

## 1. RAD S MATRICAMA

Nakon pokretanja MATLAB-a pojavljuje se sučelje podijeljeno u više prozora (slika 2.1.):

- **command window** – prozor koji se koristi za unos varijabli i pokretanje funkcija i M-datoteka,
- **command history** – linije unesene u *command window* prozoru bivaju pohranjene u *command history* prozoru iz kojeg standardnim *copy*, *paste* naredbama možemo ponoviti određene zadane naredbe,
- **launch pad** – omogućava jednostavan pristup dostupnim alatima (*tools*), demo primjerima te dokumentaciji,
- **current directory** – sve datoteke koje se pokreću, učitavaju ili snimaju nalaze se ili će se nalaziti u mapi definiranoj u tom prozoru,
- **workspace** – sadrži skup varijabli trenutno učitanih u memoriji

MATLAB može poslužiti za kao kalkulator:

```
>> 4*25+3
```

```
ans =
```

```
103
```

U radnom prostoru MATLAB-a možemo definirati varijable:

```
>> a=4
```

```
a =
```

```
4
```

```
>> b=25;
```

```
>> c=3;
```

```
>> d=a*b+c
```

```
d =
```

```
103
```

Ukoliko naredba završava sa znakom ";" ne ispisuje se rezultat te naredbe.

Kao i kod svih racunalnih jezika tako i ovdje postoje neka osnovna pravila kod imenovanja varijabli. MATLAB razlikuje velika i mala slova, tako da varijabla *var* nije isto što i varijabla *VAR*. Maksimalni broj znakova kojim

imenujemo varijable je 31, s tim da prvi znak mora biti slovo. Izraze možemo unositi i u više redova, s tim da kao oznaku za nastavak u sljedeći red upotrijebimo niz "...".

```
>> 4 + 5 + 3 ...
```

```
+ 1 + 10 + 2 ...
```

```
+ 5
```

```
ans =
```

```
30
```

### 1.1. UNOS MATRICE

Unos matrice u MATLAB možemo izvršiti na više načina:

- eksplicitno, direktno iz radnog prostora,
- učitavanjem iz datoteke,
- generiranjem matrice pomoću ugrađenih funkcija,
- kreiranjem preko vlastitih M-datoteka

Pri unosu matrice iz radnog prostora mora se voditi računa o nekoliko pravila:

- elemente matrice unosimo unutar uglatih zagrada [ ],
- elemente jednoga retka moramo odvojiti prazninama ili zarezom,
- znakom ";" ili *Enter* definirati ćemo kraj svakog reda

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

```
A =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

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

```
B =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

Vektor, odnosno jednodimenzijско polje unosimo na sljedeći način:

```
>> x=[1 2 3]

x =

    1     2     3
```

Za unos vektora možemo se poslužiti i operatorom ":" na sljedeći način:

```
>> a = 1:8

a =

    1     2     3     4     5     6     7     8

>> b = 0.0 : .2 : 1.0

b =

    0    0.2000    0.4000    0.6000    0.8000    1.0000
```

U prvom slučaju generiramo vektor jedinичnog koraka sa zadanim granicama, a u drugom eksplicitno zadajemo i korak, naveden u sredini. Unosom samo imena varijable ispisat će se njen sadržaj:

```
>> a

a =

    1     2     3     4     5     6     7     8

>> b

b =

    0    0.2000    0.4000    0.6000    0.8000    1.0000
```

Naredbama **who** i **whos** dobit ćemo listu trenutno aktivnih varijabli u memoriji, s tim da će druga dati detaljniji pregled informacija o varijablama.

```
>> who

Your variables are:

A a b
```

```
>> whos

Name      Size      Bytes Class

A         3x3         72 double array
a         1x8         64 double array
b         1x6         48 double array
```

Grand total is 23 elements using 184 bytes

Operator ":" možemo upotrijebiti i za izdvajanje određenog reda ili stupca iz matrice. Sljedeći primjeri ilustriraju tu mogućnost.

```
>> x = A(:,1)
```

```
x =

    1
    4
    7
```

```
>> y = A(2,:)
```

```
y =

    4     5     6
```

U konačnici, tim operatorom možemo izdvojiti dio matrice iz neke veće matrice. Ako definiramo matricu dimenzija 4x3, sljedećim naredbama dobit ćemo ovakve rezultate:

```
>> c = [ -1  0  0
         1  1  0
         1 -1  0
         0  0  2]
```

```
c =

    -1     0     0
     1     1     0
     1    -1     0
     0     0     2
```

```
>> d1 = c(:,2:3)
```

```
d1 =

     0     0
     1     0
```

```
-1  0
 0  2
```

```
>> d2 = c(3:4,1:2)
```

```
d2 =
```

```
 1  -1
 0   0
```

*d1* je matrica dimenzija 4x2 sastavljena od elemenata drugog i treceg stupca matrice *c*, dok je *d2* matrica dimenzija 2x2 definirana presjekom treceg i cetvrtog reda te prvog i drugog stupca.

Dimezije matrice mogu se dobiti kao:

```
size(A)
```

Rezultat možemo pridjeliti varijablama:

```
>> [red,stup]=size(R)
```

```
red =
```

```
 2
```

```
stup =
```

```
 4
```

## 1.2. GENERIRANJE ODREĐENIH TIPICNIH MATRICA

U MATLAB su ugrađene mnoge funkcije za generiranje određenih matrica. Neke od njih su:

- **eye** – jedinica matrica
- **zeros** – matrica popunjena nulama (nul-matrica)
- **ones** – matrica popunjena jedinicama
- **diag** – vraća vektor sastavljen od elemenata na dijagonali
- **triu** – gornji trokutni dio matrice
- **tril** – donji trokutni dio matrice
- **rand** – slučajno generirana matrica
- **hilb** – Hilbertova matrica
- **magic** – magični kvadrat

```
>> ones(3)
```

```
ans =
```

```
 1  1  1
 1  1  1
 1  1  1
```

```
>> eye(2)
```

```
ans =
```

```
 1  0
 0  1
```

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

```
X =
```

```
 1  2  3
 4  5  6
 7  8  9
```

```
>> diag(X)
```

```
ans =
```

```
 1
 5
 9
```

```
>> triu(X)
```

```
ans =
```

```
 1  2  3
 0  5  6
 0  0  9
```

```
>> rand(3:4)
```

```
ans =
```

```
 0.4447  0.9218  0.4057  0.4103
 0.6154  0.7382  0.9355  0.8936
 0.7919  0.1763  0.9169  0.0579
```

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

```
ans =
```

```
 8  1  6
 3  5  7
 4  9  2
```

Matrice se mogu izgradene iz više blokova, kao na primjer u ovom slučaju:

```
>> B = [X, zeros(3,2); ones(2,3), eye(2)]
```

B =

```
1 2 3 0 0
4 5 6 0 0
7 8 9 0 0
1 1 1 1 0
1 1 1 0 1
```

### 1.3. MATRICNE OPERACIJE

Sljedece matricne operacije su dostupne u MATLAB-u:

```
+ zbrajanje
- oduzimanje
* množenje
^ potenciranje
' transponiranje
\ lijevo dijeljenje
/ desno dijeljenje
```

Aritmetičke operacije nad matricama definirane su pravilima linearne algebre. No moguće je i vršiti operacije nad pojedinim elementima matrica. U tom slučaju koristit ćemo prefiks "." u kombinaciji sa znakovima operacija. Tako gore navedenom skupu operacija dodajemo i sljedeće:

```
.*
./
.\
.^
.^
.'
```

Zbrajanje i oduzimanje matrica može se provesti samo nad matricama jednakih dimenzija.

```
>> A=[1 2 3; 3 3 3; 5 3 1]
```

A =

```
1 2 3
3 3 3
5 3 1
```

```
>> B=[2 -3 4; 2 -2 2; 0 4 0]
```

B =

```
2 -3 4
2 -2 2
0 4 0
```

```
>> C = A + B
```

C =

```
3 -1 7
5 1 5
5 7 1
```

Množenje matrica  $C = A * B$  je algebarski produkt matrica A i B. Mora vrijediti da je broj stupaca matrice A jednak broju redova matrice B.

```
>> X = [2 3; 4 -1; 0 7]
```

X =

```
2 3
4 -1
0 7
```

```
>> Y = [5 -6 7 2; 1 2 3 6]
```

Y =

```
5 -6 7 2
1 2 3 6
```

```
>> D = X * Y
```

D =

```
13 -6 23 22
19 -26 25 2
7 14 21 42
```

Transponiranje matrice:

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

A =

```
1 2 3
4 5 6
7 8 9
```

```
>> B = A'
```

```
B =
```

```
1 4 7
2 5 8
3 6 9
```

Za pronalazak inverzne matrice koristit ćemo funkciju **inv**.

```
>> x = rand(3)
```

```
x =
```

```
0.0579 0.0099 0.1987
0.3529 0.1389 0.6038
0.8132 0.2028 0.2722
```

```
>> inv(x)
```

```
ans =
```

```
9.1685 -4.0748 2.3453
-42.7890 15.8005 -3.8103
4.4846 0.4030 -0.4941
```

Slijedi opis operacija koje se provode po odgovarajućim elementima matrica ukoliko su matrice jednakih dimenzija.

```
>> A, D
```

```
A =
```

```
1 2 3
4 5 6
7 8 9
```

```
D =
```

```
1 1 1
```

```
2 2 2
```

```
3 3 3
```

```
>> A .* D
```

```
ans =
```

```
1 2 3
```

```
8 10 12
```

```
21 24 27
```

```
>> A ./ D
```

```
ans =
```

```
1.0000 2.0000 3.0000
```

```
2.0000 2.5000 3.0000
```

```
2.3333 2.6667 3.0000
```

```
>> D ./ A
```

```
ans =
```

```
1.0000 0.5000 0.3333
```

```
0.5000 0.4000 0.3333
```

```
0.4286 0.3750 0.3333
```

```
>> A \ D
```

```
ans =
```

```
1.0000 0.5000 0.3333
```

```
0.5000 0.4000 0.3333
```

```
0.4286 0.3750 0.3333
```

```
>> A .^ D
```

```
ans =
```

```

1 2 3
16 25 36
343 512 729

```

MATLAB koristi i funkcije za operacije nad stupcima matrice. Stupci matrice tretiraju se kao neovisni vektori na koje se primjenjuju zadane operacije:

- **min(x)** – vraća vrijednosti najmanjih elemenata svakog stupca
- **max(x)** – vraća vrijednosti najvećih elemenata svakog stupca
- **sum(x)** – vraća sume svih elemenata pojedinih stupaca
- **prod(x)** – vraća umnožak svih elemenata pojedinih stupaca

```

>> A

A =

    1     2     3
    4     5     6
    7     8     9

>> min(A)

ans =

    1     2     3

>> sum(A)

ans =

   12   15   18

```

Relacijski operatori koje koristimo u MATLAB-u su:

- < manje
- > veće
- <= manje ili jednako
- >= veće ili jednako
- == jednako
- ~= različito

Logički operatori:

- & logičko i
- | logičko ili
- ~ logičko ne

Rješenje relacijskih i logičkih operacija jest 0 ako uvjet nije zadovoljen ili logički izraz nije točan, te 1 u obrnutom slučaju.

```

>> 5 < 4

ans =

    0

>> A

A =

    1     2     3
    4     5     6
    7     8     9

>> B

B =

    1     4     7
    2     5     8
    3     6     9

>> A > B

ans =

    0     0     0
    1     0     0
    1     1     0

>> (5 < 4) & (7 > 3) | ~0

ans =

    1

```

#### 1.4. UCITAVANJE I SPREMANJE RADNOG PROSTORA

Do sada smo se koristili matricama koje smo zadavali iz radnog prostora ili su one nastale kao rješenja nekih računskih operacija. Matrice je moguće i učitati u radnu memoriju, a isto tako sve trenutno aktivne varijable moguće je spremiti u neku vanjsku datoteku. Naredbe koje ćemo koristiti su sljedeće:

- **save** – sprema sve trenutno aktivne varijable u datoteku *matlab.mat*, u trenutno aktivni direktorij iz prozora **current directory**
- **load** – učitava sadržaj iz datoteke *matlab.mat* u radni prostor
- **save filename x y z** – sprema varijable x, y i z u datoteku *filename.mat*
- **load filename** – učitava sadržaj iz datoteke *filename* u radni prostor. Datoteka može biti binarna (*.mat*) ili u ASCII formatu
- **clear x** – briše matricu x iz radnog prostora
- **clear** – briše sve varijable iz radnog prostora

Učitavanje varijabli iz ASCII datoteke kreirane bilo kojim tekst editorom može se pokrenuti i preko padajućeg menija *File* → *Import Data...*, a analogno tomu i spremanje. Bitno je naglasiti da ASCII datoteka mora biti kreirana na način da svaki red u datoteci predstavlja red matrice čiji su elementi odvojeni prazninom. Sadržaj ASCII datoteke kojom se definira matrica dimenzija 3x3 mogao bi biti:

```
2 3 6
3 -1 0
7 0 -2
```

Takvu datoteku možemo nazvati *x.dat*, te naredbom **load x.dat** u radni prostor učitavamo matricu x. Ekstenzija *.dat* nije obavezna, ali je poželjna da se takva datoteka razlikuje od ostalih datoteka (*.m*, *.mat* ...).

Za svaku funkciju, operator, kao i za citave programske pakete u MATLAB-u postoje detaljne upute do kojih se dolazi korištenjem naredbe *help*.

- **help** – daje popis svih programskih paketa,
- **help ops** – daje popis svih operatora,
- **help mldivide** – daje upute za matricno lijevo dijeljenje,
- **help mrdivide** – daje upute za matricno desno dijeljenje,
- **help inv** – daje upute za funkciju **inv** (inverzija kvadratne matrice),
- **help ime\_naredbe** – daje upute za navedenu naredbu i sl.,
- **help elfun** – daje popis svih elementarnih funkcija.

Ukoliko ne znamo točno ime naredbe, vrlo je korisna naredba **lookfor**, koja daje popis naredbi "kandidata" za traženi pojam.

## 2. MATLAB SKRIPTE

MATLAB ima mogućnost razvoja algoritama u vlastitom programskom jeziku.

Datoteke s takvim algoritmima nazivamo M-datoteka (*.m* datoteke). Pri tome možemo razlikovati dvije vrste M-datoteka: skripte i funkcije. Skripte su

skup naredbi koje se prenose i izvršavaju u komandnom prozoru. Funkcije se razlikuju utoliko što primaju ulazne argumente, a vraćaju izlazne argumente.

Skripte su jednostavniji oblik M-datoteke – nemaju ulaznih ni izlaznih argumenata. Korisne su pri automatiziranju niza naredbi pogotovo kada je potrebna njihova višekratna uporaba. MATLAB izvršava skriptu liniju po liniju koristeći pri tome podatke koji se nalaze u radnom prostoru ili podatke koje kreira sama skripta. Sve varijable generirane u skripti ostaju u radnom prostoru i nakon izvršavanja skripte, te se mogu koristiti u daljnjim računima.

Sadržaj skripte potrebno je prvo napisati u nekom tekst editoru. Prilikom unosa naredbi znak **%** označava red komentara. Nakon što smo spremili skriptu u datoteku s ekstenzijom *.m* možemo je izvršiti iz komandnog prozora kao svaku drugu naredbu. Pokažimo to na jednostavnom primjeru.

<b>% Skripta koja iscrtava</b>	<b>% linije komentara</b>
<b>% "flower petal" graf</b>	
 <b>theta = -pi:0.01:pi;</b>	<b>% proračun</b>
<b>rho(1,:) = 2*sin(5*theta).^2;</b>	
<b>rho(2,:) = cos(10*theta).^3;</b>	
<b>rho(3,:) = sin(theta).^2;</b>	
<b>rho(4,:) = 5*cos(3.5*theta).^3;</b>	
<b>for i = 1:4</b>	
<b>    polar(theta,rho(i,:))</b>	<b>% crtanje grafa</b>
<b>    pause</b>	
<b>end</b>	

Pozivajući iz menija naredbu *File/New/M-file* otvaramo tekst editor. Spremimo li taj niz naredbi u datoteku pod nazivom *petals.m*, kreirat ćemo skriptu. Pravilo pisanja skripte nalaže da prvi red (ili više njih) bude komentar. U tom slučaju, naredbom *help petals* ispisuju se redovi komentara u komandnom prozoru. Dakle, na taj način opisujemo što skripta radi.