

# UVOD U MATLAB

MATLAB – interaktivni program za numerička izračunavanja i vizuelizaciju podataka.

Proizvod kompanije Mathworks Inc. ([www.mathworks.com](http://www.mathworks.com))

Matlab="Matrix Laboratory"

U suštini interpretirani jezik baziran na matricama.

Zašto koristiti Matlab?

- Ekstremno pogodno i jednostavno okruženje za rešavanje kompleksnih inženjerskih problema.
- Veliki broj ugrađenih funkcija: od jednostavnih matematičkih funkcija do funkcija za rešavanje simboličkih jednačina.
- Sofisticirane grafičke mogućnosti od jednostavnih grafika do animiranih 3D grafika i prikaza slika. Veliki broj različitih dodataka( u Matlabu se nazivaju *toolbox-ovi*), koji proširuju mogućnosti Matlabu u različitim oblastima primene.
- Jednostavno je razviti kompleksne sisteme: na primer, sistem upravljanja za avion F14.
- Široko se koristi u skoro svim poljima inženjerstva za nastavu, istraživanje i razvoj.

## **Start (Getting Started)**

*Prompt:*

```
>>
```

*Otkucajte:*

```
>>help  
>> save  
>> load  
>> who  
>>quit
```

## **BROJEVI**

*Radi kao kalkulator:*

```
>>10*5  
>>5/10
```

*Dodeljivanje vrednosti promenljivama*

```
>> a=10  
>>b=5  
>>c=a*b
```

*Neke očigledne stvari:*

```
>>a=2.99e8  
>>b=77e9  
>>c=a/b
```

*Osnovne matematičke operacije:*

```
>>a=2  
>>b=300  
>>c=b^a  
>>d=a+b  
>>e=(a*b +c)/d
```

## **MATRICE**

*Broj je zaista samo specijalni slučaj matrice:*

```
>> a=[10]  
>>b=[2]  
>>c=a+b
```

*Matlab misli samo o matricama i pomoću matrica.*

*Vektor vrsta:*

```
>>a=[10, 11]  
>>b=[2, 3]  
>>c=a+b
```

*Ili, vektor kolona:*

```
>> a=[10; 11]  
>>b=[2; 3]  
>>c=a+b
```

*Međutim*

```
>> a=[10; 11]
>> b=[2, 3]
>> c=a+b
```

ERROR ! Matrix Dimensions must agree ! (*GREŠKA ! Dimenzije matrica nisu odgovarajuće*)

*Prazna matrica:*

```
>> em=[]
```

*Množenje matrice i vektora:*

```
>> a=[1, 2; 3, 4]
>> b=[10; 11]
>> c=a*b
```

*I sledeće radi:*

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

*Ali, ovo je očigledna greška:*

```
>> e=b*a
ERROR ! Matrix Dimensions must agree !
```

*Podsetimo se osnovnih pravila:*

*Sabiranje matrica podrazumeva da matrice imaju isti broj vrsta i kolona, tj. da su istih dimenzija.*

*Množenje dve matrice "a\*b" podrazumeva da je broj kolona matrice "a" jednak broju vrsta matrice "b".*

*ltd.*

*Zarezi između elemenata se mogu izostaviti*

```
>> a=[1 2; 3 4]
```

*Kolone se mogu pisati u novom redu*

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

*Može i ovako (ne preporučuje se):*

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

*Tri tačke označavaju nastavak, u ovom slučaju vrste:*

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

*Ispisivanje rezultata na ekranu može se sprečiti kucanjem ";" na kraju reda:*

```
>> a=[10e-5 1.2345e3];
```

Matrica a je ipak tu:

```
>> a
```

*Lako je uvesti novu matricu*

```
>> d=a
```

*I dodeliti nove vrednosti staroj*

```
>> a=d+a
```

*Ispisivanje matrica*

```
>>a
```

```
>>d
```

*Lako je odrediti dimenzije matrica:*

```
>>size(a)
```

*ili*

```
>> [vrste, kolone]=size(a)
```

```
>>dimenzije_a=size(a)
```

*Radi i ovako nešto*

```
>> [mika, laza]=size(a)+size(b)
```

## **VELIKE I MALE MATRICE**

*Element matrice određen je svojim indeksima:*

```
>>a(1,2)
```

*Nova promenljiva*

```
>>abit=a(1,2)
```

*Dodelimo novu vrednost elementu matrice*

```
>>a(1,2)=10
```

*Operacije sa delovima matrice:*

```
>>a=[1 2 3 4; 5 6 7 8; 9 10, 11, 11];
```

```
>>b=a(2:4,2)
```

```
>>c=a(2,1:2)
```

*Izdvajanje cele vrste ili kolone*

```
>>b=a(:,2)
```

```
>>c=a(:,1)
```

*Transponovanje matrica (vrste postaju kolone)*

```
>>a'
```

```
>>b'
```

```
>>c'
```

*Formiranje matrice čiji je deo postojeća matrica:*

```
>>b=[a ; 13 14 15 16]
```

```
>>c=[a; a]
```

```
>>d=[b, b+b, b+b+b]
```

*Formiranje matrice od kolona postojeće matrice:*

```
>> e=[a(:,3), a(:,2), a(:,1)]
```

*Vodite računa o dimenzijama matrice ! :*

```
>> f=[b,a]
```

```
Error ! Matrix Dimensions Must agree !
```

*Formiranje vektora od 0 do 100 sa korakom 1:*

```
>>time=0:1:100
```

```
>>tick=0:0.1:10
```

*ili*

```
>>tick=time/10
```

## **SNIMANJE I UČITAVANJE (SAVING AND LOADING)**

*Podatke možete snimiti u fajl korišćenjem naredbe:*

```
save [ime fajla] [lista promenljivih]
```

*Na primer:*

```
>>save data1 a b;
```

*Podaci su snimljeni u specijalnu binarnu formu (ekstenzija je mat)  
data1.mat*

*Da bi snimili sve promenljive koje su trenutno u memoriji :*

```
>>save data1
```

*Ako otkucate samo "save" svi podaci će biti snimljeni u fajl  
matlab.mat*

*Podatke možete snimiti i u ascii (readable) formi korišćenjem:*

```
>>save data1.dat a b /ascii
```

*U poslednjem slučaju "mat" ekstenzija se ne koristi*

*Podaci se sa diska učitavaju naredbom:*

```
>>load data1
```

*Ako su snimljeni u ascii kodu sa "dat" ekstenzijom, onda treba to i naglasiti*

```
>>load data.dat
```

*Takođe se mogu učitati i eksterno (recimo pomoću nekog drugog programa) generisani podaci:*

```
>>load c_output.dat
```

*Napomena: Ovim su podaci smešteni u matricu "c\_output"*

## **FORMATI PRIKAZA PODATAKA**

*U slučaju da je potrebno promeniti format prikazivanja podataka koristi se naredba "format":*

*Za prikazivanje 15 značajnih cifara*

```
>>format long
```

*Vraćanje formata na default vrednost (5 značajnih cifara):*

```
>> format short
```

*Ostale opcije:*

*Pet značajnih cifara u e formatu:*

```
>> format short e
```

*Petnaest značajnih cifara u e formatu:*

```
>>format long e
```

*Samo znak broja:*

```
>>format +
```

## **SPECIJALNE VREDNOSTI**

pi konstanta 3.14.....

i,j imaginarna jedinica ( $\sqrt{-1}$ )

inf matlab oznaka za beskonačno ( $\infty$ )

NaN Not-a-Number (nije broj)

clock vreme i datum

date datum

ans ako rezultatu izračunavanja nije dodeljena promenljiva, podaci se uvek smeštaju na ovu lokaciju

## **SPECIJALNE MATRICE**

*Postoji veći broj funkcija za generisanje specijalnih matrica.*

*Matrica čiji su svi elementi jedinice:*

```
>>a=ones(4)
```

```
>>a=ones(3,2)
```

*Matrica čiji su svi elementi nule (pogodna za inicijalizaciju)*

```
>>a=zeros(10)
```

```
>>a=zeros(1,10)
```

*Jedinična matrica (elementi na glavnoj dijagonali jednaki jedinici)*

```
>>a=eye(6)
```

```
>>a=eye(8,2)
```

*Jedna interesantna matrica (magični kvadrat):*

```
>>a=magic(4)
```

## OPERACIJE NA POLJIMA

Osnovni podatak u Matlabu predstavlja polje koje ne mora biti dimenzionisano

Neka polja za početak:

```
>>a=[1 2; 3 4]
```

```
>>b=[5;6]
```

```
>>c=[7 8]
```

Množenje (vodite računa o dimenzijama !):

```
>>dm=a*b
```

```
>>em=c*a
```

```
>>fm=c*b
```

```
>>gm=b*c
```

```
>>hm=em*a
```

Množenje skalarom

```
>>ds=a*5
```

```
>>es=pi*b
```

Operacije na elementima polja (uočite razliku):

```
>>hm=em*a
```

```
>>he=em.*a
```

Sabiranje i oduzimanje:

```
>>da=b-c'
```

```
>>fa=em+a
```

```
>>ga=5-a
```

### Operacije na poljima: deljenje

Vektor ili matrica podeljena skalarom:

```
>>dd=a/5
```

```
>>ed=b/2.345
```

Deljenje sa leve strane:

```
>>fd=5\c
```

Deljenje vektora vektorom, element po element:

```
>>dd=a./em
```

```
>>ed=b'./c
```

```
>>ed=a.\em
```

Međutim, operacija deljenja nije uvek jednostavna. Matlab podrazumeva da je cilj operacije deljenja rešavanje sledećih matricnih jednačina:

$Ax=b \Rightarrow x=A^{-1}b$  (deljenje sa leve strane)

$xA=b \Rightarrow x=bA^{-1}$  (deljenje sa desne strane).

Na primer (pod pretpostavkom da postoji inverzna matrica matrice "a"):

```
>> ds=c/a
```

```
>> es=b\1
```

Međutim, Matlab će rešiti i sledeće jednačine, kada matrica nije kvadratna

```
>> fs=1/b
```

```
>> gs=c\1
```

Stoga, pri korišćenju operacije deljenja treba biti vrlo pažljiv!

## **SKRIPT FAJLOVI (SCRIPT FILES) U MATLABU**

Jedan od načina da se napiše program koji se može koristiti u Matlabu je korišćenje tzv. skripta (slobodno rečeno, programski jezik Matlaba) čiji se kod može snimiti na HDD sa odgovarajućom ekstenzijom.

Sa linije menija uđi u editor (File, New, M-File) i unesi sledeće

```
numerator=x.^3-2*x.^2 + x - 6.3;  
denominator=x.^2+0.05005*x-3.14;  
f=numerator./denominator;
```

Snimi prethodno u fajl pod imenom primer1.m (ekstenzija "m" je vrlo bitna). Obavezno voditi računa u koji direktorijum se snima fajl, tj. idi u direktorijum koji je predviđen za vežbe:

*Izvršenje programa*

```
>>x=1  
>>primer1  
>>f
```

*Ili, može se zadati i niz ulaznih vrednosti*

```
>>x=0:0.1:10  
>>primer1  
>>f
```

## **NEKE UOBIČAJENE MATEMATIČKE FUNKCIJE**

Matlab sadrži veliki broj ugrađenih funkcija

Trigonometrijske: sin, cos, tan, acos, asin, atan, sec. na primer:

```
>>sin(4.567)  
>>cos(2*pi*4.3)
```

Opšte funkcije: sqrt, log, log10, exp. Eg:

```
>>sqrt(2)  
>>exp(3)
```

Numeričke funkcije: abs, round, fix, floor, sign, ceil. Eg:

```
>>round(2.645)  
>>floor(2.645)
```

Binarne operacije: atan2, rem, etc: Eg:

```
>>rem(101,3)
```

Pogledati u helpu kompletan spisak (Elementary math functions)



## **FUNKCIJE NA POLJIMA**

Većina Matlab funkcija radi i sa vektorima:

```
>>t=0:0.1:20
>>x=sin(2*pi*t);
>>y=cos(2*pi*t)
>>suma_kvadrata=x.^2 + y.^2
```

Takođe i sa matricama:

```
>>a=[1 2; 3 4];
>>x=sin(a)
>>y=cos(a)
>>tacan_odgovor=x.^2 + y.^2
>>pogresan_odgovor=x^2 + y^2
```

Uočite razliku!!

Treba voditi računa da su izvesna funkcije definisane na različite načine za matrice i skalare. Na primer:

```
>>sqrt_t=sqrt(t)
>>sqrt_a=sqrt(a)
```

## **KOMPLEKSNI BROJEVI**

*Matlab tretira kompleksne brojeve na transparentan način:*

```
>>j
>>a=2+3j
>>b=4+5j
>>c=a/b
```

*Kompleksni brojevi kao elementi vektora*

```
>>t=-2:0.1:2
>>x=sqrt(t)
```

*Kompleksni brojevi kao elementi matrice*

```
>>a=[2 +3j , 4+5j ; 6+7j 8 +9]
>>inv(a)
```

*Napomena: Element koji je kompleksan ostaje takav čak i kada je njegov imaginarni deo jednak nuli.*

*Neke funkcije sa kompleksnim brojevima:*

```
>>abs(b)
>>real(b)
>>imag(b)
>>mag(b)
>>angle(b)
```

## GRAFICI

Matlab se odlikuje odličnim mogućnostima za prikaz i analizu grafičkih podataka. Najpre, osnovno o graficima.

U prvom koraku treba definisati podatke

```
>>t=0:0.1:10  
>>x=sin(2*pi*t)  
>>y=cos(2*pi*t)
```

Osnovni oblik naredbe "plot": na ordinati su indeksi elemenata vektora, a na apscisi sami elementi

```
>>plot(x)
```

Grafik vektora  $x$  u odnosu na vreme  $t$  (vektori moraju biti iste dužine).

```
>>plot(t,x)
```

Više linija na istom grafiku:

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

Grafik vektora  $y$  u odnosu na vektor  $x$

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

Podешavanje opsega osa grafika pomoću naredbe `axis([xmin xmax, ymin,ymax])`

```
>>axis([0 2 -2 2])  
>>ve=[-2 2 0 1]  
>>axis(v)
```

Probajte  $i$

```
>>axis('equal')  
>>axis('normal')
```

Sve opcije je moguće dobiti sa "help axis"

Naziv grafika i osa:

```
>>title('A nice sine-wave')  
>>xlabel('Magnitude')  
>>ylabel('time')  
>>text(20,0.5,'A spot')
```

Mreža pomoćnih linija

```
>>grid
```

Naredba `hold` služi za nastavak crtanja na istom grafiku

```
>>hold on  
>>plot(t,y)  
>>hold off
```

*Menjanje boja linija i označavanje tačaka.*

```
>>plot(t,x,'r',t,y,'g')
```

*Vektor x je u crvenoj, a y u zelenoj boji.*

*Tačke na grafiku se mogu označiti:*

```
>>plot(t,x,'x')
```

*Promena boja i označavanje istovremeno:*

```
>>plot(t,x,'ro',t,y,'gx')
```

Sve oznake:

y žuta	.	point
m magenta	o	circle
c cyan	x	x-mark
r crvena	+	plus
g zelena	-	solid
b plava	*	star
w bela	:	dotted
k crna	-.	dashdot
	--	dashed

*Nacrtajte grafik date funkcije x(t)*

```
>>t=0:0.1:10
```

```
>>x=exp(-t)+exp(-4t)
```

```
>>plot(t,x)
```

*Crtanje grafika sa osama u logaritamskoj skali*

```
>>semilogx(t,x)
```

```
>>semilogy(t,x)
```

```
>>loglog(t,x)
```

*U inženjerskim aplikacijama je uobičajeno da se koriste logaritamski grafici za prikaz reda veličine promene amplitude.*

*Naredbe koje su ranije navedene u vezi naredbe "plot" rade i u slučaju logaritamskih grafika, kao*

*axis, grid, colour, hold, itd.*

```
>>grid
```

*Grafici u polarnom koordinatnom sistemu*

```
>>theta=t/3; % u radijanima !
```

```
>>polar(theta,t)
```

```
>>title('Polarni koordinatni sistem')
```

```
>>grid
```

*Prethodna tri koraka u jednom:*

```
>> polar(theta*2,t); title('My Second Polar Plot'); grid;
```

*Specijalni grafici:*

```
>>bar(t(1,1:20),theta(1,1:20))
```

```
>> stairs(t(1,1:20),theta(1,1:20))
```

Generisanje više grafika je moguće na dva načina: više grafika u istom grafičkom prozoru ili generisanje u posebnim prozorima.

Za više grafika u jednom grafičkom prozoru treba podeliti grafički prozor korišćenjem naredbe "subplot(mnp)", čime je jedan prozor podeljen na  $m \cdot n$  prozora, dok  $p$  označava potprozor u kome se crta grafik pomoću naredbi koje slede

```
>>t=0:0.1:10;
>>x=sin(t);
>>y=cos(t);
>>subplot(211)
>>plot(t,x); title('Sinusna funkcija');
>>subplot(212)
>>plot(t,y); title('Kosinusna funkcija');
```

Novi prozor:

```
>>figure(2)
>>plot(t,x)
>>figure(1)
>>plot(t,y)
```

Naredba za brisanje sadržaja grafičkog prozora

```
>>clg
```

### **FORMIRANJE FUNKCIJSKIH DATOTEKA**

Matlab predstavlja moćan programski jezik. Formiranje funkcijske  $m$ .datoteke omogućava njeno korišćenje kao da je ugrađena funkcija.

Prost primer. Otvorite editor i otkucajte:

```
function povrsina=c_povrsina(radius)
povrsina=pi*radius^2
```

Snimite datoteku pod nazivom "c\_povrsina.m". Vratite se u komandni prozor Matlaba:

```
>>radius=45;
>>povrsina=c_povrsina(radius)
```

Imena promenljivih su potpuno proizvoljna:

```
>>diam=10;
>>velicina=c_povrsina(diam/2)
```

U sledećem primeru formiraćemo funkciju više argumenata. Otkucajte u editoru:

```
function f=sawtooth(t,T)
% SAWTOOTH generator zupcastih talasa
% SAWTOOTH(t,T) izracunava vrednosti za talas
% periode T po promenljivoj t
f=10*rem(t,T)/T;
```

Posle snimanja funkcijske datoteke pod imenom "sawtooth.m", u komandnom prozoru:

```
>>t=0:0.1:10;
>>f=sawtooth(t,2);
>>plot(t,f)
```

Komentar ima funkciju helpa:

```
>>help sawtooth
```

Matlab prikazuje prvi niz redova komentara koji je sastavni deo funkcijske datoteke kao pomoć za tu funkciju. U suštini, Matlab je sagrađen od funkcija koje su napisane upravo na ovaj način, korišćenjem osnovnih komandi.

## RELACIONI OPERATORI

U Matlabu je definisano 6 relacionih operatora

<u>operator</u>	<u>opis</u>
<	manje
<=	manje ili jednako
>	veće
>=	veće ili jednako
==	jednako
~=	nije jednako

Nula je oznaka za logički netačno.  
Jedinica je oznaka za logički tačno.

Primeri formiranja logičkih izraza:

```
>> a=[2 4 6]; b=[3, 5, 1];  
>> a<b  
[1 1 0]  
>>a ~= b  
[1 1 1]
```

I ovde treba uočiti da relacioni operatori u logičkim izrazima rade samo kada su matrice istih dimenzija

## LOGIČKI IZRAZI

U Matlabu su definisana 3 logička operatora:

<u>operator</u>	<u>opis</u>
&	logičko I
	logičko ILI
~	logičko NE

Nula označava logički netačno. Svaki drugi broj (uključujući negativne brojeve) označava logički tačno.

Na primer:

```
>> a=[1 -2 4]; b= [ 0 2 3];  
>> a&b  
[0 1 1]
```

Primer 2:

```
>> a=[2 4 6]; b=[3 5 1]; c=[4 2 2];  
>> a<b & b<c  
[1 0 0]
```

Primer 3:

```
>> ~(b==c | b<a)
```

Primer 4:  
>> a>b&b>c

## **OSNOVE PROGRAMIRANJA U MATLABU (FOR; WHILE; IF-THEN-ELSE)**

- For petlje
- While petlje
- If-Then-Else strukture

### **FOR petlje**

*Jednostavan primer*

```
>>for n=1:10  
x(n)=sin(n*pi/10);  
end  
>>
```

*Obratite pažnju **end** – se obavezno stavlja na kraj ciklusa.*

*Dodeljivanje vrednosti promenljivoj  $n=1:10$  je uobičajeno*

```
>>n=1:10
```

*Moguće je odrediti i korak:*

```
>> for n=1:2:20  
y(n)=sin(n*pi/10);  
end  
>>
```

*Interesantan je sledeći primer*

```
>>data=[3 9 45 6; 7 16 -5 5]  
>>for n=data  
x=n(1)-n(2);  
end  
>>
```

*Generalno pravilo*

*for n=vector*

*[komande koje se odnose na izračunavanje n-tog elementa vektora]  
end*

*For petlje jedne unutar druge:*

```
>> for n = 1:5  
for m = 5:-1:1  
A(n,m)=n^2+m^2;  
end  
disp(n)  
end
```

*Izvršavanje ciklusa se ne može prekinuti dodeljivanjem krajnje vrednosti promenljivoj*

```
>>for n=1:10  
x(n)=sin(n*pi/10);  
n=10; % ova linija nema nikakvog efekta  
end  
>>
```

*For petlje treba izbegavati kad god je to moguće:*

```
>>n=1:10;
```

```
>>x=sin(n*pi/10);
Da bi se postigla veća brzina pri izračunavanju alokaciju prostora treba izvršiti unapred
>>z=zeros(1,10); % alokacija prostora unapred
>>for n=1:10
z(n)=sin(n*pi/10);
end
```

### WHILE petlje

*For petlje izvršavaju grupu komandi unapred određeni broj puta.  
While petlje izvršavaju grupu komandi beskonačno mnogo puta:*

```
while {izraz}
[niz komandi]
end
```

*gde je {izraz} bilo koji logički izraz koji za rezultat ima tačno ili netačno. Na primer:*

```
>> num=0; EPS=1;
>> while (1+EPS)>1
EPS=EPS/2;
num=num+1;
end
>>
>> num
num= 53
>> EPS=2*EPS
EPS= 2.2204e-16
>>
```

*Metod za izračunavanje rezolucije u Matlabu.*

### IF-THEN-ELSE

*Jednostavan if struktura:*

```
if {izraz}
[komande]
end
```

*Primer:*

```
>>racunari=10;
>>cena=racunari*25
cena=250
>> if racunari>5 % 20% popust na količinu
cena=(1-20/100)*cena;
end
```

```
>>cena
cena=200
```

**Else:**

```
if {izraz}
[komande]
else
[komande]
end
```

*Primer:*

```
>>if racunari > 5
cena=(1-20/100)*racunari*cena_po_racunaru;
else
cost=racunari*cena_po_racunaru;
end
```

## **IF-THEN-ELSE**

*Opšta forma IF-ELSEIF-ELSE:*

```
if {izraz1}
[komande 1]
elseif {izraz 2}
[komande 2]
else if {izraz 3}
[komande 3]
.....
else
[komande n]
end
```

Način da se prekine petlja:

```
>>EPS=1;
>> for num=1:1000
EPS=EPS/2;
if (1+EPS) <= 1
EPS=EPS*2;
break;
end
end
>>EPS
EPS=2.2204e-16
>>num
num=53
```

**Primer:**

```
function f=mmono(x)
% MMONO Test monotonosti vektora.
% MMONO(X) X je ulaz i označava vektor, izlazi su:
% 2 ako je X strogo rastući
% 1 ako je X neopadajući
% -1 ako je X nerastući
% -2 ako je X strogo opadajući
% 0 u ostalim slučajevima
x=x(:); % x se transformiše u vektor kolonu
y=diff(x); % nalazi razliku između dva susedna elementa
if all(y>0) % test da li su sve razlike veće od nule
f=2;
elseif all(y>=0)
f=1;
elseif all(y<0) % test da li su sve razlike manje od nule
f=-2;
elseif all(y<=0)
```

Profil: MEHATRONIKA  
Predmet: "Sistemi upravljanja u mehatronici"  
I deo Uvod u Matlab



```
f=-1;
else
f=0;
end
>>mmono(1:12)
ans=2
>>mmono([1:12 12 13:14])
ans=1
>>
```

## **POLINOMI**

*U Matlabu, polinomi se predstavljaju vektorom vrstom njegovih koeficijenata.*

*Na primer, polinom*

$$x^4 - 12x^3 - 25x + 116$$

*prestavljen je vektorom vrstom čiji su elementi koeficijenti polinoma*

```
>> p=[ 1 -12 0 25 116]
```

*Napomena: Elementi vektora vrste koji predstavljaju koeficijente polinoma navode se počev od najvišeg stepena (nula se stavlja na mesto koeficijenta koji je jednak nuli).*

*Matlab poseduje veliki broj funkcija za rad sa polinomima. Na primer, koreni polinoma se određuju pomoću:*

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

```
r=
```

```
11.7473
```

```
2.7028
```

```
-1.2251 + 1.4672i
```

```
-1.2251 - 1.4672i
```

*Po konvenciji, u Matlabu se polinomi predstavljaju vektorom vrstom, a koreni polinoma vektorom kolonom.*

### **Polinomna algebra**

*Polinom se formira na osnovu datih nula pomoću funkcije **poly**.*

```
>>pp=poly(r)
```

*Ponekad treba iskoristiti komandu **real** (da bi koeficijenti postali realni):*

```
>> pp=real(pp)
```

### **Množenje polinoma: funkcija conv**

*Na primer:  $a(x)=x^3 + 2x^2 + 3x + 4$ ,  $b(x) = x^3 + 4x^2 + 9x + 16$*

```
>>a=[1 2 3 4]; b=[1 4 9 16]
```

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

```
c= 1 6 20 50 75 84 64
```

```
>>
```

*Rezultat je*

$$c(x) = x^6 + 6x^5 + 20x^4 + 50x^3 + 75x^2 + 84x + 64$$

**Sabiranje polinoma:** *Pod uslovom da su polinomi istog reda, onda se uobičajena operacija sabiranja vektora koristi za sabiranje polinoma*

```
>> d=a+b;
```

*Ako su polinomi različitog reda potrebno je dovesti ih na isti red jednostavnom transformacijom:*

```
>>e=c+[0 0 0 d]
```

### **Deljenje polinoma: komanda deconv**

```
Profil:      MEHATRONIKA
Predmet:    "Sistemi upravljanja u mehatronici"
I deo      Uvod u Matlab
```

```
>> [q,r]=deconv(c,b)
q=
1 2 3 4
r=
0 0 0 0 0
```

Dakle, rezultat deljenja polinoma **c** polinomom **b** dat je količnikom **q** i ostatkom **r**.

Matlab poseduje specijalnu funkciju **polyval** za sračunavanje vrednosti polinoma u zadatim tačkama

```
>>x=linspace(-1,3); % generiše 100 tačaka u intervalu [-1,3]
>>p=[1 4 -7 -10]; % polinom x^3 + 4x^2-7x-10
>>v=polyval(p,x);
>>plot(x,v); title('x^3+4x^2-7x-10');
```

Komanda **residue** se koristi za rastavljanje količnika dva polinoma u zbir parcijalnih razlomaka

```
>>num=10*[ 1 2]; den=poly([-1; -3; -4]);
>>[res,poles,k]=residue(num,den)
```

## Aproksimacija podataka polinomom

Neka su dati podaci u vektorima

```
>>x=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]
>>y=[-0.447 1.978 3.28 6.16 7.08 7.34 7.66 9.56 9.48 9.30 11.2]
```

Za nalaženje njihove aproksimacije u obliku polinoma zadanog reda koristi se komanda **polyfit**

```
>>n=2;
>>p=polyfit(x,y,n)
p= -9.8108 20.1293 -0.0317
```

Ovaj rezultat odgovara polinomu  
 $-9.8108x^2 + 20.1293x - 0.0317$

Nacrtajmmo grafik:

```
>>xi=linspace(0,1,100);
>>z=polyval(p,xi);
>>plot(x,y,'o',x,y,xi,z,':')
```

Postavlja se pitanje koji red polinoma treba izabrati za date podatke.

Proba sa polinomom 10-tog reda

```
>>pp=polyfit(x,y,10);
>>zz=polyval(p,xi);
>>plot(x,y,'o',xi,z,':xi,zz,':')
```

## Jedno-dimenzionalna interpolacija

Ukoliko se drugačije ne specificira, Matlab sprovodi linearnu interpolaciju između susednih podataka. Na primer:

```
>>x1=linspace(0,2*pi,60);
>>x2=linspace(0,2*pi,6);
>>plot(x1,sin(x1),x2,sin(x2),'-');
```

Ukoliko su nam potrebni boji rezultati nego što daje linearna interpolacija, na raspolaganju nam je ugrađena funkcija **interp1**:

```
Profil:      MEHATRONIKA
Predmet:    "Sistemi upravljanja u mehatronici"
I deo      Uvod u Matlab
```

```
>>vreme=1:12; % mora da bude monoton
>>temp=[5,8,9,15,25,29,31,30,22,25,27,24];
>>plot(vreme,temp,vreme,temp '+' );
```

*Linearna interpolacija:*

```
>>t=interp1(vreme,temp,9.3) % temperatura u 9.3
t=22.9000
>> t=interp1(vreme,temp,[3.2 6.5 7.1 11.7])
t=10.2 30.0 30.9 24.9
```

*interp1 može da koristi kubni splajn za "glatkije" rezultate*

```
>> t=interp1(vreme,temp,[3.2 6.5 7.1 11.7],'spline')
t=10.2 30.0 30.9 24.9
```

*Grafik:*

```
>>h=1:0.1:12;
>>t=interp1(vreme,temp,h,'spline');
>>plot(vreme,temp,'-',vreme,temp,'+',h,t)
```

### **Dvodimenzionalna interpolacija**

*Dvodimenzionalnu interpolaciju primenjujemo u slučaju funkcije dve promenljive  $z=f(x,y)$*

*Raspodela temperature u ploči:*

```
>>sirina=1:5; % indeks sirine ploče
>>dubina=1:3; % indeks dubine ploče
>>temp=[82 81 80 82 84;
79 63 61 65 81;
84 84 82 85 86];
```

*Interpolacija u jednom pravcu:*

```
>>wi=1:0.2:5; % tacke po sirini
>>d=2; % na dubini 2
>>zlinear=interp2(sirina,dubina,temp,wi,d); % linear
>>zcubic=interp2(sirina,dubina,temp,wi,d,'cubic');
>>plot(wi,zlinear,'-',wi,zcubic); grid;
>>title(['Temperatura na dubini = ' num2str(d)])
```

*Interpolacija po dve dimenzije:*

```
>>mesh(dubina,sirina,temp)
>>di=1:0.2:3;
>>wi=1:0.2:5;
>>zcubic=interp2(dubina,sirina,temp,wi,di,'cubic');
>>mesh(wi,di,zcubic)
>>xlabel('Sirina ploce'); ylabel('Dubina ploce');
>>zlabel('Stepeni Celzijusa'); axis('ij'); grid;
```