

## Vježba 1. Uvod u MATLAB

### Uvodne napomene, vektori, matrice

MATLAB se pokreće biranjem **Start**→**Programs**→**Matlab**. Nakon ovoga otvara se komandni prozor i pojavljuje MATLAB prompt `>>`.

Iz MATLAB-a se može izići kucanjem komande:

```
>>quit
```

U MATLAB je ugrađen i obiman sistem pomoći koji se dobija komandom:

```
>>help
```

Odnosno, ako znate ime funkcije, ali se ne možete sjetiti tačne sintakse:

```
>>help ime_funkcije
```

Većina matematičkih funkcija je ugrađena u MATLAB i obično je korisno da se komanda `help` izvrši za željenu funkciju da biste se upoznali sa njenim mogućnostima.

Osnovna struktura podataka u MATLAB-u je matrica. Specijalni slučajevi su  $1 \times 1$  matrica (skalar) i matrice čija je jedna dimenzija 1 – vektori.

Matrice se mogu unijeti:

1. nabranjem svih elemenata,
2. učitavanjem matrice iz fajla,
3. generisanje pomoću ugrađenih funkcija,
4. generisanje pomoću sopstvenih (korisničkih) funkcija.

```
>>x = 5
```

```
>>b = [1 3 8 12.1 7]
```

```
>>A = [1, 7, 8; 2, 9, 13; 4, 8, 11; 2, 5, 3]
```

MATLAB nakon izvršenja svake komande ispisuje rezultat. To se može izbjeći ako se linija završi sa ;  
Pojedininim elementima matrice može se pristupiti navođenjem njihovih indeksa:

```
>> b(1)
```

```
ans =
```

```
1
```

```
>> A(1,2) = 1;
```

```
>> A(1,:) = [1, 7, 8];
```

```
>> A(:,1) = [1; 2; 4; 2];
```

Koristan način za zadavanje vektora može biti operator dvotačka : čija je sintaksa

```
x = start:korak:kraj
```

Na primjer:

```
» x = 3:8
```

```
x =
```

```
3 4 5 6 7 8
```

```
» x = 2:2:10
```

```
x =
```

```
2 4 6 8 10
```

Postoji i niz funkcija koje generišu karakteristične matrice. Dvije često korištene su

```
» A = zeros(n,m);
```

```
» B = ones(n,m);
```

Funkcija `zeros` generiše matricu dimenzija  $n \times m$  čiji su svi elementi nule, a `ones` matricu istih dimenzija čiji su svi elementi jedinice. Ovo je vrlo korisno kod zadavanja signala.

Pored promjenljivih definisanih na pomenute načine u MATLAB-u postoje i neke specijalne predefinisane promjenljive:

Specijalne promjenljive	Opis
<code>ans</code>	Rezultat poslednje operacije
<code>pi</code>	Broj $\pi$
<code>eps</code>	Najmanji broj koji kada se doda jedinici, daje broj koji je na računaru veći od jedinice.
<code>flops</code>	Broj floating-point operacija
<code>inf</code>	Beskonačno (npr. $1/0$ ).
<code>NaN (or) nan</code>	Nije broj – Not-a-Number (npr. $0/0$ ).
<code>i (and) j</code>	$i = j = \sqrt{-1}$
<code>realmin</code>	Najmanji mogući realan broj
<code>realmax</code>	Najveći upotrebljiv pozitivan realni broj

## Osnovne operacije i funkcije nad matricama

Osnovne aritmetičke operacije  $+$ ,  $-$ ,  $*$ ,  $/$  mogu da se koriste na isti način kao i u drugim višim programskim jezicima s tim što njihovi argumenti mogu biti i vektori, odnosno matrice. Isto vrijedi i za elementarne funkcije: `exp`, `log`, `sqrt`, `sin`, `cos` itd. Ako su argumenti ovih funkcija matrice onda se elementarne funkcije izračunavaju po elementima.

Transponovanje niza/matrice vrši se pomoću operatora `'`

```
» C = C'
```

```
C =
```

```
1
```

```
2
```

```
3
```

4  
5  
6  
7  
8

Da biste izvršili skalarnu operaciju na vektoru koristite obične matematičke operacije. Na primjer, da biste pomnožili svaki element vektora C sa 2 koristite komandu:

```
» C = 2*C
```

```
C =
```

```
2 4 6 8 10 12 14 16
```

Pretpostavimo da želite da pomnožite svaki element vektora a odgovarajućim elementom vektora b. U tom slučaju koristite operator `.` koji izvršava operacije po elementima. Na primjer:

```
» a = [2 2 2 2]
```

```
a =
```

```
2 2 2 2
```

```
» b = [1 2 3 4]
```

```
b =
```

```
1 2 3 4
```

```
» a.*b
```

```
ans =
```

```
2 4 6 8
```

Ako biste koristili obično množenje `*` dobili biste:

```
» a*b
```

```
??? Error using ==> *
```

```
Inner matrix dimensions must agree.
```

zato što MATLAB pokušava da pomnoži matrice a i b, prema pravilima za množenje matrica.

Inverzna matrica se izračunava korištenjem funkcije `inv`, a kod rješavanja sistema linearnih jednačina korisni su operatori `/` i `\` čija značenja su:

```
>> F = D / E % F = DE-1
```

```
>> F = D \ E % F = D-1E
```

Konkatenacija nizova je veoma jednostavna:

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

```
>> B = [5 6 7 8];
```

```
>> C = [A B]
```

```
C =
```

```
[1 2 3 4 5 6 7 8]
```

U sledećoj tabeli date su još neke korisne funkcije:

Funkcija	Opis
sum(A)	suma elemenata matrice A po kolonama
diag(A)	elementi glavne dijagonale matrice A
max(A)	vektor najvećih elemenata matrice A po kolonama
min(A)	vektor najmanjih elemenata matrice A po kolonama
mean(A)	srednja vrijednost elemenata matrice A po kolonama
size(A)	dimenzije matrice A
length(x)	dužina (broj elemenata) vektora x

## Polinomi

Polinomi se u MATLAB-u predstavljaju pomoću nizova. Jedna mogućnost je da niz sadrži koeficijente polinoma od člana najvišeg do člana najnižeg stepena. Ako član ne postoji, unosi se vrijednost koeficijenta 0. Na primjer, ako je dat polinom:

$$y(x) = 9x^3 + 5x + 1,$$

njegova reprezentacija u MATLAB-u bila bi:

```
» y = [9 0 5 1]
```

Drugi način da se zadaju polinomi je pomoću njihovih korijena. Na primjer:

$$y(x) = (x + 3)(x - 5)(x + 9),$$

se u MATLAB-u predstavlja kao:

```
» y = [-3 5 -9]
```

Korisne funkcije pri radu sa polinomima su `roots` i `poly`. Funkcija `roots(y)` pronalazi korijene polinoma i kao rezultat vraća vektor korijena. Funkcija `poly(y)` radi suprotno. Ona uzima vektor korijena polinoma i kao rezultat vraća vektor koeficijenata.

Za množenje dva polinoma koristi se funkcija `conv(y, z)`, koja uzima dva vektora koeficijenata polinoma i kao rezultat vraća vektor koeficijenata polinoma koji je njihov proizvod. Ova funkcija se, jasno, koristi i za izračunavanje konvolucije nizova y i z.

## Kompleksni brojevi

MATLAB ima ugrađenu klasu kompleksnih brojeva, a aritmetički operatori i elementarne funkcije mogu kao argumente da prime kompleksne brojeve, odnosno, promjenljive ove klase. Predefinisano je da je imaginarna jedinica dodijeljena promjenljivima `i` i `j`, ali vrijednosti ovih promjenljivih je moguće i promijeniti, a takođe je moguće i nekoj drugoj promjenljivoj dodijeliti imaginarnu jedinicu. Sada je kompleksne brojeve moguće zadati na neki od sledećih načina:

```
>> k = sqrt(-1);
```

```
>> z = 2 + 3*j;
```

```
>> u = 1.2 + j*3.4;
```

```
>> q = 1.4 + 4.2i;
```

Treba obratiti pažnju da u poslednjem izrazu nema operatora množenja `*`. Međutim, izraz

```
>> v = 3 + i2;
```

vjerovatno ne bi doveo do željenog rezultata zato što je `i2` validan identifikator.

Realni i imaginarni dio kompleksnog broja mogu se dobiti funkcijama `real` i `imag`, a moduo i argument funkcijama `abs` i `angle`, respektivno.

## Pisanje programa u MATLAB-u

Programi u MATLAB-u su tekstualni fajlovi koji sadrže MATLAB komande. Ekstenzija ovih fajlova je `.m`, pa se zbog toga oni ponekad zovu i m-fajlovi. Ovi fajlovi mogu da se kreiraju pomoću bilo kojeg editora teksta, a može da se i koristi MATLAB-ov ugrađeni editor/debuger koji postoji od verzije 5.

Da biste izvršili program u MATLAB-u, pređite u direktorijum u kojem se nalazi m-fajl i otkucajte ime fajla bez ekstenzije. Na primjer, ako je ime fajla `lab1.m`, program pokrećete kucajući

```
» lab1
```

Ukoliko se u fajlu nalaze validne MATLAB komande i funkcije one će se izvršiti i promjenljive koje su kreirane nalaziće se u radnoj memoriji MATLAB-a i sa njima se može dalje raditi na uobičajen način.

### Naredbe za kontrolu toka

Osnovna naredba za kontrolu toka je `if` čija je sintaksa:

```
if logički_izraz1
    naredbe1
elseif logički_izraz2
    naredbe2
else
    naredbe3
end
```

Postoji i `case` naredba, a zainteresovani studenti mogu više podataka o njoj naći u MATLAB-ovoj dokumentaciji.

U MATLAB-u takođe postoje i osnovne petlje: `while` čija je sintaksa:

```
while logički_izraz
    naredbe
end
```

I `for` petlja čija je sintaksa:

```
for brojač = start:korak:kraj
    naredbe
end
```

### Funkcije u MATLAB-u

Pošto je MATLAB potpun programski jezik u njemu je moguće pisati i funkcije. Kao primjer data ja funkcija `myfac` koja računa faktorijel argumenta.

```
function r = myfac( k )
% Funkcija za racunanje faktorijela (nerekurzivna)
```

```

if k<0
    r = 0;
    return;
end
r = 1;
while k>1
    r = r * k;
    k = k - 1;
end

```

Važno je primijetiti da naredba `return` u MATLAB-u ne služi za vraćanje vrijednosti iz funkcije već samo za izlazak iz nje, tj. za prekid izvršavanja. Izlazni argumenti se navode u definiciji funkcije (u našem slučaju to je `r`) i vrijednost koju funkcija vraća treba dodijeliti izlaznoj promjenljivoj (argumentu).

U MATLAB-u je najbolja praksa da se svaka funkcija nalazi u sopstvenom m-fajlu čije ime je isto kao i ime funkcije, npr. `myfac.m` pošto MATLAB funkcije pronalazi po imenu fajla u kojem se nalaze.

## Grafičko prikazivanje podataka

Osnovna funkcija za grafički prikaz podataka je `plot`. U zavisnosti od ulaznih argumenata dobijaju se različiti rezultati. Na primjer,

```

» x = [1, 7, 5, 4.3, 2, 9, 11, 8.8];
» plot(x)

```

će nacrtati tačke iz vektora `x` u funkciji njihovog indeksa i povezati ih pravolinijskim segmentima, a

```

» plot(x, 'o')

```

će nacrtati tačke označene kružićima i neće ih povezati.

### Primjer

Nacrtati funkciju  $y = x^2$  na intervalu  $[-2, 2]$ .

```

% vektor vrijednosti u kojima se izračunavaju vrijednosti funkcije
% potrebno je uzeti relativno malu vrijednost koraka kako bi grafik
% izgledao gladak
x = -2:0.01:2;
% izračunavanje vrijednosti funkcije u zadatim tačkama
y = x.^2;
% crtanje grafika
plot(x,y);

```

Moguće je na jednoj slici nacrtati više grafika i MATLAB ih crta različitim bojama iz predefinisano skup.

```

y1 = x.^2 - 4;
y2 = x.^2 + 4;
plot(x,y,x,y1,x,y2);

```

Boja, oblik tačkica, tip linija se mogu zadati eksplicitno, pogledati help stranicu za komandu `plot`.

Komanda `plot` automatski otvara novi prozor, ako ne postoji ni jedan već otvoren. Ako je neki grafički prozor otvoren, `plot` crta u njega, a prethodni sadržaj se briše. Novi prozor se otvara pomoću komande `figure` i on postaje aktivan. Sledeća `plot` komanda crta u njega. Već postojeći prozor se aktivira sa `figure(n)`, gdje je `n` broj prozora koji se nalazi u naslovnoj liniji, npr. Figure No.1

Pomenuto je da se crtanjem u već postojeći prozor njegov prethodni sadržaj briše. Ovakvo se ponašanje može promijeniti pomoću komande

```
>> hold on
```

koja zadržava postojeći grafik i sledeća komanda za crtanje crta preko njega.

Predefinisano ponašanje vraća se sa:

```
>> hold off
```

Graficima se lako mogu dodati naslov, kao i oznake osa.

```
t = -pi:pi/100:pi;
y = sin(t);
plot(t,y);
title('Grafik funkcije sinus');
xlabel('t');
ylabel('sin(t)');
```

Moguće je i u jednom prozor nacrtati više grafika korištenjem naredbe `subplot`. Sintaksa ove naredbe je

```
subplot(mnp)
subplot(m, n, p)
```

i ona dijeli prozor na matricu sa `m` vrsta i `n` kolona, a naredna komanda za crtanje se odnosi na `p`-ti podgrafik. Na primjer:

```
t = -pi:pi/100:pi;
y1 = sin(t);
y2 = cos(t);
subplot(211);
plot(t, y1);
subplot(212);
plot(t, y2);
```

## 3D grafika

MATLAB posjeduje i veoma napredne mogućnosti za 3D inženjersku grafiku. Osnovne naredbe za generisanje 3D grafika su `surf` i `mesh`. Kao demonstracija može da posluži sledeća sekvenca naredbi:

```
surf(40*membrane(1,25));
shading interp;
daspect([1 1 0.9]);
camlight
```

```
axis off tight vis3d;
```

Iz linije sa alatima izabrati Rotate 3D koji omogućava rotiranje prikazanog objekta klikom i pomjeranjem miša.

## Zadaci

Sledeće zadatke riješiti u MATLAB-u:

1. Generisati matricu  $\mathbf{A}$  dimenzija  $3 \times 3$  čiji su svi elementi jednaki 1. Kako biste generisali matricu dimenzija  $100 \times 100$  čiji su svi elementi jednaki 1?
2. Generisati vektor kolonu  $\mathbf{b}$  čiji su elementi 1, -1 i 2.
3. Korištenjem jedne naredbe dodjele dodijelite trećoj vrsti matrice  $\mathbf{A}$  vektor  $[1 \ 0 \ 2]$ , a zatim drugoj koloni dobijene matrice  $\mathbf{A}$  vektor  $\mathbf{b}$ . Ispisati dobijenu matricu  $\mathbf{A}$ .
4. Dodijelite prvom elementu treće vrste dobijene matrice  $\mathbf{A}$  vrijednost  $-0.5$ . Kolika je determinanta nove matrice  $\mathbf{A}$ ? Odrediti njenu inverznu matricu.
5. Riješiti sistem od tri linearne jednačine sa tri nepoznate  $\mathbf{Ax} = \mathbf{b}$ . Ispisati dobijeni vektor  $\mathbf{x}$ .
6. Izračunati vrijednost kompleksnih brojeva:

$$z = \frac{(3 + j6)(6 + j4)}{(2 + j1)j2} + 7 + j10$$

$$y = 0.5 + j6 + 3.5e^{j0.6} + (3 + j6)e^{j0.3\pi}.$$

7. Dat je polinom

$$P(x) = x^3 - 4x^2 - 2x + 6.$$

- a) Nacrtati grafik funkcije  $P(x)$  na intervalu  $[-2, 5]$ ;
  - b) Pronaći nule funkcije  $P(x)$ .
8. Generisati vektor  $t$  sa vrijednostima od 0 do 5 i korakom 0.01. Odrediti i napisati broj elemenata vektora.
  9. Generisati signal  $x(t) = e^{-t} \cos(2\pi t)$  na vremenskom intervalu zadanom vektorom  $t$ . Nacrtati ovaj signal. Adekvatno označiti ose. Ispisati svaki peti član vektora  $x$ .
  10. Generisati signal  $x(t) = te^{jt}$  na intervalu  $[-10, 10]$  sa korakom 0.1. Nacrtati realni i imaginarni dio ovog signala jedan iznad drugog (koristiti `subplot`). Adekvatno označiti ose.
  11. Napisati program za sabiranje članove reda

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots$$

dok suma ne premaši 1.995. Ispisati sumu i broj članova potreban da suma premaši 1.995.

## Izvještaj

U izvještaj sa laboratorijskih vježbi uključiti MATLAB naredbe korištene za rješavanje zadataka iz svake tačke, dobijene grafike i odgovore na postavljena pitanja.