grafika, poželjno je pomeriti tamo gde ne prekriva grafik nijedne funkcije.
- Legenda ima parametar **Location** koji određuje položaj legende i čija se vrednost zadaje u obliku strana sveta, tj. **North**, **South**, **East**, **West**, **NorthEast** (podrazumevano), **NorthWest**, **SouthEast**, **SouthWest** itd.
- Vrednost parametra **Best** specificira položaj legende sa najmanjim preklapanjem grafika. Ovo, međutim, ne daje svaki put željeni rezultat.

`legend(...,'Location','NorthWest')`



`legend(...,'Location','Best')`

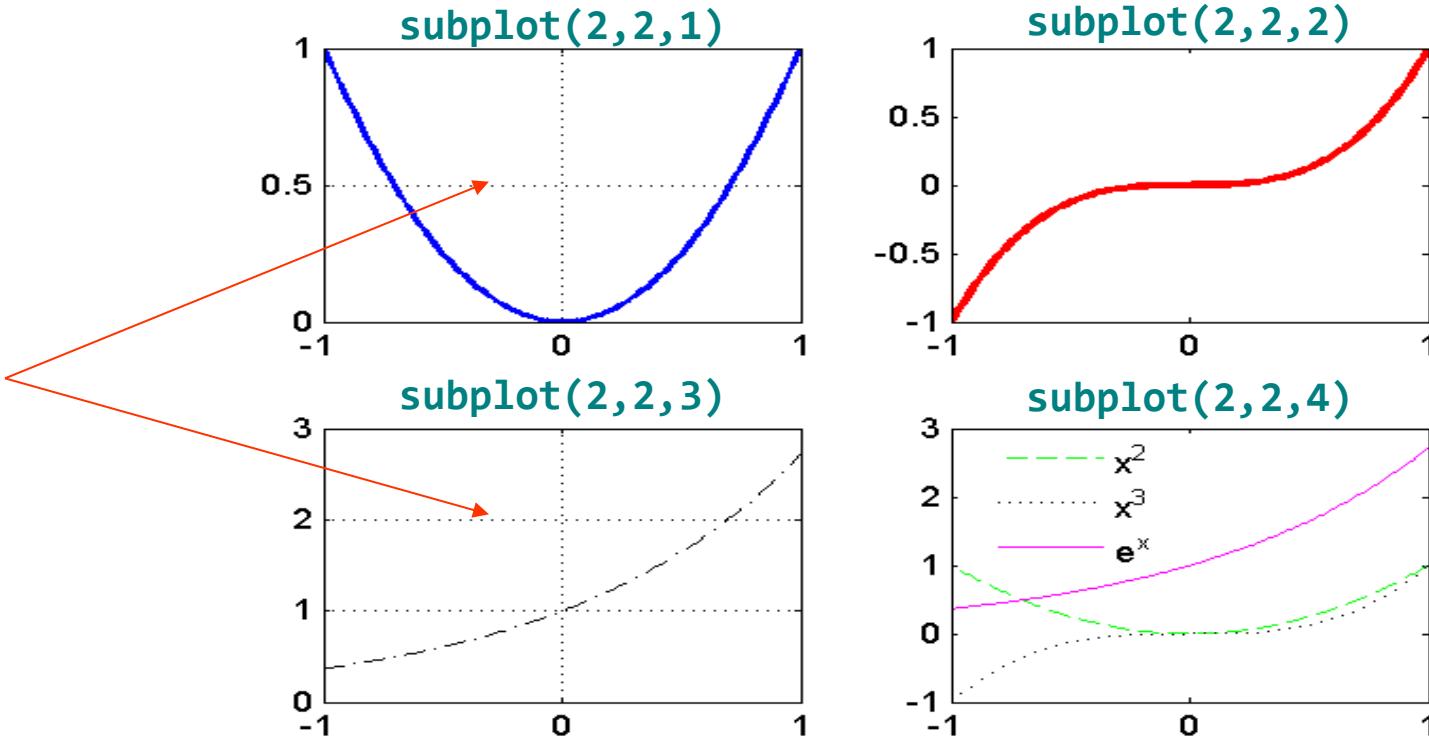


Podela grafičkog prozora

- **subplot(x,y,z)** – podela grafičkog prozora na **x** delova po vertikali, **y** po horizontali i pozicioniranje se u delu **z** za crtanje narednog grafika.

- Primer:

```
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
```

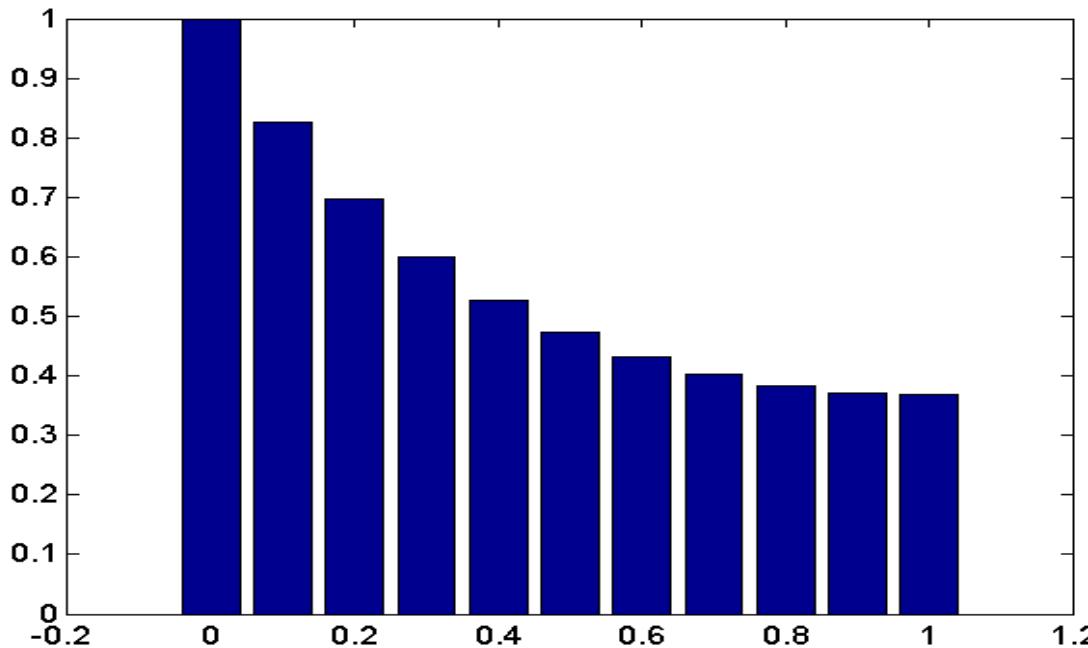


Operator **^** za ispis superscript teksta. Za ispis subscript teksta se koristi **_**

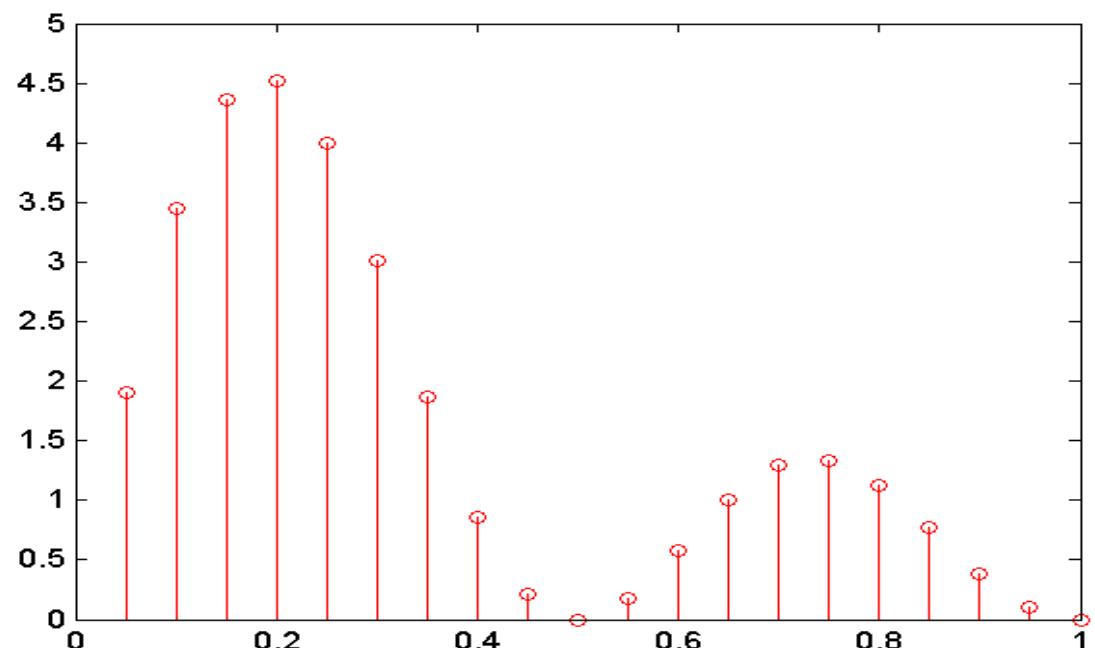
Drugi tipovi grafika

- **bar(x)** – crtanje grafika sa stupcima
- **stem(x)** – crtanje “stem” grafika (ne povezuju se tačke)

```
x = linspace(0,1,11);
y = exp(x.^2-2*x);
bar(x,y)
```



```
x = linspace(0,1,21);
y = sin(2*pi*x).^2./x;
stem(x,y, 'r')
```



Ponavljanje

- Tabela u ovom primeru sadrži podatke nekog preduzeća o prodaji za godine od 1988. do 1994
- Za grafikon datih podataka generisati vektore i potom ih upotrebiti u komandi *plot*

GODINA	1988	1989	1990	1991	1992	1993	1994
PRODAJA (u milionima)	8	12	20	22	18	24	27

Na grafik postaviti naslov PODACI O PREDUZEĆU

Podatke na X osi nazvati GODINA

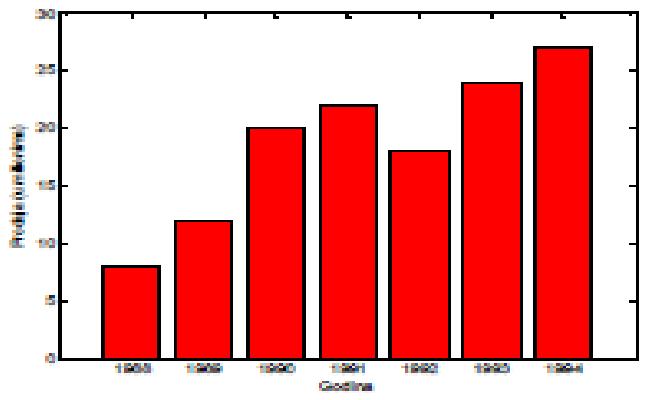
Podatke na Y osi nazvati PRODAJA

Neka grafik bude iscrtan crvenom bojom, debljina linije 3

Trakasti grafikoni

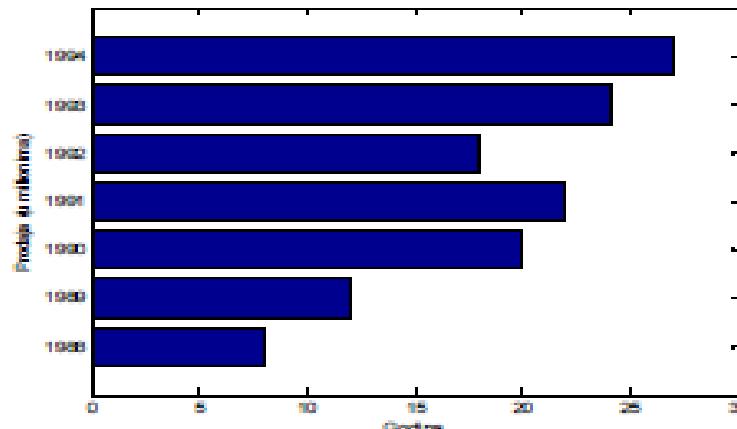
- Trakasti grafikon (vertikalni i horizontalni), stepenasti grafikon i grafikon diskretnih podataka, biće ilustrovani na podacima o prodaji iz ranije navedenog primera

Vertikalni
trakasti
grafikon
`bar(x,y)`



```
god=[1988:1994];  
pro=[8 12 20 22 18 24 27];  
bar(god,pro,'r')  
xlabel('Godina')  
ylabel('Prodaja (u milionima)')
```

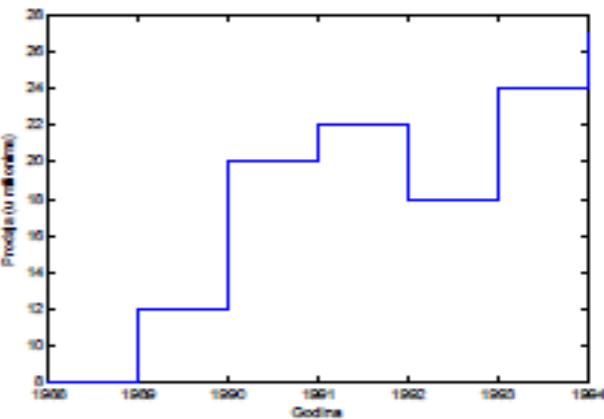
Horizontalni
trakasti
grafikon
`barh(x,y)`



```
god=[1988:1994];  
pro=[8 12 20 22 18 24 27];  
barh(god,pro)  
xlabel('Godina')  
ylabel('Prodaja (u milionima)')
```

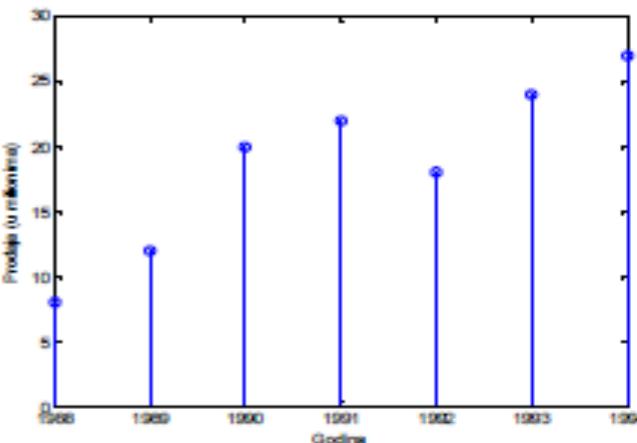
Stepenasti i diskretni grafikoni

Stepenasti
grafikon
stairs(x,y)



```
god=[1988:1994];  
pro=[8 12 20 22 18 24 27];  
stairs(god,pro)  
xlabel('Godina')  
ylabel('Prodaja (u milionima)')
```

Grafikon
diskretnih
podataka
stem(x,y)



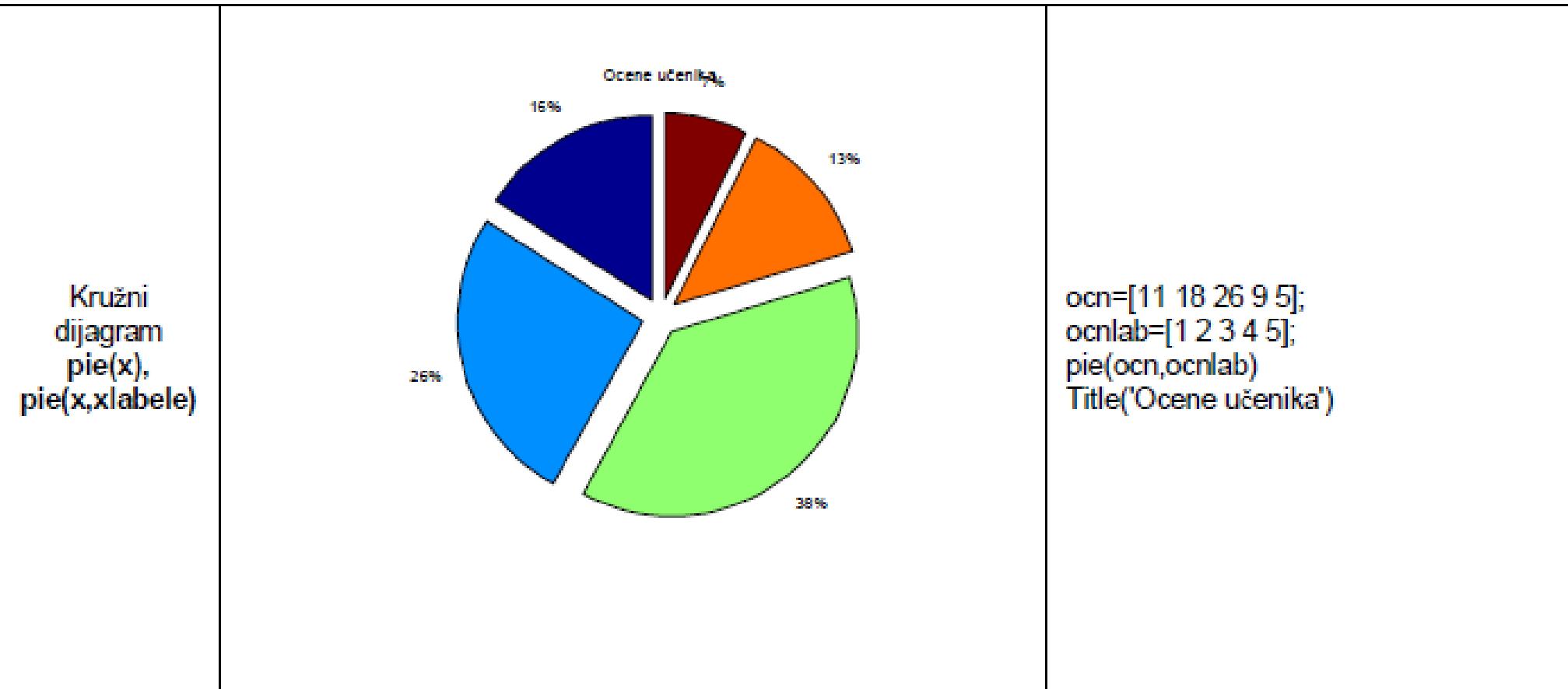
```
god=[1988:1994];  
pro=[8 12 20 22 18 24 27];  
stem(god,pro)  
xlabel('Godina')  
ylabel('Prodaja (u milionima)')
```

Kružni dijagrami

- Kružni dijagrami su podesni za predstavljanje relativnih odnosa različitih ali srodnih veličina
- Na primer, u sledećoj tabeli date su ocene dodeljene određenom razredu
- Pomoću njih su napravljeni kružni dijagram koji slede

Ocena	1	2	3	4	5
Broj učenika	11	18	26	9	5

Kružni 2D dijagram



3D gafikoni

- Trodimenzionalni (3D) grafikoni mogu biti veoma korisni kada treba predstaviti podatke u kojima postoji više od dve promjenljive.
- U MATLAB-u postoji više mogućnosti za prikazivanje trodimenzionalnih grafikona. To mogu biti linijski, žičani, površinski, mrežasti i mnogi drugi grafikoni.
- Grafikoni se lako formatiraju tako da imaju specifičan izgled, a mogu im se dodati i specijalni efekti.
- Pomenućemo neke mogućnosti za trodimenzionalno predstavljanje podataka. Dodatna objašnjenja mogu se naći u prozoru sistema za pomoć, pod odrednicom *Plotting and Data Visualization*

Linjski grafikoni

- Trodimenzionalni linijski grafikon (*lineplot*) jeste linija dobijena povezivanjem tačaka u trodimenzionalnom prostoru
- Osnovni 3-D grafikon crta se pomoću komande *plot3*, veoma slične komandi *plot*, koja ima sledeći oblik:

```
plot3(x, y, z, 'oznake linije', 'ImeSvojstva', vrednost svojstva)
```

x, y i z su vektori koordinata tačaka.

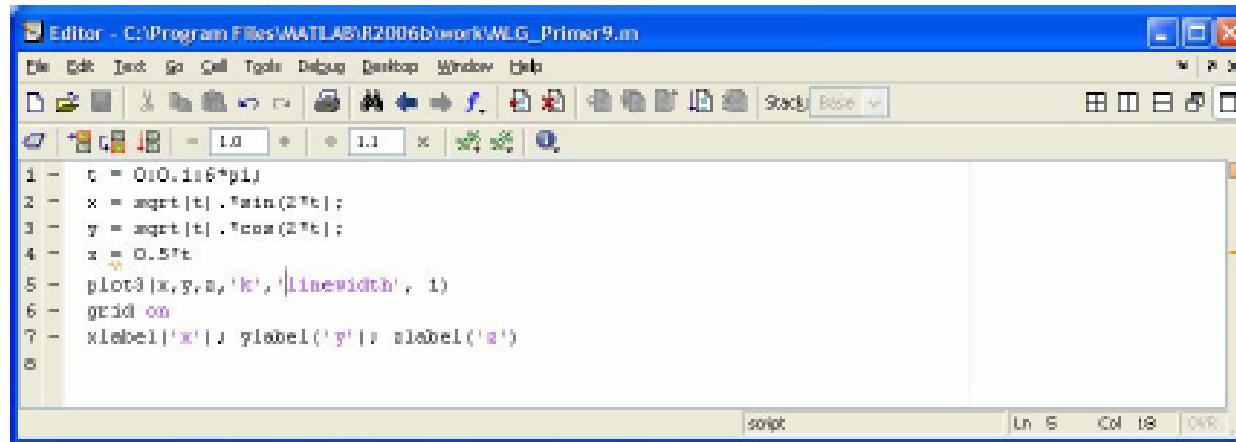
(Opciono) Oznake koje određuju tip i boju linija i markera.

(Opciono) Svojstva koja se mogu upotrebiti za zadavanje parametara kao što su debljina linija, veličina markera, boje ivica i popuna.

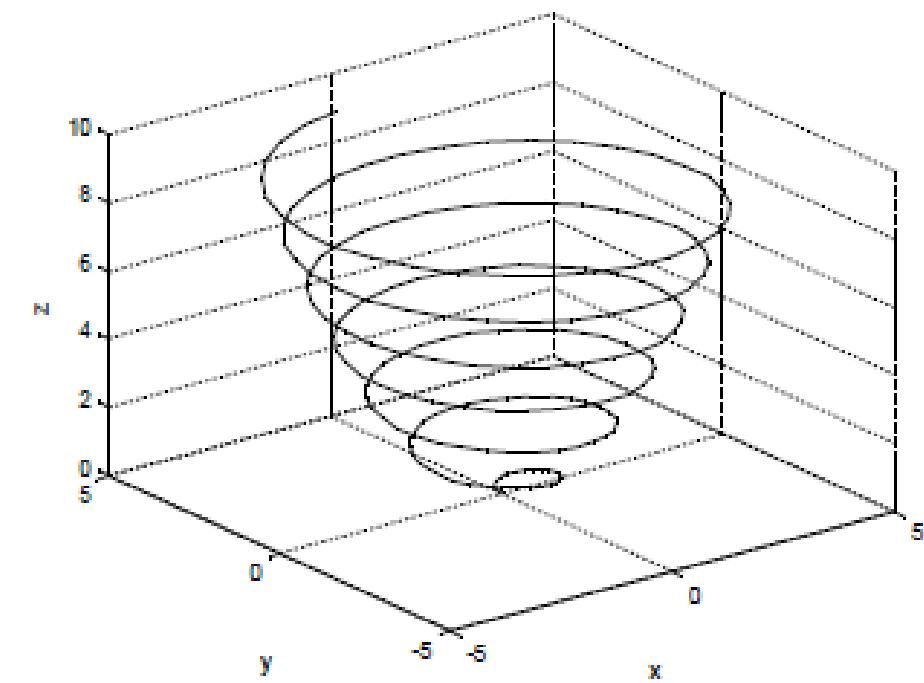
- Sva tri vektora sa koordinatama tačaka koje predstavljaju podatke (*x, y* i *z*) moraju imati jednak broj elemenata
- Oznake linija, svojstva i vrednosti svojstava isti su kao i za dvodimenzionalne grafikone

Primer

- Ako su koordinate x , y i z date u funkciji parametra t sledećim formulama: $x = \sqrt{t} \sin(2t)$, $y = \sqrt{t} \cos(2t)$, $z = 0.5t$ grafikon tačaka za $0 \leq t \leq 6$ može se nacrtati pomoću skript datoteke sa slike levo, a kada se ovaj program izvrši, dobiće se grafikon prikazan na slici desno



```
Editor - C:\Program Files\MATLAB\R2006b\work\WLG_Primer9.m
File Edit Doc Cell Tools Debug Desktop Window Help
New Open Save Run Cell Stop Break Stop All Base Workspace
1 - t = 0:0.1:6*pi;
2 - x = sqrt(t).*sin(2*t);
3 - y = sqrt(t).*cos(2*t);
4 - z = 0.5*t
5 - plot3(x,y,z,'k', 'LineWidth', 1)
6 - grid on
7 - xlabel('x'); ylabel('y'); zlabel('z')
8
```

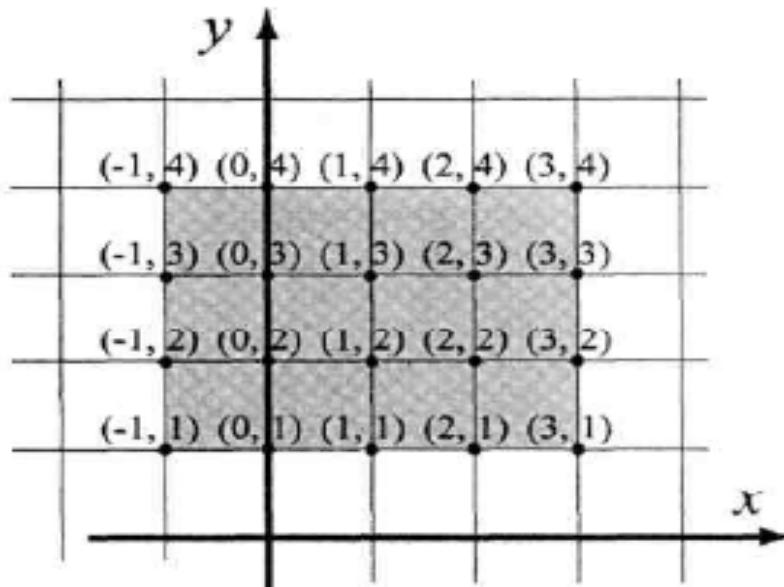


Mrežasti i površinski grafikoni

- Mrežasti grafikoni (*meshplots*) i površinski grafikoni (*surface plots*) su trodimenzionalni grafikoni koji omogućavaju predstavljanje funkcija tipa $z = f(x,y)$, gde su x i y nezavisne promenljive, a z je zavisna promenljiva
- To znači da se u datom domenu vrednost promenljive z može izračunati za svaku kombinaciju x i y
- Mrežasti i površinski grafikoni crtaju se u tri koraka
- Prvi korak je formiranje u ravni x - y rešetke (tj. 3-D koordinatnog sistema) koja pokriva domen (tj. oblast definisanosti) funkcije
- Drugi korak je izračunavanje vrednosti z u svakoj tački rešetke
- Treći korak je crtanje samog grafikona

Formiranje rešetke u ravni x-y:

- Rešetka (*grid*) predstavlja skup tačaka u ravni x-y koji obuhvata oblast definisanosti, tj. domen funkcije
- Gustinu domena (broj tačaka kojim se definiše domen) zadaje korisnik
- Na primer, slika prikazuje rešetku za domen - $-1 \leq x \leq 3$ i $1 \leq y \leq 4$, s korakom 1



Formiranje matrica za rešetku

- U prethodno formiranoj rešetki, razmak izmedu tačaka je jedna jedinica
- Tačke rešetke se mogu definisati pomoću dve matrice, X i Y
- Matrica X sadrži x koordinate svih tačaka rešetke, a matrica Y sadrži y koordinate:

$$X = \begin{bmatrix} -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \\ -1 & 0 & 1 & 2 & 3 \end{bmatrix} \quad i \quad Y = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Funkcija meshgrid

- Matrica X sadrži jednake vrste zato što u svakoj vrsti tačke imaju iste x koordinate, a matrica Y sadrži jednake kolone zato što u svakoj koloni tačke imaju iste y koordinate
- U MATLAB-u postoji ugrađena funkcija *meshgrid* koja omogućava formiranje matrica X i Y i koja ima sledeći oblik:

[X,Y] = meshgrid(x,y)

gde je X matrica x koordinata tačaka rešetke, a x je vektor koji deli domen pomenljive x, dok je Y je matrica y koordinata tačaka rešetke, a y je vektor koji deli domen pomenljive y

- U vektorima x i y, prvi i poslednji element su granice domena
- Gustinu rešetke određuje broj elemenata tih vektora

Primer

- Na primer, matrice X i Y koje odgovaraju rešetki iz prethodnog primer mogu se generisati pomoću komande *meshgrid* sa slike
- Pošto se matrice rešetke formiraju, one se koriste za izračunavanje vrednosti z u svakoj tački rešetke

The screenshot shows the MATLAB Command Window with the title bar "Command Window". The window displays the following code and its output:

```
>> x=-1:3;
>> y=1:4;
>> [X, Y] = meshgrid(x, y)

X =

```

-1	0	1	2	3
-1	0	1	2	3
-1	0	1	2	3
-1	0	1	2	3


```
Y =

```

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4

Izračunavanje vrednosti z u svakoj tački rešetke

- Vrednost z u svakoj tački rešetke izračunava se element po element, isto kao kada se radi s vektorima
- Kada su nezavisne promenljive x i y matrice (moraju biti jednakih dimenzija), izračunata zavisna promenljiva takođe je matrica iste veličine
- Vrednost z na svakoj adresi izračunava se na osnovu odgovarajućih vrednosti x i y
- Nakon definisanja sve tri matrice, one se mogu upotrebiti za crtanje mrežastih ili površinskih grafikona

Primer

- Ako je z zadato formulom: $z = \frac{xy^2}{x^2 + y^2}$

vrednost z u svakoj tački rešetke koju smo prethodno definisali izračunava se kao na slici

The image shows a screenshot of a MATLAB Command Window. The title bar is blue and reads "Command Window". The window contains the following text:

```
>> Z=X.*Y.^2./ (X.^2+Y.^2)
```

Z =

-0.5000	0	0.5000	0.4000	0.3000
-0.8000	0	0.8000	1.0000	0.9231
-0.9000	0	0.9000	1.3846	1.5000
-0.9412	0	0.9412	1.6000	1.9200

Crtanje mrežastih ili površinskih grafikona

- Mrežasti ili površinski grafikon crta se pomoću komandi *mesh* ili *surf*, koje imaju sledeće oblike:

mesh(X,Y,Z)

surf (X,Y,Z)

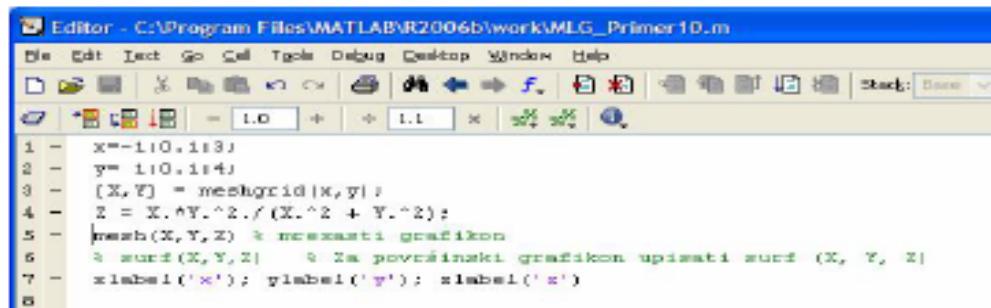
gde su X, Y matrice koordinata rešetke, a Z matrica vrednosti zavisne promenljive z u tačkama rešetke

- Mrežasti grafikon se sastoji od linija koje povezuju tačke
- Na površinskom grafikonu, oblasti između linija mreže ispunjene su bojom

Primer

- Skript datoteka sa slike sadrži program koji formira rešetku, a zatim crta mrežasti (ili površinski) grafikon funkcije

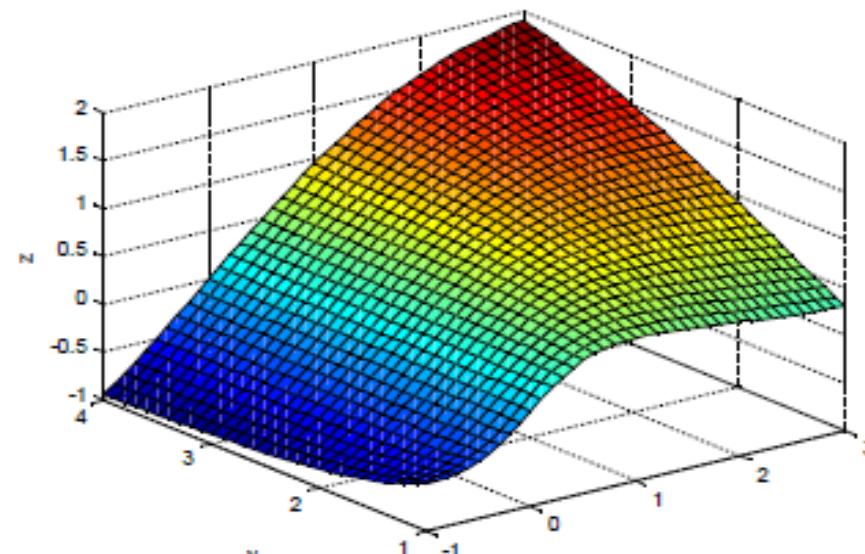
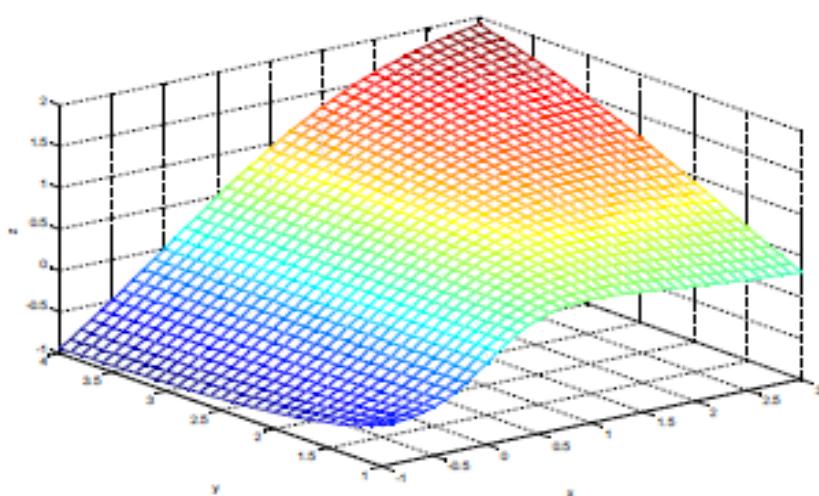
$$z = \frac{xy}{x^2 + y^2} \text{ za } -1 \leq x \leq 3 \text{ i } 1 \leq y \leq 4$$



The screenshot shows the MATLAB Editor window with the script file MEG_Primer10.m. The code defines a grid of points (X, Y), calculates the function Z = XY / (X^2 + Y^2), and then creates a mesh plot of the function over the specified domain.

```
Editor - C:\Program Files\MATLABNR2006b\work\MEG_Primer10.m
File Edit Insert Go Cell Tools Debug Desktop Windows Help
1 - X=-1:0.1:3;
2 - Y= 1:0.1:4;
3 - [X, Y] = meshgrid(X, Y);
4 - Z = X.*Y.^2 ./ (X.^2 + Y.^2);
5 - mesh(X, Y, Z) % mrežasti grafikon
6 - % mesh(X, Y, Z) % Za površinski grafikon upisati sučin (X, Y, Z)
7 - xlabel('x'); ylabel('y'); zlabel('z')
8
```

- U ovom programu vektori x i y imaju relativno mali korak, a manji korak daje gušću rešetku - rezultati ovog programa su prikazani na slici

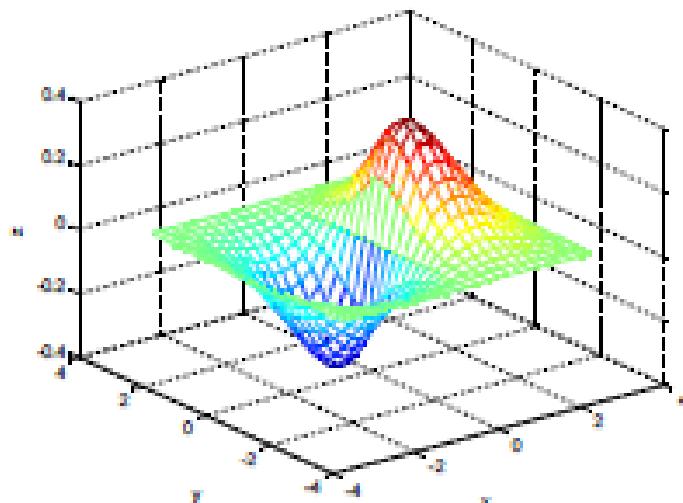


Dodatne komande

- Kada se da komanda *mesh*, prikazivanje linija rešetke se podrazumeva
- Prikazivanje linija rešetke može se isključiti pomoću komande *grid off*
- Pomoću komande *box* može se nacrtati okvir oko dijagrama
- Komande *mesh* i *surf* mogu se upotrebiti i u obliku *mesh (z)* i *surf (z)* u kom slučaju se vrednosti z crtaju kao funkcija njihovih adresa u matrici - broj vrste je na osi x, a broj kolone na osi y
- Postoji više dodatnih komandi za crtanje grafikona, sličnih komandama *mesh* i *surf*, koje omogućavaju crtanje grafikona drugačijih osobina
- Svi naredni primeri prikazuju grafikone vrednosti funkcije

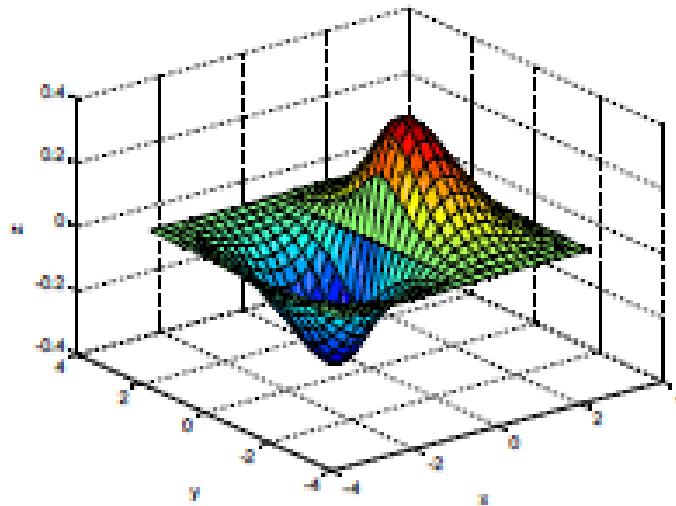
$$z = 1.8^{-1.5\sqrt{x^2+y^2}} \sin(x)\cos(0.5y) \quad \text{za } -3 \leq x \leq 3 \text{ i } -3 \leq y \leq 3$$

Mrežasti
mesh(X,Y,Z)



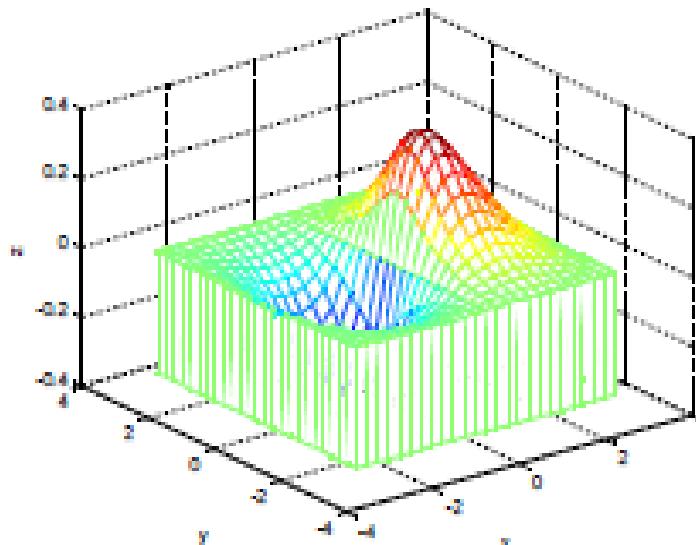
```
x=-3:0.25:3;  
y= -3:0.25:3;  
[X,Y] = meshgrid(x,y);  
Z = 1.8.^(-1.5*sqrt(X.^2+Y.^2))  
. *cos(0.5*Y).*sin(X);  
mesh(X,Y,Z)  
xlabel('x'); ylabel('y'); zlabel('z')
```

Površinski
surf(X,Y,Z)



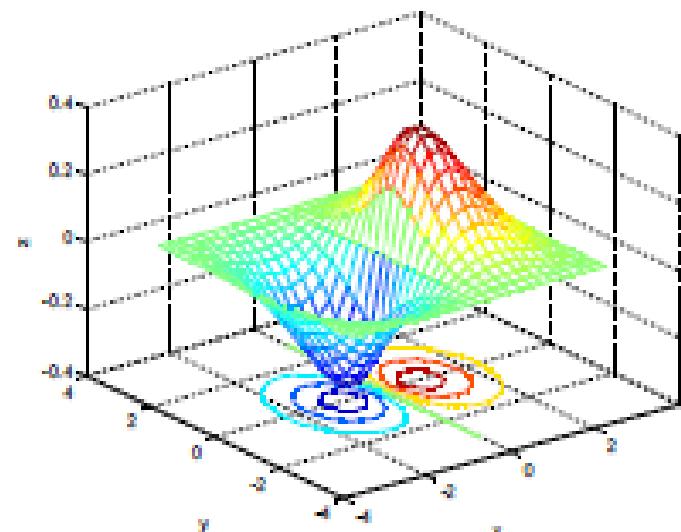
```
x=-3:0.25:3;  
y= -3:0.25:3;  
[X,Y] = meshgrid(x,y);  
Z = 1.8.^(-1.5*sqrt(X.^2+Y.^2))  
. *cos(0.5*Y).*sin(X);  
surf(X,Y,Z)  
xlabel('x'); ylabel('y'); zlabel('z')
```

Mrežasti sa
zavesom
meshz(X,Y,Z)



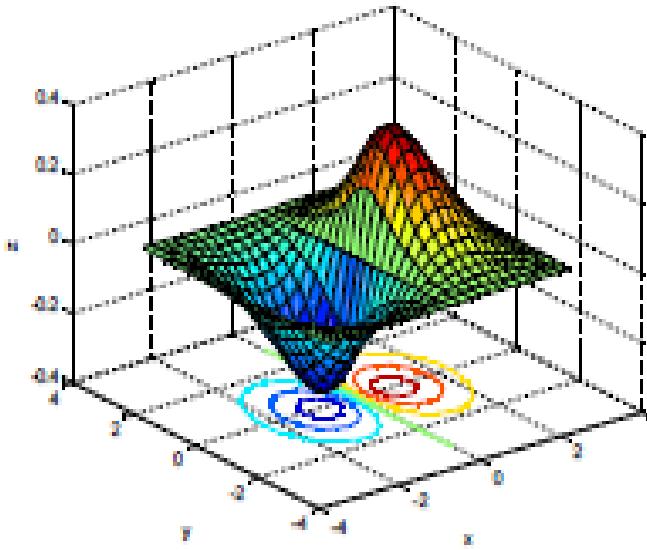
```
x=-3:0.25:3;  
y= -3:0.25:3;  
[X,Y] = meshgrid(x,y);  
Z = 1.8.^(-1.5*sqrt(X.^2+Y.^2))  
. *cos(0.5*Y).*sin(X);  
meshz(X,Y,Z)  
xlabel('x'); ylabel('y'); zlabel('z')
```

Mrežasti sa
konturom
meshc(X,Y,Z)



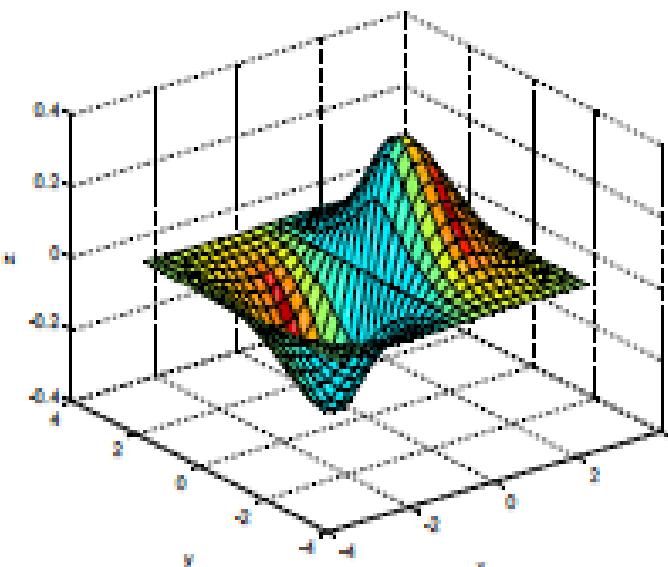
```
x=-3:0.25:3;  
y= -3:0.25:3;  
[X,Y] = meshgrid(x,y);  
Z = 1.8.^(-1.5*sqrt(X.^2+Y.^2))  
. *cos(0.5*Y).*sin(X);  
meshc(X,Y,Z)  
xlabel('x'); ylabel('y'); zlabel('z')
```

Površinski sa
konturom
surf(X,Y,Z)



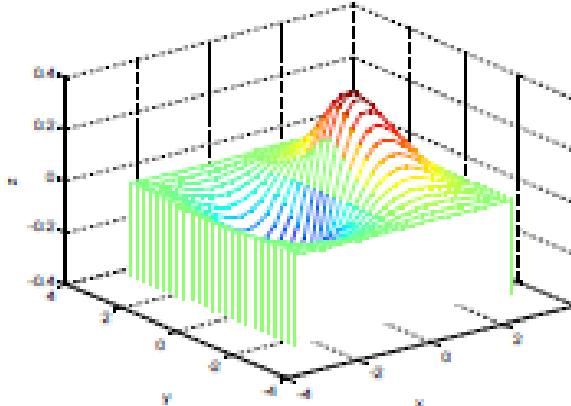
```
x=-3:0.25:3;  
y= -3:0.25:3;  
[X,Y] = meshgrid(x,y);  
Z = 1.8.^(-1.5*sqrt(X.^2+Y.^2))  
. *cos(0.5*Y).*sin(X);  
surf(X,Y,Z)  
xlabel('x'); ylabel('y'); zlabel('z')
```

Površinski sa
konturom
surfl(X,Y,Z)



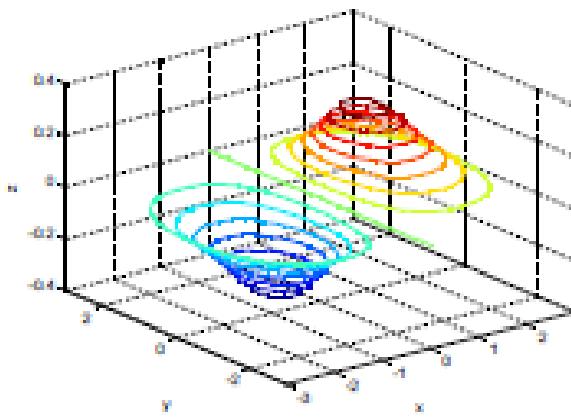
```
x=-3:0.25:3;  
y= -3:0.25:3;  
[X,Y] = meshgrid(x,y);  
Z = 1.8.^(-1.5*sqrt(X.^2+Y.^2))  
. *cos(0.5*Y).*sin(X);  
surfl(X,Y,Z)  
xlabel('x'); ylabel('y'); zlabel('z')
```

Kaskadni
waterfall(X,Y,Z)



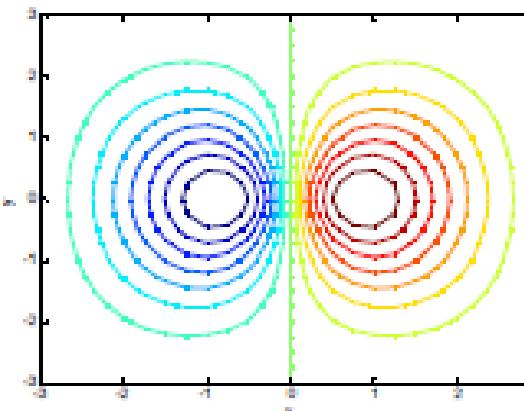
```
x=-3:0.25:3;  
y= -3:0.25:3;  
[X,Y] = meshgrid(x,y);  
Z = 1.8.^(-1.5*sqrt(X.^2+Y.^2))  
. *cos(0.5*Y).*sin(X);  
waterfall(X,Y,Z)  
xlabel('x'); ylabel('y'); zlabel('z')
```

3D kontummi
Contour3(X,Y,Z,n)



```
x=-3:0.25:3;  
y= -3:0.25:3;  
[X,Y] = meshgrid(x,y);  
Z = 1.8.^(-1.5*sqrt(X.^2+Y.^2))  
. *cos(0.5*Y).*sin(X);  
contour3(X,Y,Z,15)  
xlabel('x'); ylabel('y'); zlabel('z')
```

2D kontummi
contour(X,Y,Z,n)



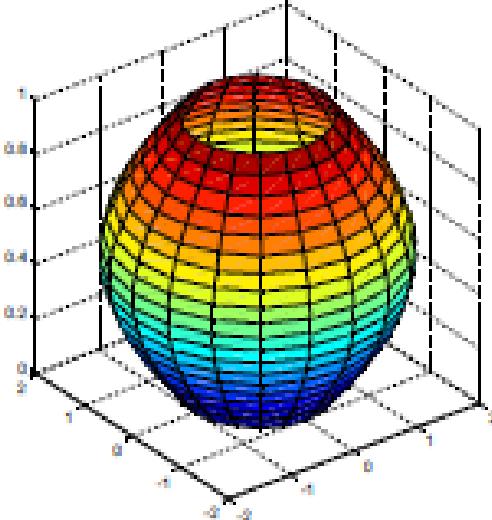
```
x=-3:0.25:3;  
y= -3:0.25:3;  
[X,Y] = meshgrid(x,y);  
Z = 1.8.^(-1.5*sqrt(X.^2+Y.^2))  
. *cos(0.5*Y).*sin(X);  
meshz(X,Y,Z)  
xlabel('x'); ylabel('y'); zlabel('z')
```

Specijalni trodimenzionalni grafikoni

- U MATLAB-u postoje funkcije za izradu raznih vrsta trodimenzionalnih grafikona sa specijalnim efektima
- Lista svih mogućnosti nalazi se u prozoru sistema za pomoć (Help), pod odrednicom *Plotting and Data Visualization*

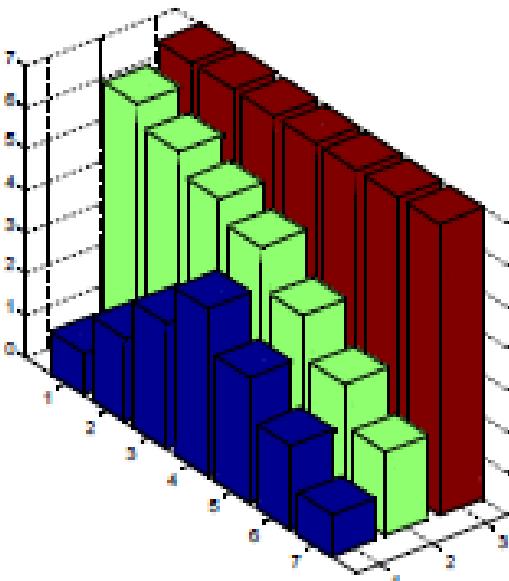
Crtanje jedinične sfere (sa n površina) sphere , sphere(n)		[X,Y,Z] =sphere(20); surf(X,Y,Z)
---	--	-------------------------------------

Crtanje
cilindra
cylinder(r)



```
t=linspace(0,pi,20);  
r=1+sin(t);  
[X,Y,Z]=cylinder(r);  
surf(X,Y,Z)  
axis square
```

3D
stubičasti
Bar3(Y)



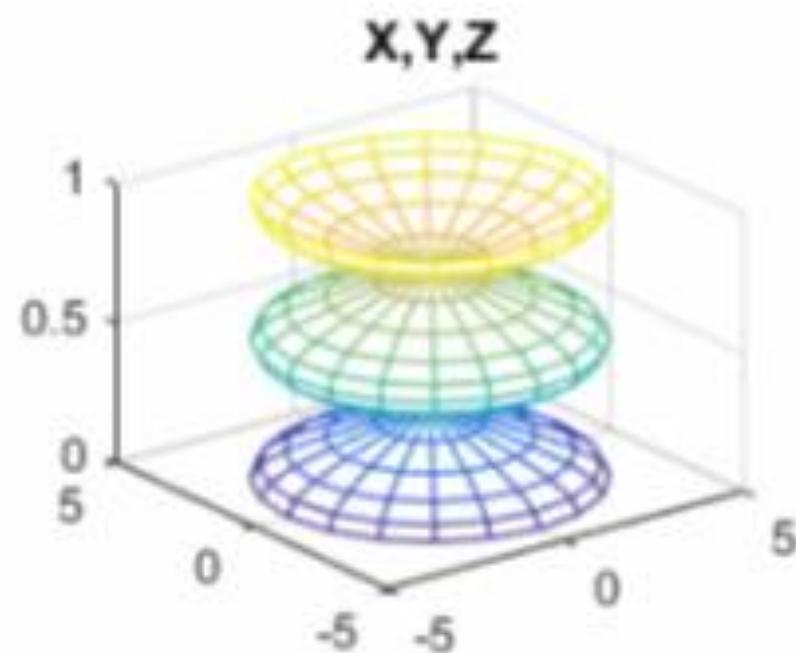
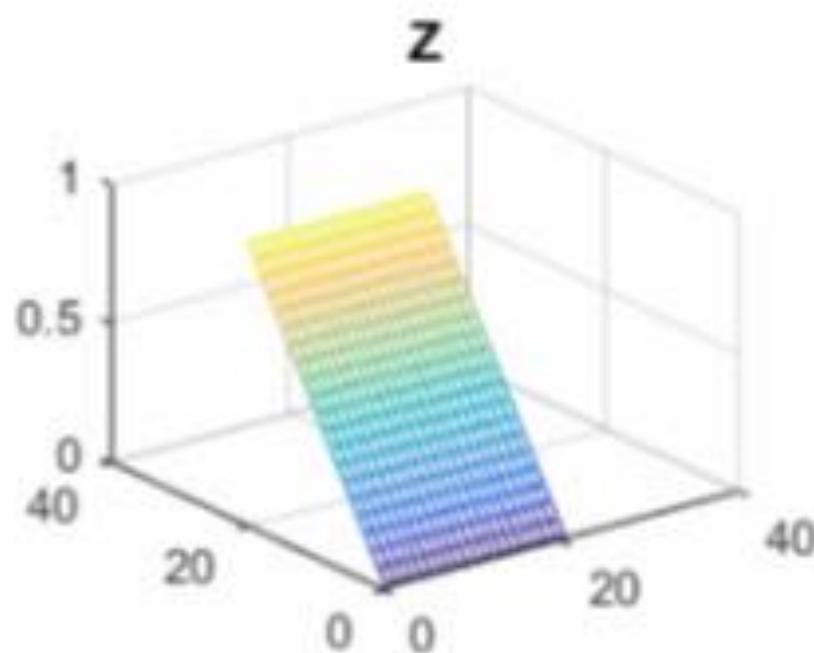
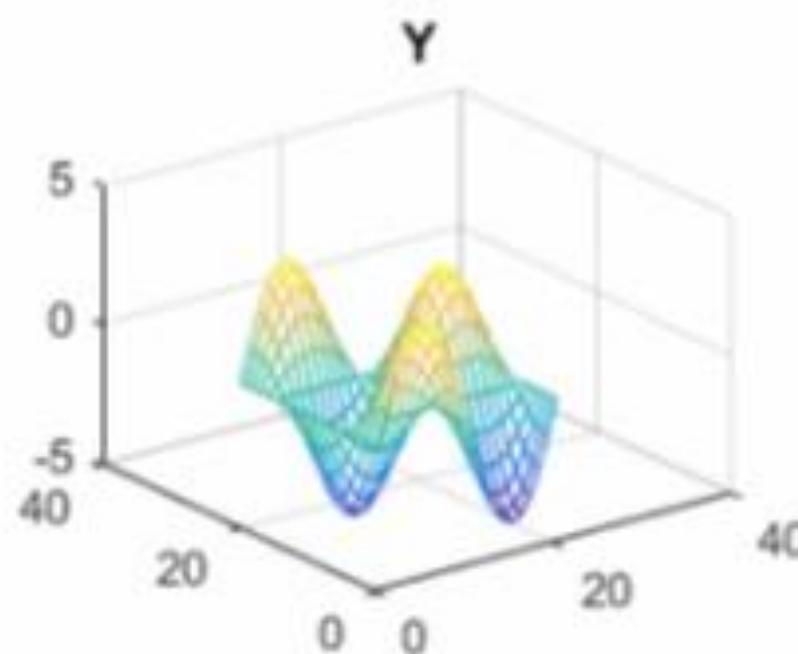
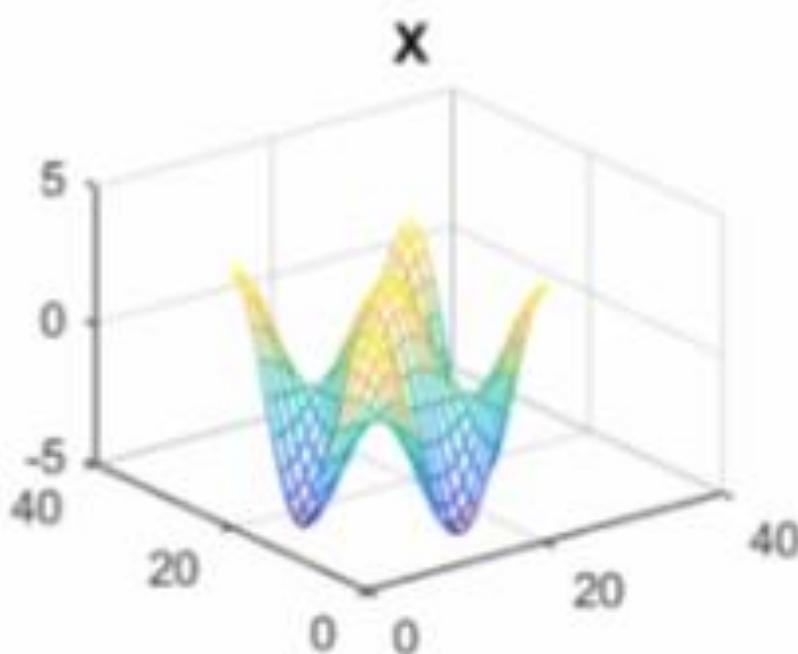
```
Y=[  
1 6.5 7;  
2 6 7;  
3 5.5 7;  
4 5 7;  
3 4 7;  
2 3 7;  
1 2 7];  
bar3(Y)
```

Podcrteži

- Moguće je prikazati više crteža u različitim podregionima istog prozora koristeći funkciju subplot.

Prva dva ulaza za subplot indiciraju broj crteža u svakom redu i koloni.
Treći ulaz specificira koji je crtež aktivan. Na primjer, kreirajmo četiri
crteža u 2-sa-2 rešetki unutar *figure* prozora.

- ```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1); mesh(X); title('X');
subplot(2,2,2); mesh(Y); title('Y');
subplot(2,2,3); mesh(Z); title('Z');
subplot(2,2,4); mesh(X,Y,Z); title('X,Y,Z');
```



# Zadaci za vježbu

# Zadatak 1

- Izračunajte:

$$\frac{33 \cdot 5 \cdot 55 - 7^3}{45 + 5^2}$$

## Zadatak 2

$$(2+15)^3 + \frac{187^{\frac{2}{3}}}{2} + \frac{55^2}{3}$$

## Zadatak 3

- Definišite promenljive  $x$  i  $z$  kao  $x=10,7$   $z=5,8$  i izračunajte:

$$xz^2 - \left( \frac{2z}{3x} \right)^{\frac{3}{5}}$$

## Zadatak 4

- Rjesiti sistem

$$3x - 2y + 5z + 4u = 2$$

$$6x - 4y + 4z + 3u = 3$$

$$9x - 6y + 3z + 2u = 4$$

## Zadatak 5

$$3x - y + 4z + 4u - v = 0, \quad 6$$

$$x - 2y + 2z + 5u + 7v = 0, \quad 9$$

$$x - 3y + 4z + 8u + 9v = 0$$

## Zadatak 6

- Riješiti sljedeće sisteme jednačina:

$$2x - 3y + z = 11$$

$$-3x + 5y - 2z = -19$$

$$x - 2y + 3z = 14$$

$$x + y + z = 0$$

$$2x - 2y - 3z = -7$$

$$-x - y + 2z = 9$$

$$2x + y + 3z = 1$$

$$4x - 2y + 3z = 5$$

$$x + 3y + z = -\frac{13}{6}$$

$$2x + y - z = 1$$

$$5x - 4y + 7z = 2$$

$$7x - 3y + 6z = 3$$

$$x + 2y - 3z = 5$$

$$-3x + y - z = -8$$

$$x - y + z = 0$$

Dat je sistem:

$$1.02x - 0.05y - 0.10z = 0.795$$

$$-0.11x + 1.03y - 0.05z = 0.849$$

$$-0.11x - 0.12y + 1.04z = 1.398$$

Matricnom metodom rjesiti dati sistem:

$$qx - 1.8y + 3.6z = -1.7$$

$$3.1x + 2.3y - 1.2z = 3.6$$

$$1.8x + 2.5y + 4.6z = 2.2$$

- Izračunati nule datog polinoma:

$$288x^5 - 720x^4 + 694x^3 - 321x^2 + 71x - 6 = 0.$$

- Nacrtati funkciju

$$y = x^4 - 3x - 3 \quad \text{u domenu} \quad [-5,5]$$

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

$$y = \frac{x^2 + x - 12}{x - 4}; \quad [-7,7]$$

$$y = \frac{x}{x^2 - 4x + 3}; \quad [-9,9]$$

- Nacrtati funkciju  $f(x) = \frac{x^2 - x + 1}{x^2 + x + 1}$  za  $-10 \leq x \leq 10$

- Nacrtati dva zasebna grafika funkcije  $f(x) = 0.6x^5 - 5x^3 + 9x + 2$ ;
  - jedan za interval  $-4 \leq x \leq 4$ , a drugi za  $-2.7 \leq x \leq 2.7$
- Na istom grafikonu nacrtati funkciju  $y = 3x^3 - 26x + 10$  i njen prvi i drugi izvod, u granicama  $-2 < x < 4$ 
  - Prvi izvod funkcije je:  $y_1 = 9x^2 - 26$ , a drugi izvod funkcije je:  $y_2 = 18x$
  - Sva tri grafika prikazati drugom bojom i drugim stilom linije debljine 2 i definisati legendu grafika.

Zadatak za vježbu:

Tekući položaj čestice u pokretu dat je u funkciji vremena sledećim jednačinama:

$$x = (2 + 4\cos(t))\cos(t)$$

$$y = (2 + 4\cos(t))\sin(t)$$

$$z = t^2$$

Nacrtati grafikon položaja čestice za  $0 \leq t \leq 20$