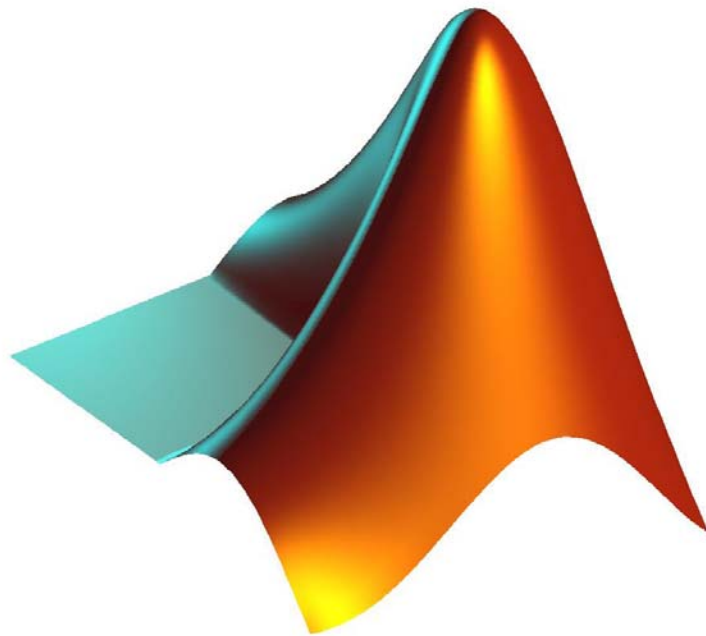


**MATLAB**  
**- priručnik za vježbe -**



**A. Đugum, S. Burak**

## UVOD

### Šta je MATLAB?

MATLAB® (MATrix LABoratory – laboratorija za matrice) je program za tehničke proračune koji ima mogućnost i vizualizacije rezultata kao i programiranja u jednostavnom okruženju.

MATLAB je interaktivni sistem čiji je osnovni element podataka niz koji ne zahtjeva dimenzionisanje. Time je omogućeno rješavanje mnogih tehničkih proračuna u djeliću vremena koje bi trebalo da se napiše program u neinteraktivnom jeziku poput Fortran-a ili C-a. MATLAB takođe sadrži familiju specifičnih rješenja koji se zovu *toolbox*-ovi (alati) koji omogućavaju da se nauči i primjeni specijalizovana tehnologija. Oni predstavljaju kolekciju MATLAB funkcija (*M-files* - *M-datoteke*) koji omogućavaju rješenje specifičnih grupa problema iz područja procesiranja signala, kontrole sistema, akvizicije podataka, statistike, obrade slike, baza podataka, itd.

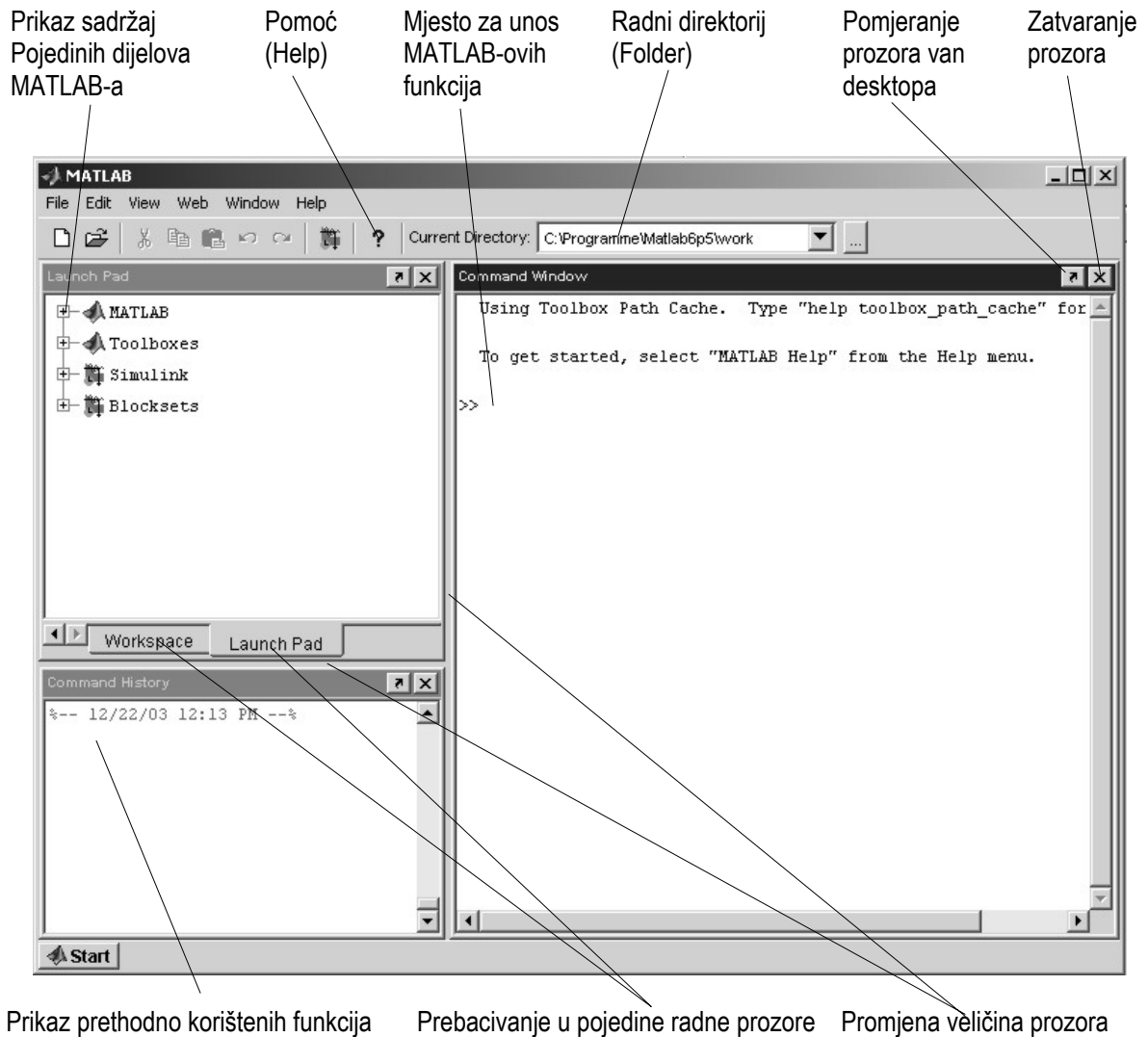
MATLAB sadrži obimnu dokumentaciju za pomoć korisnicima koja se nalazi meniju *Help*. Takođe, tu se nalaze i mnogi demonstracioni programi (birajući *Demos* u meniju *Help*) koji ističu mogućnosti i karakteristike MATLAB-a, njegovih alata i funkcija. Zvanična internet adresa proizvođača MATLAB-a je [www.mathworks.com](http://www.mathworks.com).

### MATLAB desktop

Pokretanjem MATLAB-a pojavljuje se MATLAB desktop koji sadrži alate za upravljanje file-ovima, varijablama i aplikacijama. Prvi put nakon pokretanja, MATLAB desktop će izgledati kako je prikazano na slici 1.

Izgled desktopa može se mijenjati otvaranjem, zatvaranjem, pomijeranjem i promjenom dimenzija alata u njemu. Takođe je moguće pomijerati alate van desktopa i vraćati ih nazad. Određene karakteristike desktopa mogu se podesiti birajući *Preferences* u meniju *File*.

## MATLAB - priručnik za vježbe

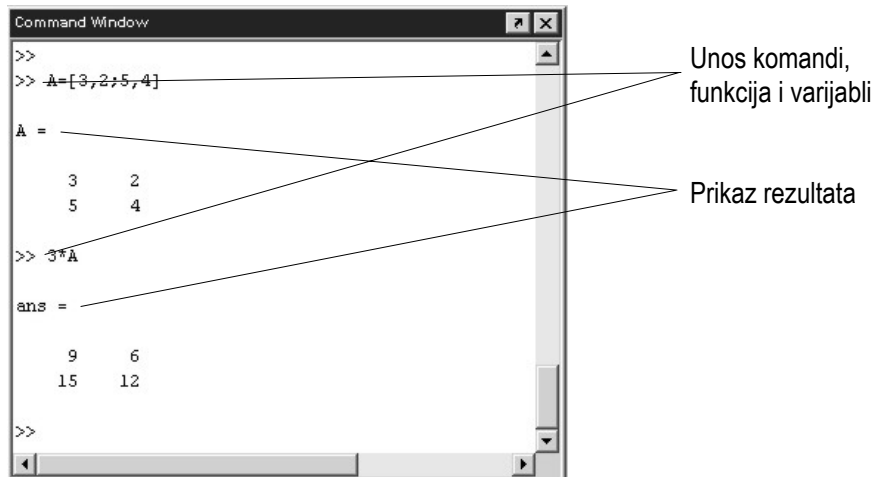


Slika 1. MATLAB desktop

## Desktop alati

U desktop alate spadaju:

- **Komandni prozor** (Command Window)  
Služi za unos varijabli, pokretanje funkcija i M-datoteka.



Slika 2. Komandni prozor

## MATLAB - priručnik za vježbe

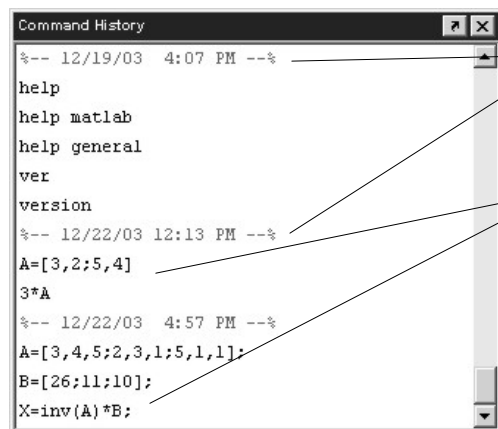
Iz komandnog prozora mogu se takođe pokretati i drugi programi koji nisu u direktnoj vezi s MATLAB-om. Stavljanjem uzvičnika ! na početku linije u komandnom prostoru znači da se ostatak linije odnosi na na komandu operativnom sistemu. Tako npr.

```
!notepad vjezba.m
```

pokreće Notepad editor i otvara dokument vjezba.m.

### - Istorija komandi (Command History)

U njoj su zapisane sve linije unesene u komandni prozor. U tom prozoru se mogu vidjeti prethodno korištene funkcije i ponovo izvršavati željene linije.



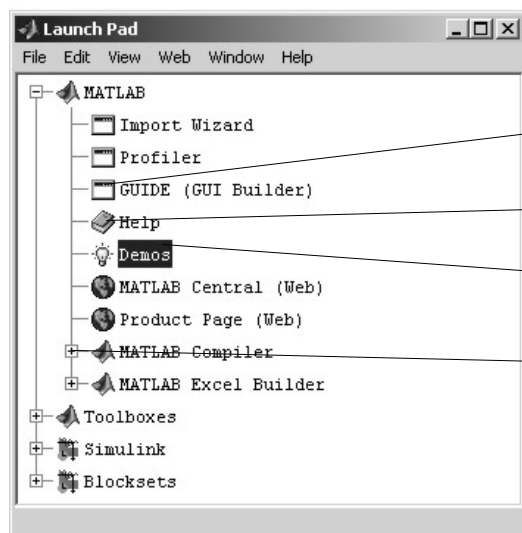
Vrijeme početka rada

Selekcijom jedne ili više linija i desnim klikom na njih možemo kopirati selekciju, ponovo je izvršiti ili napraviti M-datoteku od nje.

**Slika 3.** Istorija komandi

### - Prozor sa aplikacijama (Launch Pad)

Omogućuje lagan pristup alatima, demonstracionim programima i dokumentaciji. U njemu je prikazana lista svih instaliranih MATLAB aplikacija na određenom računaru. Primjer sadržaja prozora s aplikacijama dat je na slici 4.



Alati – otvaraju se dvostrukim klikom


Pomoć

Demonstracioni programi

Klikom na + prikazuje se sadržaj aplikacije

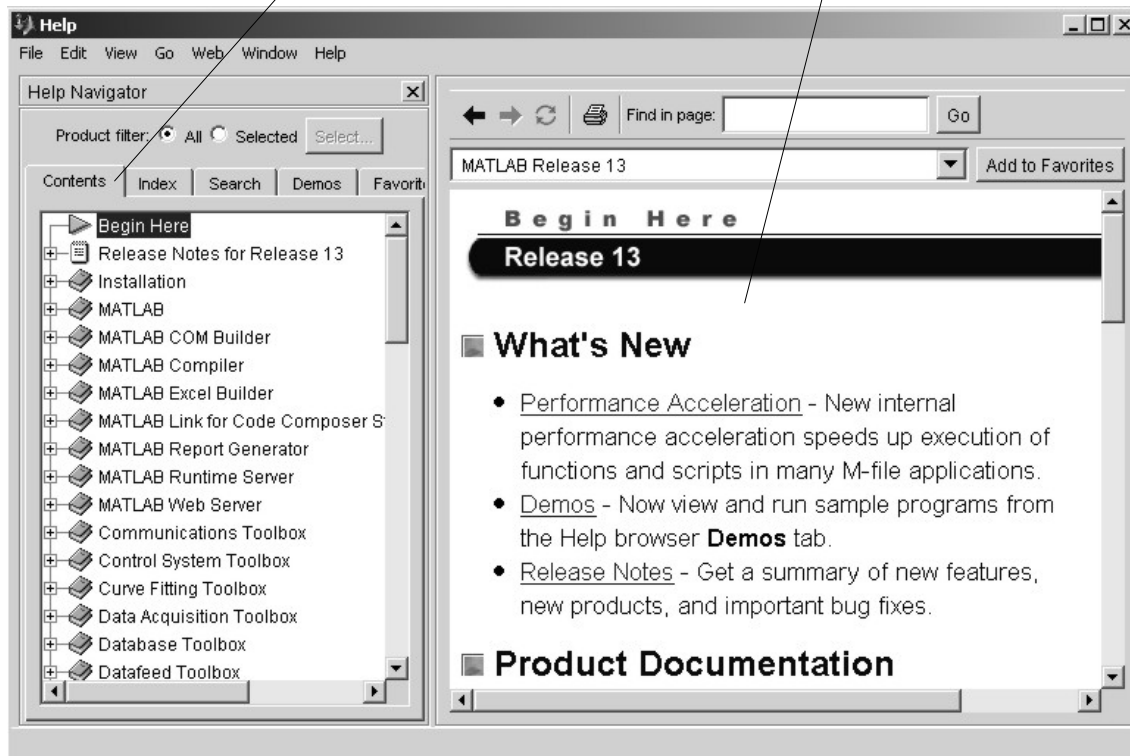
**Slika 4.** Prozor s aplikacijama

- **Pretraživač za pomoć** (Help Browser)

Dokumentacija s uputstvima za upotrebu za sve aplikacije u MATLAB-u nalazi se u pretraživaču za pomoć. Otvara se klikom na ikonicu  na MATLAB desktopu, ili izvršenjem naredbe `helpbrowser` u komandnom prozoru.

Načini traženja dokumentacije u navigatoru za pomoć

Preglednik za pomoć



**Slika 5.** Pretraživač za pomoć

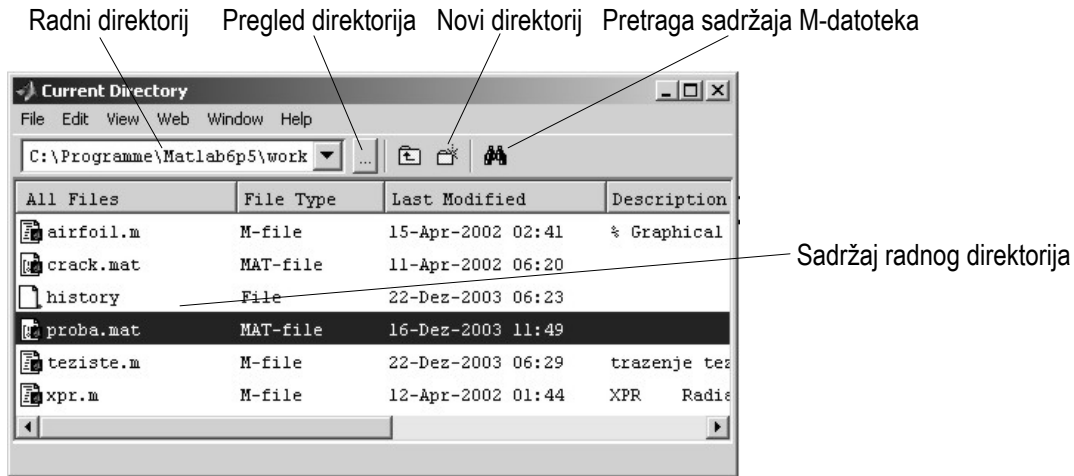
Preglednik za pomoć se sastoji od dva dijela, navigatora za pomoć, koji služi za traženje željenih informacija, i preglednika, u kojem čitamo tražene informacije.

Biranjem odgovarajućih opcija u sklopu navigatora za pomoć imamo više načina da dodjemo do željene informacije. **Product filter** pokazuje dokumentaciju samo za specificirane proizvode. **Contents** prikazuje naslove i sadržaj dokumentacije. **Index** omogućuje pretragu preko odgovarajućih ključnih riječi u sadržaju dokumentacije. **Search** traži željenu frazu unutar dokumentacije. **Favorites** daje listu dokumenata koje smo prethodno označili kao važne.

Nakon nalaženja, željenu dokumentaciju čitamo u pregledniku za pomoć. Pomoću strelica na vrhu i dnu stranice, kao i klikom na strelice "naprijed" i "nazad", prelazimo na druge stranice. Klikom na **Add to Favorites** dugme označavamo važne dokumente da bi im ubuduće imali lakši pristup. Traženje željenog termina ili fraze na stranici vrši se unošenjem iste u **Find in page** polje i klikom na **Go**.

Pomoć također možemo dobiti i korištenjem funkcija za pomoć. Ako želimo dobiti pomoć za određenu funkciju, npr. funkciju `det`, unutar pretraživača za pomoć, u komandnom prozoru otkucamo `doc det`. Za pomoć unutar komandnog prozora treba otkucati `help det`.

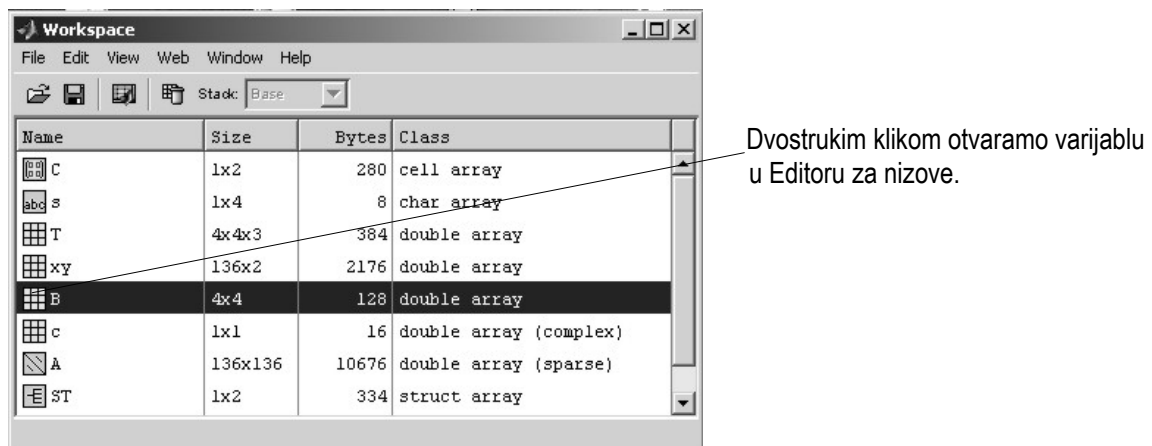
- **Pretraživač radnog direktorija** (Current Directory Browser)  
Služi za laganu pretragu, pregled otvaranje i mijenjanje MATLAB-ovih direktorija i dokumenata. U tu svrhu se takođe mogu koristiti i funkcije `dir`, `cd` i `delete`.



**Slika 6.** Preglednik radnog direktorija

Da bismo izvršili neku funkciju, M-datoteku ili bilo koji MATLAB file, on se mora nalaziti ili u trenutnom radnom direktoriju ili na odgovarajućim lokacijama (direktorijima) u računaru koje se zovu path (put). Za sve funkcije i file-ove koji dodju s MATLAB instalacijom path već postoji. Ako želimo dodati jos neki direktorij u path, odnosno ako želimo da pokrećemo odgovarajuće file-ove iz tog direktorija u MATLAB-u, postavivćemo ga u MATLAB path biranjem **Set Path** iz menija **File**. U komandnom prozoru funkcijom `path` dobijamo listu direktorija koji su u path-u, `addpath` funkcijom dodajemo direktorije u path, a `rmpath` funkcijom ih brisemo iz path-a.

- **Pretraživač radnog prostora** (Workspace Browser)  
Radni prostor se sastoji od varijabli koje su formirane tokom upotrebe MATLAB-a i smještene u memoriju. Varijable dodajemo u radni prostor koristeći funkcije, pokrećući M-datoteke i učitavanjem spašenih radnih prostora. Za pregledanje radnog prostora, osnovne informacije o varijablama, njihovo brisanje ili mijenjanje koristimo pretraživač radnog prostora ili funkcije `who` i `whos`. Funkcijom `clear` brišemo sve varijable iz radnog prostora.

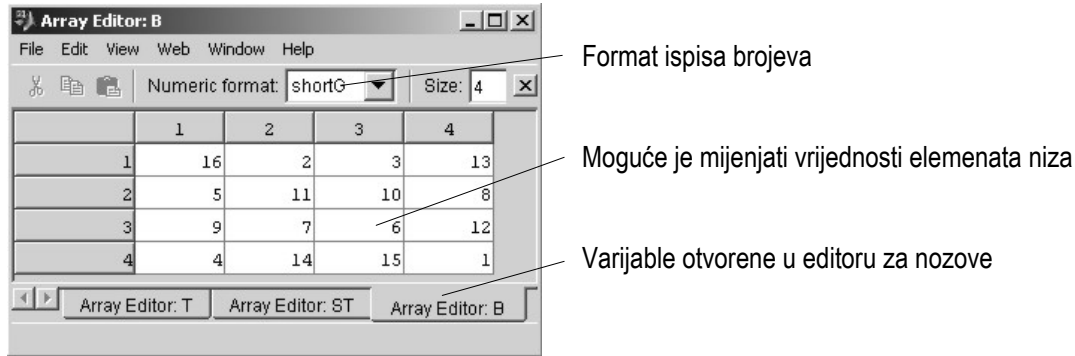


**Slika 7.** Radni prostor

## MATLAB - priručnik za vježbe

Radni prostor nije sačuvan nakon završetka rada u MATLAB-u. Korištenjem funkcije `save` ili biranjem **Save Workspace As** iz menija **File** spašavamo radni prostor u MAT-file koji ima `.mat` ekstenziju. MAT-file učitavamo biranjem **Import Data** iz menija **File** ili korištenjem funkcije `load`.

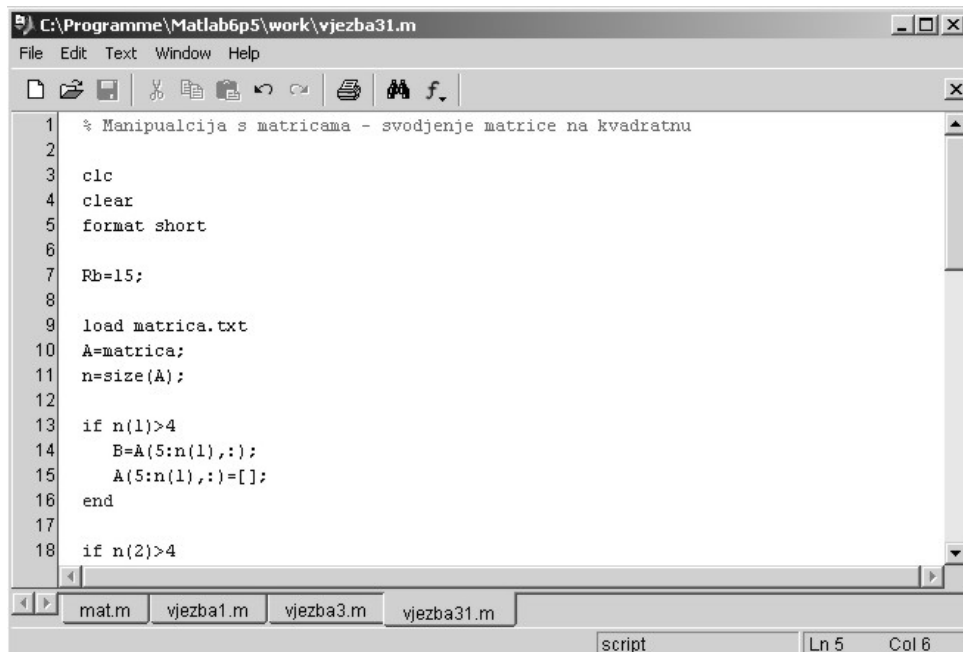
Dvostrukim klikom na varijablu u pretraživaču radnog prostora otvaramo je u editoru za nizove. Time omogućavamo lakše mijenjanje jedno- i dvodimenzionalnih nizova, odnosno varijabli.



**Slika 8.** Editor za nizove

### - **Editor** (Editor/Debugger)

Koristi se za pisanje i testiranje M-datoteka, čijim se pokretanjem izvršava niz u njima napisanih komandi. Za pisanje M-datoteka može se koristiti bilo koji tekst editor, a MATLAB-ov editor prikazan je na slici 8.



**Slika 9.** MATLAB Editor

## M – datoteke (files)

Zadavanje MATLAB komandi direktno u komandnom prozoru očigledno ima svojih nedostataka. Gašenjem računara iz radnog prostora se gube sve varijable, pa bi bilo potrebno ponovo unijeti i izvršiti sve komande.

U cilju prevazilaženja gore navedenih problema, MATLAB omogućuje alternativni rad korištenjem m-datoteka (m-files). U principu, m-datoteke predstavljaju jednostavne tekstualne datoteke u kojima su MATLAB komande upisane redoslijedom kojim trebaju da se izvrše. One moraju imati nastavak (extension) .m da bi ih MATLAB prepoznao kao m-datoteku. Primjer: program1.m, grafik.m itd. Treba imati na umu da postoje dvije vrste M-datoteka, skripte, koje nemaju ulaznih i izlaznih argumenata, i funkcije, koje mogu imati ulazne i izlazne argumente.

Pozivanjem skripte MATLAB izvršava komande redoslijedom kako su napisane u njoj i radeći s već postojećim varijablama iz radnog prostora ili praveći nove. Sve novonapravljene varijable ostaju u radnom prostoru i mogu se koristiti u daljem toku proračuna.

Kao primjer, napravimo M-datoteku slijedećeg sadržaja i nazovimo je operacije.m, i spasimo u direktoriju work (ili nekom drugom direktoriju koji se nalazi u path-u):

```
% Osnovne operacije nad brojevima a i b,
% zbir, razlika, proizvod i kolicnik
z =a + b;
r =a - b;
p =a * b;
k =a / b;
s=sprintf('a=%d, b=%d, \na+b=%d, a-b=%d, a*b=%d, a/b=%d.', ...
          a,b,z,r,p,k);
disp(s)
```

Ako se u radnom prostoru MATLAB-a nalaze varijable a i b (npr. a=6 i b=2), komanda za izvršenje M-datoteke, koja se ukucava u radnom prozoru MATLAB-a ili unutar neke druge M-datoteke, i odgovarajući ispis je:

```
>> operacije
a=6, b=2,
a+b=8, a-b=4, a*b=12, a/b=3.
```

Nakon izvršenja M-datoteke numeričke varijable z, r, p i k, i znakovna varijabla s ostaju u radnom prostoru. Međutim, ako se jedna od varijabli a i b, ili obe, ne nalazi u radnom prostoru MATLAB-a, M-datoteka se ne može izvršiti, odnosno MATLAB javlja grešku.

```
>> operacije
??? Undefined function or variable 'a'.
```

```
Error in ==> C:\Programme\Matlab6p5\work\operacije.m
On line 3 ==> z=a+b;
```



## MATLAB - priručnik za vježbe

Prva linija funkcijske M-datoteke rezervirana je za ključnu riječ `function` iza koje slijedi varijabla (vrijednost te varijable po izvršenju M-datoteke je u stvari rezultat funkcije) kojoj se pridružuje funkcija određenog imena sa svojim argumentima. Prethoda M-datoteka napisan kao funkcija sa ulaznim argumentima `a` i `b`, nazvana `operacije_f.m`, bi izgledao:

```
function s = operacije_f(a,b)
% Osnovne operacije nad brojevima a i b,
% zbir, razlika, proizvod i kolicnik
z = a + b;
r = a - b;
p = a * b;
k = a / b;
s=sprintf('a=%d, b=%d,\na+b=%d, a-b=%d, a*b=%d, a/b=%d.', ...
          a,b,z,r,p,k);
```

Izvršenjem ove M-datoteke

```
>> operacije_f(6,3)
```

```
ans =
```

```
a=6, b=3,
a+b=9, a-b=3, a*b=18, a/b=2.
```

kao rezultat bi dobili samo jednu varijablu `ans` (`answer` = odgovor. Varijabla `ans` poprima vrijednosti izvršenja svih operacija i funkcija koje nisu pridružene nekoj određenoj varijabli), dok su sve ostale varijable unutar funkcije lokalne, odnosno njihove vrijednosti nisu pohranjene u radnom prostoru MATLAB-a i ne mogu se koristiti u daljem toku proračuna nakon izvršenja funkcije. Ako želimo da funkcija `operacije_f.m` ima više izlaznih argumenata, na primjer pored tekstualnog odgovora da još jedna varijabla poprimi vrijednost sume ulaznih argumenata, to bi unutar funkcije zadali na sljedeći način (ostatak funkcije ostaje isti)

```
function [s,z] = operacije_f(a,b)
```

Da bi varijable `tekst` i `suma` poprimile rezultat izvršenja ove funkcije trebamo izvršiti komandu

```
>> [tekst,suma] = operacije_f(a,b)
```

```
tekst =
```

```
a=6, b=3,
a+b=9, a-b=3, a*b=18, a/b=2.
```

```
suma =
```

```
9
```

Unutar tijela svake funkcije na raspolaganju su varijable `nargin` i `nargout` koje imaju vrijednost broja ulaznih odnosno izlaznih argumenata, i mogu biti korisne kod pisanja nekih funkcija.

## MATLAB - priručnik za vježbe

Komentarisanje, odnosno označavanje linija unutar M-datoteke koje nisu komande nego koje služe samo kao pojašnjenja i pomoć, vrši se stavljanjem znaka % ispred teksta komentara. Komentar na samom početku skripti, odnosno komentar odmah ispod prve linije funkcija, predstavlja tekst pomoći i ispisuje se u komandnom prozoru izvršenjem komande

```
help operacije ili help operacije_f
```

Ako želimo samo vidjeti sadržaj M-datoteke, pomoću funkcije `type` ćemo ga ispisati u komandnom prozoru.

```
type operacije ili type operacije_f
```

## MATRICE U MATLAB-U

Dok većina programskih jezika vrši operacije na pojedinačnim broječanim vrijednostima, MATLAB omogućava jednostavan rad sa čitavim matrcama. U MATLAB-u matrica je dvodimenzionalni niz brojeva. Matrice dimenzija 1x1 su skalari, a matrice koje sadrže samo jedan red ili kolonu nazivaju se vektori. Iako MATLAB ima različite mogućnosti memorisanja numeričkih i skalarnih vrijednosti, za početak je najbolje zamisliti sve varijable kao matrice i naučiti operacije nad njima.

### Osnove

U MATLAB-u nije potrebno deklarisanje i dimenzionisanje varijabli prije njihove upotrebe, kao što je to slučaj kod nekih drugih programskih jezika. Imena varijabli moraju početi slovom, a zatim mogu sadržati još i brojeve i donju crticu. MATLAB koristi "samo" prvi 31 znak za ime varijable, što znači da će svi ostali eventualni znakovi iza 31 biti izbačeni iz imena varijable. Treba naglasiti da MATLAB pravi razliku između velikih i malih slova (case sensitive), tako da npr.  $x$  i  $X$  predstavljaju dvije potpuno različite varijable. Primjeri varijabli: `redni_broj`, `RedniBroj`, `Rb`, `a2`, `a2b3`, itd.

Brojevi su predstavljeni *konvencionalnom notacijom*. U slučaju potrebe dodaju se znakovi plus ili minus ispred broja kao i decimalna tačka, a ne zarez (3, -99, 3.1415, -9.81). Vrlo veliki ili mali brojevi predstavljeni su *naučnom notacijom* kod koje se koristi slovo `e` da označi stepen broja 10 ( $1234000=1.234e+6$ ,  $0.00000123=1.23e-6$ ). Sufiksi `i` ili `j` se koriste za pisanje imaginarnih brojeva ( $3i$ ,  $2+2j$ ,  $-3e5i$ ,  $2.21j$ ).

Svi brojevi i proračuni u MATLAB-u rade se dvostrukom preciznošću (double precision) što znači da imaju tačnost od 16 značajnih cifara (broj napisan s 20 cifara gubi podatke o zadnje četiri cifre, praktično postaju nule). Međutim, način ispisa brojeva u komandnom prozoru možemo kontrolisati komandom `format`. Posmatrajmo ispis niza od dva broja `x` u različitim formatima:

```
>>x = [ 2/3 , 1.23e-6 ]

format short
    0.6667    0.0000

format short e
    6.6667e-001    1.2300e-006

format short g
    0.66667    1.23e-006

format long
    0.666666666666667    0.00000123000000

format long e
    6.666666666666666e-001    1.230000000000000e-006
```

## MATLAB - priručnik za vježbe

```
format long g
0.6666666666666667          1.23e-006

format bank
0.67          0.00

format rat
2/3          1/813008
```

Komanda `format compact` izbacuje prazne linije između ispisivanih vrijednosti u komandnom prozoru, tako da u vidljivi dio staje više podataka.

Tipkanjem neke komande ili varijable u komandnom prozoru i pritiskom na tipku *Return* ili *Enter*, MATLAB pohranjuje varijablu ili rezultat izvršenja u svoju memoriju odn. radni prostor, a rezultat prikazuje u komandnom prozoru. Npr.

```
>> x = [ 2/3 , 1/3 ]
x =
0.6667    0.3333
```

Često je korisno spriječiti ispis rezultata na ekranu, posebno ako se komande izvršavaju unutar petlji, jer bi komandni prostor bio "zatrpan" nekorisnim podacima. To činimo stavljanjem znaka `;` iza komande, tj.

```
>> x = [ 2/3 , 1/3 ];
```

U MATLAB je ugrađen veliki broj elementarnih, kao i naprednijih matematičkih funkcija. Neke od tih funkcija su u jezgro MATLAB programa, kao npr. `sin` ili `log`, vrlo su efikasne ali numerički metodi kojima se izračunavaju nisu vidljivi korisniku. Druge funkcije, kao npr. `factor` ili `cross`, su implementirane u m-datotekama, tako da se njihov kod može pogledati pa čak i modificirati. Matlab efikasno radi sa kompleksnim brojevima, tako da određivanje kvadratnog korijena ili logaritma negativnog broja ne signalizira grešku nego daje odgovarajuće kompleksno rješenje. U nastavku će biti objašnjena upotreba jednog dijela tih funkcija, odn. komandi. Međutim, ako želite listu komandi i funkcija koje stoje na raspolaganju korisniku osnovnog MATLAB okruženja (bez okruženja alata – *toolbox*-eva), u komandnom prozoru otkucajte `help` i riječ:

<code>general</code>	- komande osnovne namjene,
<code>ops</code>	- operatori i specijalni karakteri,
<code>elfun</code>	- elementarne matematičke funkcije,
<code>specfun</code>	- specijalizirane matematičke funkcije,
<code>elmat</code>	- elementarne matrice i matični račun,
<code>funfun</code>	- funkcije nad funkcijama i osnovne diferencijalne jednačine,
<code>polyfun</code>	- interpolacija i polinomi,
<code>datafun</code>	- analiza podataka i Fourier-ova transformacija,
<code>lang</code>	- komande programskog jezika,
<code>timefun</code>	- funkcije datuma i vremena.
<code>graph2d</code>	- dvodimenzionalni grafici,
<code>graph3d</code>	- trodimenzionalni grafici,
<code>graphics</code>	- rukovanje i operacije na graficima.

## MATLAB - priručnik za vježbe

Slijedeći izrazi predstavljaju korisne konstante ili vrijednosti ugrađene u MATLAB.

<code>pi</code>	- $\pi$ , $\pi = 3.1415926\dots$ ,
<code>i</code>	- imaginarna jedinica, $\sqrt{-1}$ ,
<code>j</code>	- isto kao <code>i</code> ,
<code>eps</code>	- relativna preciznost iza decimalne tačke, $\epsilon = 2^{-52} \approx 2.22 \cdot 10^{-16}$ ,
<code>realmin</code>	- najmanji decimalni broj, $2^{-1022} \approx 2.23 \cdot 10^{-308}$ ,
<code>realmax</code>	- najveći realni broj, $(2 - \epsilon) \cdot 2^{1023} \approx 1.80 \cdot 10^{308}$ ,
<code>Inf</code>	- beskonačnost,
<code>NaN</code>	- nebrojčana vrijednost.

Beskonačnost `Inf` se dobija dijeljenjem nekog broja različitog od nule nulom ili ako izračunata vrijednost matematičkog izraza prelazi vrijednost `realmax` (*overflow*). Nebrojčana vrijednost `NaN` se dobija pri pokušaju dijeljenja varijabli koje imaju vrijednosti nula ili beskonačno (`0/0` ili `Inf/Inf`), kao i pri pokušaju oduzimanja varijabli koje imaju vrijednost beskonačno (`Inf-Inf`).

Imena ovih varijabli nisu rezervisana, tj. njihovu vrijednost možemo promijeniti jednostavnom komandom prdruživanja, npr.

```
>> eps=0.000001
```

Originalna vrijednost se vraća komandom

```
>> clear eps
```

## Formiranje matrica

Matrice se mogu unijeti u MATLAB na nekoliko načina: pojedinačnim unošenjem čitave liste elemenata, učitavanjem iz file-ova s podacima i generisanjem pomoću MATLAB funkcija ili posebno napravljenih M-file-ova.

Prilikom direktnog unosa matrice u komandnom prozoru uglasta zagrada označava matricu, elementi redova odvajaju se zarezom ili praznim mjestom, a prelazak u novi red označava se znakom "tačka-zarez", `;`. Unošenjem

```
>> A=[3,4,8,1; 2,13,7,5; 16,12,1,10; 11,6,9,14]
```

u komandnom prozoru MATLAB-a, ispisaće se matrica:

```
A =  
     3     4     8     1  
     2    13     7     5  
    16    12     1    10  
    11     6     9    14
```

Matricu možemo unijeti u radni prostor i komandom `load` na slijedeći način. Napravimo tekst file u nekom editoru van MATLAB-a (npr. Notepad-u) koji sadrži listu brojeva odvojenih praznim mjestima u kojem je u svakoj liniji jednak broj elemenata

## MATLAB - priručnik za vježbe

```
3.0  4.0  8.0  1.0
2.0  13.0 7.0  5.0
16.0 12.0 1.0  10.0
11.0  6.0  9.0  14.0
```

i spasimo ga u neki direktorij koji se nalazi u MATLAB-ovom path-u pod imenom `A.txt`. Komandom

```
>> load A.txt
```

formira se varijabla, odnosno matrica `A` koja sadrži elemente file-a `A.txt`.

Matrice se takođe mogu generisati pomoću MATLAB-ovih funkcija:

<code>zeros</code>	Matrica čiji su svi elementi nule,
<code>ones</code>	Matrica čiji su svi elementi jedinice,
<code>eye</code>	Jedinična matrica,
<code>rand</code>	Uniformno distribuirani slučajni elementi u intervalu $(0.0, 1.0)$ ,
<code>randn</code>	Normalno distribuirani slučajni elementi sa srednjom vrijednošću nula, a varijansom i standardnom devijacijom jedan.

Na primjer:

```
>> B=zeros(3)
```

```
B =
    0    0    0
    0    0    0
    0    0    0
```

```
>> C=3*ones(2,5)
```

```
C =
    3    3    3    3    3
    3    3    3    3    3
```

```
>> D=eye(3)
```

```
D =
    1    0    0
    0    1    0
    0    0    1
```

```
>> E=10*rand(3)
```

```
E =
    4.4470    9.2181    4.0571
    6.1543    7.3821    9.3547
    7.9194    1.7627    9.1690
```

```
>> F=randn(1,6)
```

```
F =
   -0.0592   -1.0106    0.6145    0.5077    1.6924    0.5913
```

Ako želimo cijele brojeve u matricama slučajnih brojeva, za zaokruživanje možemo koristiti komande `round` (zaokružuje na najbližu cjelobrojnu vrijednost) ili `fix` (zaokružuje na vrijednost bližu nuli, odn. briše decimalna mjesta).

```
>> G=round(F)
G =
    0    -1     1     1     2     1

>> H=fix(F)
H =
    0    -1     0     0     1     0
```

Još jedan od načina da se matrica unese u radni prostor MATLABA je korištenje M-file-ova. Napravimo file koji sadrži sljedeće

```
A = [ 3   4   8   1
      2  13   7   5
      16  12   1  10
      11   6   9  14];
```

i spasimo ga pod imenom `matrica_A.m`. Naredbom

```
>> matrica_A
```

izvršavamo M-file, odnosno formiramo matricu A.

Matricu takođe možemo dobiti spajanjem više već postojećih, modifikovanih ili ne, manjih matrica. Npr.

```
>> B=[A A-16; A+16 A*2]
B =
     3     4     8     1    -13    -12     -8    -15
     2    13     7     5    -14     -3     -9    -11
    16    12     1    10     0     -4    -15     -6
    11     6     9    14     -5    -10     -7     -2
    19    20    24    17     6     8    16     2
    18    29    23    21     4    26    14    10
    32    28    17    26    32    24     2    20
    27    22    25    30    22    12    18    28
```

## Osnovne operacije nad matricama

### Aritmetički operatori

MATLAB koristi uobičajenu notaciju i pravila za aritmetičke operacije, ali se osnovne operacije mogu izvršavati i pozivanjem odgovarajućih funkcija Osnovni aritmetički operatori i njima odgovarajuće funkcije su:

Funkcija	Opis funkcije ili operatora	Operator
plus	sabiranje	+
minus	oduzimanje	-
mtimes	matrično množenje	*
times	množenje element po element	.*

## MATLAB - priručnik za vježbe

mpower	matrično stepenovanje	^
power	stepenovanje element po element	.^
mldivide	matrično lijevo dijeljenje	\
mrdivide	matrično desno dijeljenje	/
ldivide	lijevo dijeljenje element po element (2.\4=4./2=2)	.\
rdivide	desno dijeljenje element po element	./

Izraz "plus(A,B)" odgovara izrazu "A+B". Isto vrijedi i za sve ostale operatore. Matrice A i B moraju biti odgovarajuće (u slučaju sabiranja, iste) dimenzije osim ako je jedna od njih skalar. U tom slučaju se nad svakim elementom matrice vrši željena operacija tim skalarom.

```
>> A+2
ans =
     5     6    10     3
     4    15     9     7
    18    14     3    12
    13     8    11    16
```

Matrično množenje, dijeljenje i stepenovanje (operator bez tačke ispred) podrazumjeva izvršenje na osnovu zakonitosti koje vrijede za matrice, dok množenje, dijeljenje i stepenovanje element po element (operator sa tačkom ispred) podrazumjeva izvršenje operacije nad odgovarajućim elementima matrica (koji imaju iste indekse – nalaze se na istim pozicijama).

```
>> A.^2
ans =
     9    16    64     1
     4   169    49    25
   256   144     1   100
   121    36    81   196

>> A^2
ans =
   156   166    69   117
   199   291   159   207
   198   292   303   226
   343   314   265   327
```

### Operator dvotačka

Jedan od najvažnijih operatora u MATLAB-u, koji se ne koristi samo kod matrica je dvotačka, : . Njime dobijamo niz brojeva odnosno cjelobrojnu jediničnu matricu (vektor) u jednom redu između zadatih granica. Npr.

```
>> 1:10
ans =
     1     2     3     4     5     6     7     8     9    10
```

Ako želimo dobiti vektor v, niz brojeva sa nekim drugim korakom, to činimo na slijedeći način:



```
>> V=20:-3:0
V =
    20    17    14    11     8     5     2
```

U narednom dijelu kroz primjere će se vidjeti upotreba i korisnost ovog operatora.

### Indeksi. Unošenje i brisanje elemenata matrice

Nakon učitavanja matrice A u radni prostor MATLAB-a na jedan od prethodno objašnjenih načina, na njoj možemo pokazati neke osnovne funkcije i operacije nad matricama.

Do elementa matrice u i-tom redu i j-toj koloni koloni dolazimo jednostavnim kucanjem komande  $A(i, j)$ , tako npr.

```
>> A(2,3)
ans =
     7
```

dobijamo element u drugom redu i trećoj koloni matrice A. Do pojedinih elemenata matrice možemo doći koristeći samo jedan indeks. U tom slučaju matrica se smatra kao jedna velika kolona formirana redom od kolona originalne matrice. Za našu matricu A element  $A(10)$  je u stvari element  $A(2, 3)$ .

Ako pokušamo dobiti vrijednost elementa koji je van dimenzija matrice, npr. elementa  $A(4,5)$ , MATLAB će javiti grešku. Međutim, ako pohranimo neku vrijednost u element koji je van matrice, matrica mijenja dimenzije da obuhvati taj element.

```
>> X=A;
>> X(1,5)=17
X =
     3     4     8     1    17
     2    13     7     5     0
    16    12     1    10     0
    11     6     9    14     0
```

Do jednog segmenta matrice možemo doći upotrebom operatora  $:$ . Ako želimo ispisati ili pridružiti nekoj varijabli sve elemente jedne kolone, to činimo na slijedeći način:

```
>> A(:,3)
ans =
     8
     7
     1
     9
```

Vidimo da kada se operator dvotačka nađe na mjestu jednog od indeksa koji označavaju red ili kolonu, znači da su obuhvaćeni svi elementi reda odnosno kolone. U slučaju da su date granice, onda se naredba odnosi samo na elemente obuhvaćene tim granicama, kao na primjer:

```
>> A(2:3,2:4)
ans =
    13     7     5
    12     1    10
```

Kada nisu poznate dimenzije matrice najlakši način da dodjemo do posljednjeg reda ili kolone, ili reda i kolone u odnosu na posljednji, je korištenjem ključne riječi `end`.

```
>> A(end,1:end-1)
ans =
    11     6     9
```

Brisanje redova ili kolona iz matrice vrši se praznim uglastim zagradama:

```
>> X=A;
>> X(3,:)=[]
X =
     3     4     8     1
     2    13     7     5
    11     6     9    14
```

Ovako smo izbrisali treći red matrice `x` koja je na početku bila jednaka matrici `A`. Ako izbrišemo samo jedan ili više elemenata matrice koji se ne nalaze u istim redovima ili kolonama, rezultat više nije matrica nego linijski vektor (jednodimenzionalna matrica s jednim redom). Ako pokušamo sada matrici `x` izbrisati samo jedan element, recimo u drugom redu i drugoj koloni, koristeći oba indeksa

```
>> X(2,2)=[]
```

MATLAB javlja grešku. Međutim, ako koristimo samo jedan indeks tada možemo izbrisati odgovarajući element

```
>> X(5)=[]
X =
     3     2    11     4     6     8     7     9     1     5    14
```

Kao što vidimo, rezultat više nije matrica nego linijski vektor (jednodimenzionalna matrica s jednim redom) koji se sastoji od ostalih elemenata matrice. Pri brisanju pojedinih elemenata takođe možemo koristiti operator dvotačka

```
>> X=A;
>> X(3:2:11)=[]
X =
     3     2    11    13     6     7     9     1     5    10    14
```

Ovim smo izbrisali svaki drugi element počevši od trećeg zaključno s jedanaestim.

Osnovne informacije o dimenzijama matrice možemo dobiti komandama `size` i `length`. Rezultat izvršenja komande `size` je linijski vektor čiji su elementi dimenzije matrice (broj redova i broj kolona za dvodimenzionalnu matricu), a rezultat izvršenja komande `length` je najveća dimenzija matrice

```
>> size(X)
ans =
     1     11
>> length(X)
ans =
     11
```

## Manipulacija matricama

Transponovanu matricu, odnosno matricu čiji su redovi i kolone zamijenili mjesta, dobijamo korištenjem operatora prim, '.

```
>> AT=A'
AT =
     3     2    16    11
     4    13    12     6
     8     7     1     9
     1     5    10    14
```

Komandom `fliplr` dobijamo matricu kojoj su kolone zamijenile redoslijed, pa je:

```
>> fliplr(A)
ans =
     1     8     4     3
     5     7    13     2
    10     1    12    16
    14     9     6    11
```

Slično ovome, komandom `flipud` dobijamo matricu sa zamijenjenim redoslijedom redova.

Matricu zarotiranu za  $90^\circ$  u pozitivnom matematičkom smijeru (suprotno od kazaljke na satu) dobijamo komandom `rot90`

```
>> rot90(A)
ans =
     1     5    10    14
     8     7     1     9
     4    13    12     6
     3     2    16    11
```

Do elemenata matrice po glavnoj dijagonali dolazimo naredbom `diag`:

```
>> diag(A)
ans =
     3
    13
     1
    14
```

Determinantu matrice dobijamo naredbom `det`

```
>> det(A)
```

```
ans =
    -12353
```

## Osnovne operacije

Zbir elemenata po kolonama, a ne zbir svih elemenata matrice, dobijamo naredbom `sum`:

```
>> sum(A)
ans =
    32    35    25    30
```

Može se primjetiti da naredba `sum` daje elemente u jednom redu, dok naredba `diag` u jednoj koloni. Kod nekih proračuna vrlo je bitno znati da li su elementi matrice zapisani u redu ili koloni, npr. kod množenja matrica.

Na sličan način radi i naredba `prod` koja daje proizvod elemenata matrice po kolonama:

```
>> prod(A)
ans =
    1056    3744    504    700
```

Najveći i najmanji element matrice po kolonama dobijamo naredbama `max` i `min`

```
>> max(A)
ans =
    16    13     9    14
```

dok naredbom `mean` dobijamo srednju vrijednost elemenata po kolonama

```
>> mean(A)
ans =
    8.0000    8.7500    6.2500    7.5000
```

Ukoliko je matrica jednodimenzionalna, odnosno vektor, tada se sve prethodne komande vrše nad tom jedinom postojećom dimenzijom (redom ili kolonom). Tako na primjer, ako želimo naći sumu, proizvod, srednju vrijednost svih elemenata matrice  $A$ , ili njen najveći element, to dobijamo dvostrukom odgovarajućom komandom

```
>> sum(sum(A))
ans =
    122
```

Naredbom `sort` sortiramo (redamo po veličini) elemente po kolonama

```
>> sort(A)
ans =
     2     4     1     1
     3     6     7     5
    11    12     8    10
    16    13     9    14
```

Naredbom `inv` dobija se inverzna matrica matrice A

```
>> inv(A)
ans =
    0.0999    -0.0863     0.0752    -0.0300
   -0.0231     0.0815     0.0256    -0.0457
    0.1171    -0.0136    -0.0351     0.0215
   -0.1439     0.0416    -0.0475     0.1008
```

Tako npr. ukoliko želimo riješiti sistem jednačina

$$\begin{aligned} 3x_1 + 4x_2 + 2x_3 &= 13 \\ 2x_1 + x_2 - x_3 &= 6 \\ x_1 + x_2 + 5x_3 &= -1 \end{aligned}$$

predstavili bi ga u matricnom obliku  $a \cdot x = b$ , i izvršenjem sljedećeg niza naredbi dobja se željeno rješenje

```
>> a=[3,4,2;2,1,-1;1,1,5];
>> b=[13;6;-1];
>> x=inv(a)*b
x =
    1.0000
    3.0000
   -1.0000
```

iz kojeg vidimo da je  $x_1=1$ ,  $x_2=3$  i  $x_3=-1$ .

## Grafika

MATLAB je takođe i vrlo moćan alat za vizualizaciju odn. grafičko predstavljanje određenih vrijednosti. Jednostavnom komandom

```
>> plot(A)
```

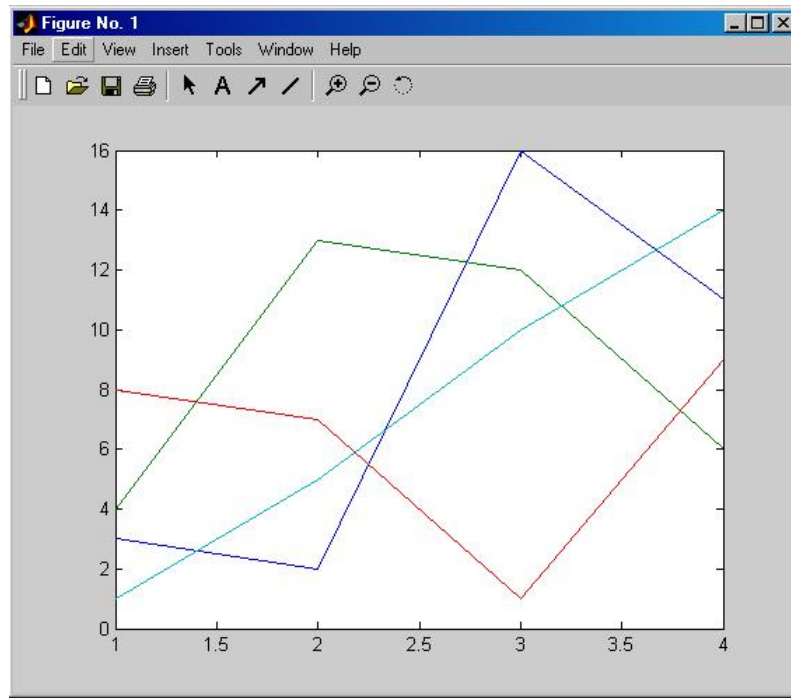
dobija se prozor za slike na kojem su različitim linijama predstavljene kolone matrice A u zavisnosti od pozicije u koloni matrice A (1, 2, 3 odn. 4), kao što je predstavljeno na slici 10. Na već gotovoj slici moguće je ubacivati odgovarajuće detalje, poput naziva slike, naziva osa, legende, itd, pomoću padajućih menija u prozoru za slike. Takođe je moguće sve te detalje, kao i boju i vrstu linije, te oblik tačke, definisati odgovarajućim komandama, a više detalja o tome dobija se izvršenjem naredbe

```
>> help plot
```

To je naročito pogodno za formatiranje slika dobijenih pomoću naredbi u okviru `m-file`-ova. **Te naredbe, kao i slične naredbe za crtanje (npr. logaritamski grafici) samostalno preraditi!**

Ukoliko želimo grafički predstaviti neku funkciju, npr  $y = x^3 - 2x^2 - 5x + 6$  na određenom intervalu, to činimo na sljedeći način

```
>> x=-3:0.01:4;
>> y=x.^3-2*x.^2-5.*x+6;
>> plot(x,y)
```



**Slika 10.** Prozor za slike

Prvom naredbom smo dobili vektor (jednodimenzionalnu matricu)  $x$  u intervalu od -3 do 4 s korakom 0.01, dakle, 701 vrijednost. Drugom naredbom za svaku od tih vrijednosti  $x$  dobijamo vrijednost varijable  $y$  (obratiti pažnju na operator  $.$ ^), dok trećom naredbom dobijamo grafik te funkcije na zadanom intervalu tako što MATLAB u prozor za sliku ucrtaj svaku odgovarajuću tačku, i međusobno ih spoji. Zbog međusobne blizine tačaka ti prelazi se ne vide, dok ukoliko bi taj interval, odn varijablu  $x$  formirali sa većim korakom, npr. 0.5, prelazi bi bili vidljivi.

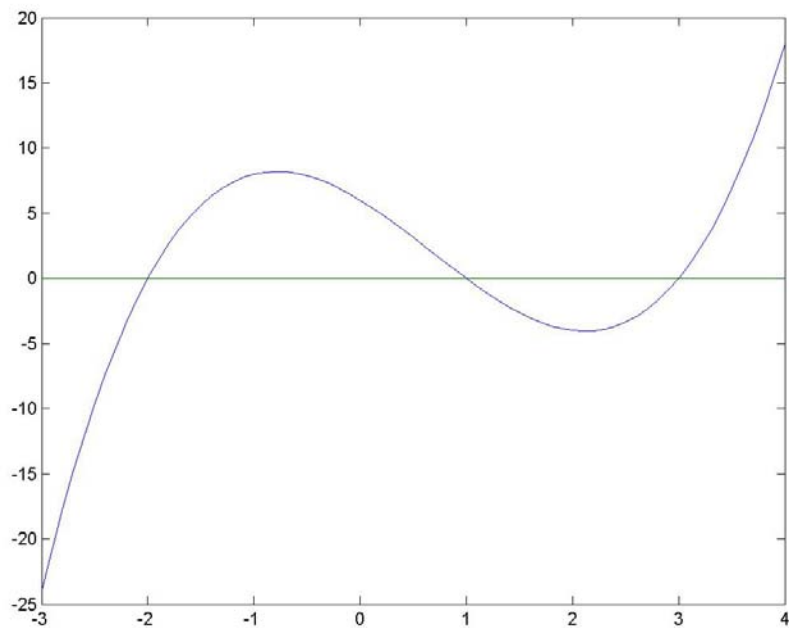
Koruštenjem naredbe `plot` u prozoru za slike briše se prethodni crtež, i ubacuje novi. Ukoliko zelimo u jednom prozoru imati više funkcija, to mozemo dobiti ukucavanjem naredbe `hold on`, čime se svaka iduća naredba `plot` ostavlja u istom prozoru, sve dok se u komandnom prozoru ne ukuca naredba `hold off`.

Drugi način je da se u okviru iste `plot` naredbe unutar zagrade dodaju  $x$  i  $y$  vrijednosti funkcija koje želimo da su u istom prozoru (`plot(x,y,x2,y2,...)`). Npr, ako želimo pored grafika gore navedenog polinoma da ucrtamo i funkciju  $y=0$ , to možemo uraditi sljedećom naredbom

```
>> plot(x,y,x,zeros(1,length(x)))
```

Ovim smo dobili grafik polinoma, kao i funkciju  $y=0$ . (za svaku vrijednost promjenljive  $x$ ), sto je prikazano na slici 11.

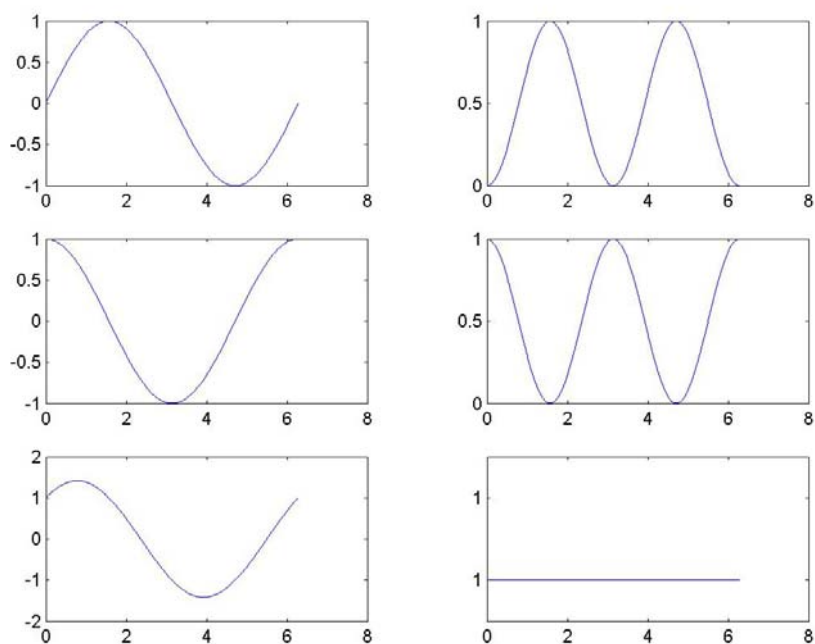
Crtanje više grafika u jednom prozoru postiže se naredbom `subplot(m,n,p)`, kod koje su  $m$  i  $n$  broj redova odn. kolona u kojima se nalaze grafici ( $m \cdot n$  je broj grafika u jednom prozoru), a  $p$  poprima vrijednosti od 1 do  $m \cdot n$ , i određuje odgovarajući grafik funkcije. Ova naredba radi u kombinaciji sa naredbom `plot`, i sva pravila koja vrijede pojedinačno za tu naredbu, vrijede i u okviru naredbe `subplot`, za svaki grafik posebno. Tako npr, ako želimo nacrtati više trigonometrijskih funkcija unutar jednog prozora, to činimo na sljedeći način:



**Slika 11.** Grafik zadane funkcije

```
>> x=0:0.01:2*pi;
>> subplot(3,2,1), plot(x,sin(x))
>> subplot(3,2,2), plot(x,sin(x).^2)
>> subplot(3,2,3), plot(x,cos(x))
>> subplot(3,2,4), plot(x,cos(x).^2)
>> subplot(3,2,5), plot(x,sin(x)+cos(x))
>> subplot(3,2,6), plot(x,sin(x).^2+cos(x).^2)
```

Rezultat izvršenja ovog niza naredbi dat je na slici 12.



**Slika 12.** Grafici određenih funkcija dobijeni naredbom subplot.

## Neke operacije nad vektorima (1-D matricama) i polinomima

Vratimo se polinomu  $y = x^3 + 2x^2 - 5x + 6$ , grafički prikazanom na slici 11. Korjene (nule) ovog polinoma dobijamo kada ga izjednačimo s nulom, i nađemo rješenja. To izvodimo naredbom `roots`, koja operiše nad vektorom, na sljedeći način

```
>> a=[1,-2,-5,6];
>> nule=roots(a)
```

ili

```
>> nule=roots([1,-2,-5,6])
nule =
   -2.0000
    3.0000
    1.0000
```

Dakle, funkcija  $y = x^3 + 2x^2 - 5x + 6$  siječe y-osu u tačkama  $x = -2, 1$  i  $3$ , odn. to su rješenja jednačine  $x^3 + 2x^2 - 5x + 6 = 0$ , i pridružena su varijabli `nule`.

Ukoliko poznajemo nule nekog polinoma, naredbom `poly` dobijamo njegove koeficijente, odn.

```
>> a=poly(nule)
```

ili

```
>> a=poly([-2,1,3])
a =
    1    -2    -5     6
```

Dakle, ukoliko želimo dobiti koeficijente polinoma koji ima nule u tačkama  $x = 1, 3, 5$  i  $7$  to dobijamo izvršenjem naredbe

```
>> poly([1 3 5 7])
ans =
    1   -16    86  -176   105
```

odn. to je polinom  $y = x^4 - 16x^3 + 86x^2 - 176x + 105$ .

Naredbom `polyfit(x,y,n)` određujemo koeficijente polinoma n-tog stepena koji najbolje aproksimira diskretne podatke (tačke) sa  $x$  i  $y$  koordinatama sadržanim u varijablama  $x$  i  $y$ . Izvršenjem te naredbe nad  $x$  i  $y$  vrijednostima koje već imamo u radnom prostoru MATLAB-a dobijamo:

```
>> polyfit(x,y,3)
ans =
    1.0000   -2.0000   -5.0000    6.0000
```

što u stvari i jesu koeficijenti polinoma  $y$  (pogledati kako je dobijeno  $y$ !).



Izvršenjem naredbe

```
>> polyfit(x,y,4)
ans =
    -0.0000    1.0000   -2.0000   -5.0000    6.0000
```

trebali bi dobiti polinom četvrtog reda, ali je prvi koeficijent nula, dakle opet je dobijen polinom trećeg reda, upravo iz razloga jer je y i zadato kao polinom trećeg reda. Izvršenjem naredbe

```
>> polyfit(x,y,2)
ans =
    -0.5000    1.8080    2.3460
```

dobijamo koeficijente polinoma drugog reda koji najbolje aproksimira polinom y na intervalu vrijednosti x-a metodom najmanjih kvadrata, odn to je  $y_2 = -0.5x^2 + 1.808x + 2.34$ .

Ukoliko bi željeli odrediti koeficijente polinoma koji prolazi kroz tačke (1,1), (2,4), (4,2) i (7,0) to bi uradili na sljedeći način:

```
>> xp=[1 2 4 7];
>> yp=[1 4 2 0];
>> polyfit(xp,yp,3)
ans =
    0.2333   -2.9667   10.2667   -6.5333
```

Za vježbu nacrtati ovaj polinom, diskretne tačke kroz koje on prolazi, kao i njegovu aproksimaciju polinomom drugog reda na istom grafiku. Provjeriti šta se dešava u slučaju da želimo aproksimaciju polinomom četvrtog i višeg reda.

Naredbom `trapz` dobija se numerička vrijednost integrala na određenom domenu. Tako npr. nizom naredbi

```
>> x=-3:0.1:4;
>> y=x.^3-2*x.^2-5.*x+6;
>> trapz(x,y)
ans =
    7.5775
```

dobijamo vrijednost određenog integrala funkcije y na intervalu (-3,4). Analitičko rješenje ovog integrala je 7.5833. Ova razlika u vrijednostima je nastala zbog numeričkog načina integracije trapezoidnim pravilom. U slučaju da se varijabla x zada sa manjim korakom, greška integracije se smanjuje i vrijednost se približava analitičkoj (provjeriti!). Ukoliko želimo vrijednost integrala na nekom drugom intervalu, potrebno je prvo izmijeniti taj interval, izračunati vrijednosti funkcije za njega, te integrirati, na sljedeći način

```
>> x=-2:0.1:1;
>> y=x.^3-2*x.^2-5.*x+6;
>> trapz(x,y) % vrijednost integrala na intervalu (-2,1)
ans =
    15.7325
```

## MATLAB PROGRAMSKI JEZIK

MATLAB sadrži u sebi i klasične naredbe programskih jezika čime se omogućuje efikasnija obrada i manipulacija podacima u okviru m-file-ova. U ovom poglavlju date su najosnovnije postavke s nekim primjerima.

### if - naredba

if - naredba ima sljedeću strukturu

```
if izraz1
    naredbe1
elseif izraz2
    naredbe2
elseif ...
.
.
else
    naredbeN
end
```

naredbe1 će se izvršiti ako izraz1 ima vrijednost različitu od nule (odnosno sve realne vrijednosti veće od nule, ukoliko se radi o matrici), i odmah se izlazi iz if - naredbe, što znači da bez obzira i ako je izraz2 različit od nule, naredbe2, niti bilo koji ostali skup naredbi iza elseif/else naredbi. naredbeN će se izvršiti samo ako su svi prethodni izrazi jednaki nuli, tj. ako se niti jedan prethodni skup naredbi ne izvrši. U okviru if - naredbe može biti proizvoljan broj elseif naredbi i samo jedna else naredba.

Izrazi u okviru if - naredbe su obično relacioni operatori u kombinaciji sa operatorima matematičke logike, i poprimaju vrijednost jedan, ako je izraz tačan, odnosno nula ako je netačan. Ti operatori su:

Naziv	Funkcija	Simb. oznaka	Opis
<b>Relacioni operatori</b>			
jednakost	eq(A,B)	A==B	Operiraju nad matricama A i B, element po element (A i B moraju biti istih dimenzija), a rezultat izvrš. je 1, ukoliko je izraz tačan odn. 0 ukoliko je netačan.
nejednakost	ne(A,B)	A~=B	
manje od	lt(A,B)	A<B	
veće od	gt(A,B)	A>B	
manje ili jednako	le(A,B)	A<=B	
veće ili jednako	ge(A,B)	A>=B	
<b>Logički operatori</b>			
logičko "i"		A&&B	Operiraju nad skalarima
logičko "ili"		A B	
logičko "i"	and(A,B)	A&B	Operiraju nad matricama, element po element
logičko "ili"	or(A,B)	A B	
logičko "ne"	not(A)	~A	
logičko "ekskl. ili"	xor(A,B)		
logičko "bilo kojii"	any(A)	Daje vrijednost 1 ukoliko je bilo koji elemnat	
logičko "svi"	all(A)	vektora A (any) odn. svi (all) različiti od nule.	

Upotreba if - naredbe vidi se sa primjera.

## MATLAB - priručnik za vježbe

Ukoliko trebamo napraviti m-file funkciju koja će za unesene vrijednosti koeficijenta  $a$ ,  $b$  i  $c$  riješiti kvadratnu jednačinu  $ax^2 + bx + c = 0$  ukoliko ima realna rješenja, odn. ispisati poruku na ekranu ukoliko su rješenja kompleksna, to možemo uraditi na sljedeći način:

```
function rj=kvadratna_real(a,b,c)
% Realna rjesenja kvadratne jednacine ax^2+bx+c=0
if length(a)>1||length(b)>1||length(c)>1
    rj='Koeficijenti kvadr. jedn. moraju biti skalarne vrijedn.!';
else
    D=b^2-4*a*c;
    if D >= 0
        rj(1,1)=(-b-sqrt(D))/(2*a);
        rj(2,1)=(-b+sqrt(D))/(2*a);
    else
        rj='Kvadr. jedn. zadatih parametara nema realnih rjes.!';
    end
end
end
```

M - file ovakvog sadržaja treba spasiti pod imenom kvadratna\_real.m u radni direktorij, i njegovim pozivanjem dobijamo odgovarajući rezultat, npr:

```
>> kvadratna_real(1,2,-3)
ans =
    -3
     1
ili
>> kvadratna_real(1,2,3)
ans =
Kvadr. jedn. zadatih parametara nema realnih rjes.!
```

## for - petlja

Ukoliko postoji potreba da se određeni skup naredbi izvrši više puta (naročito kada je tačno poznat broj potrebnih izvršavanja) koristi se for - petlja. Sintaksa for - petlje je

```
for brojac = pocetak : korak : kraj
    naredbe
end
```

Naredbe u okviru ove petlje će se izvršiti onoliko puta koliko je potrebno da brojač dođe od vrijednosti `pocetak` do vrijednosti `kraj` sa vrijednošću koraka `korak`. Ukoliko se korak izostavi, podrazumjeva se da ima vrijednost jedan. Jedna ili više naredbi unutar petlje najčešće imaju neku zavisnost od brojača. U okviru jedne petlje moguće je da se nalazi još jedna ili više petlji, što omogućava višestruko variranje brojača i naročito je pogodno za manipulaciju elementima matrice. U tom slučaju, brojač vanjske for - petlje prelazi na sljedeću vrijednost tek nakon što brojač unutrašnje petlje poprimi sve vrijednosti. To se najbolje može ilustrovati sljedećim primjerom. Napravite m - file pod imenom `demonstracija_for.m` sljedećeg sadržaja i spasite ga u radni direktorij

## MATLAB - priručnik za vježbe

```
broj_izvršenja=0;
for i=1:3
    for j=1:0.5:3
        broj_izvršenja=broj_izvršenja+1;
        sprintf('Broj izvršenja naredbi u okviru petlje je
                %d, a brojac i=%d, j=%d',broj_izvršenja,i,j)
    end
end
```

Njegovo izvršavanje ilustruje promjenu brojača i odn. j, kao i ukupan broj izvršenja naredbi unutar petlji.

Ukoliko trebamo formirati matricu dimenzija (n x n) čiji elementi su neparni brojevi od 1 do  $2n^2-1$  to bi mogli učiniti na sljedeći način pomoću m - file-a

```
n= input ('Unesite dimenziju kvadratne matrice, n = ');
neparni=-1;
for j=1:n
    for i=1:n
        neparni=neparni+2;
        A(i,j)=neparni;
    end
end
A
```

Ukoliko želimo formirati matricu A dimenzije (m x n) slučajno odabranih prirodnih brojeva između 0 i 50, te formirati matricu B koja će imati elemente jednake elementima matrice A ukoliko su veći ili jednaki 25, a elemente vrijednosti 0 na pozicijama gdje su elementi matrice A manji od 25, i matricu C koja ima elemente vrijednosti 1 na pozicijama gdje su elementi matrice A veći ili jednaki od 25, odn. vrijednosti 0 na pozicijama gdje su elementi matrice A manji od 25, to možemo dobiti ukoliko napravimo m-file idućeg sadržaja:

```
m=input('Unesite broj redova matrice A, m = ');
n=input('Unesite broj kolona matrice A, n = ');
A=round(50*(rand(m,n)))
for i=1:m
    for j=1:n
        if A(i,j)>=25
            B(i,j)=A(i,j);
        else
            B(i,j)=0;
        end
    end
end
C=A>25
```

U MATLAB-u postoje još i while petlje, te switch - case odn. try - catch blokovi naredbi koje nećemo detaljnije opisivati. Naredbom `continue` prekidamo izvršenje naredbi unutar petlje i prebacujemo se na iduću vrijednost brojača, dok naredbom `break` potpuno prekidamo izvršenje naredbi unutar petlje i izlazimo iz nje.

## Primjeri upotrebe

### PRIMJER 1. M-file koji crta polinom kroz unesene tačke

```
clear
disp('Unosenje koordinata tacaka')
test='d';
i=1;
while test=='d' || test=='D'
    disp('-----')
    disp(['Unesite koord. tacke ',int2str(i),':'])
    x(i)=input('x-koordinata: ');
    y(i)=input('y-koordinata: ');
    i=i+1;
    test=input('Unesite D za novu tacku (za prekid bilo koji drugi
                znak): ','s');
end
nx=length(x);
a=polyfit(x,y,nx-1);
na=length(a)
x2=min(x):(max(x)-min(x))/1000:max(x);
nx2=length(x2);
for j=1:nx2
    y2(j)=0.;
    for i=1:na
        y2(j)=y2(j)+a(i)*x2(j)^(na-i);
    end
end
plot(x,y,'o',x2,y2)
```

### Primjer 2. Animacija kosog hica

```
clear
v0=input('Unesite pocetnu brzinu, v0= ');
alfa=input('Unesite ugao ispaljenja u stepenima, alfa= ');
t=input('Unesite duzinu trajanja animacije, t= ');
alfarad=alfa*pi/180; g=9.81;
domet=(v0*cos(alfarad)*t);
xe=1.05*domet;
xs=-0.05*domet;
xx=[0,xe];
yy=[0,0];
ys=max(0,sign(alfarad)*(v0*sin(alfarad))^2/(2*g));
ye=min(0,v0*t*sin(alfarad)-g*t^2/2);
margin=0.05*abs(ys-ye);
dt=t/100;
h=plot(0.,0.,'.',xx,yy);
hold on;
axis equal;
axis([xs xe min(ys, ye)-margin max(ys, ye)+margin]);
for i=1:100
    pause(dt);
    t=i*dt;
    x=v0*cos(alfarad)*t;
    y=v0*t*sin(alfarad)-g*t^2/2;
    plot(x,y,'.');
end
```

## LISTA KORISNIH NAREDBI

>> **help general**

General purpose commands.  
MATLAB Version 6.5 (R13) 20-Jun-2002

General information

helpbrowser - Bring up the help browser.  
help - M-file help, displayed at the command line.  
helpwin - M-file help, displayed in the help browser.

Managing the workspace.

who - List current variables.  
whos - List current variables, long form.  
clear - Clear variables and functions from memory.  
load - Load workspace variables from disk.

Managing commands and functions.

type - List M-file.  
open - Open files by extension.

>> **help elfun**

Elementary math functions.

Trigonometric.

sin - Sine.  
sinh - Hyperbolic sine.  
asin - Inverse sine.  
asinh - Inverse hyperbolic sine.  
cos - Cosine.  
cosh - Hyperbolic cosine.  
acos - Inverse cosine.  
acosh - Inverse hyperbolic cosine.  
tan - Tangent.  
tanh - Hyperbolic tangent.  
atan - Inverse tangent.  
atan2 - Four quadrant inverse tangent.  
atanh - Inverse hyperbolic tangent.  
sec - Secant.  
sech - Hyperbolic secant.  
asec - Inverse secant.  
asech - Inverse hyperbolic secant.  
csc - Cosecant.  
csch - Hyperbolic cosecant.  
acsc - Inverse cosecant.  
acsch - Inverse hyperbolic cosecant.  
cot - Cotangent.  
coth - Hyperbolic cotangent.  
acot - Inverse cotangent.  
acoth - Inverse hyperbolic cotangent.

Exponential.

exp - Exponential.  
log - Natural logarithm.  
log10 - Common (base 10) logarithm.  
log2 - Base 2 logarithm and dissect floating point number.  
pow2 - Base 2 power and scale floating point number.  
realpow - Power that will error out on complex result.  
reallog - Natural logarithm of real number.  
realsqrt - Square root of number greater than or equal to zero.  
sqrt - Square root.  
nextpow2 - Next higher power of 2.

## MATLAB - priručnik za vježbe

### Complex.

abs	- Absolute value.
complex	- Construct complex data from real and imaginary parts.
conj	- Complex conjugate.
imag	- Complex imaginary part.
real	- Complex real part.
isreal	- True for real array.

### Rounding and remainder.

fix	- Round towards zero.
floor	- Round towards minus infinity.
ceil	- Round towards plus infinity.
round	- Round towards nearest integer.
mod	- Modulus (signed remainder after division).
rem	- Remainder after division.
sign	- Signum.

### >> help specfun

cross	- Vector cross product.
dot	- Vector dot product.

### Number theoretic functions.

factor	- Prime factors.
isprime	- True for prime numbers.
primes	- Generate list of prime numbers.
gcd	- Greatest common divisor.
lcm	- Least common multiple.
rat	- Rational approximation.
rats	- Rational output.
perms	- All possible permutations.
nchoosek	- All combinations of N elements taken K at a time.
factorial	- Factorial function.

### Coordinate transforms.

cart2sph	- Transform Cartesian to spherical coordinates.
cart2pol	- Transform Cartesian to polar coordinates.
pol2cart	- Transform polar to Cartesian coordinates.
sph2cart	- Transform spherical to Cartesian coordinates.

### >> help elmat

#### Elementary matrices and matrix manipulation.

##### Elementary matrices.

zeros	- Zeros array.
ones	- Ones array.
eye	- Identity matrix.
rand	- Uniformly distributed random numbers.
randn	- Normally distributed random numbers.
linspace	- Linearly spaced vector.
logspace	- Logarithmically spaced vector.
meshgrid	- X and Y arrays for 3-D plots.
:	- Regularly spaced vector and index into matrix.

##### Basic array information.

size	- Size of array.
length	- Length of vector.
ndims	- Number of dimensions.
disp	- Display matrix or text.

##### Matrix manipulation.

reshape	- Change size.
diag	- Diagonal matrices and diagonals of matrix.
fliplr	- Flip matrix in left/right direction.

## MATLAB - priručnik za vježbe

flipud - Flip matrix in up/down direction.  
rot90 - Rotate matrix 90 degrees.  
: - Regularly spaced vector and index into matrix.  
find - Find indices of nonzero elements.

Multi-dimensional array functions.

Special variables and constants.

ans - Most recent answer.  
eps - Floating point relative accuracy.  
realmax - Largest positive floating point number.  
realmin - Smallest positive floating point number.  
pi - 3.1415926535897....  
i, j - Imaginary unit.  
inf - Infinity.  
NaN - Not-a-Number.  
isnan - True for Not-a-Number.  
isinf - True for infinite elements.  
isfinite - True for finite elements.

Specialized matrices.

### >> help funfun

Function functions and ODE solvers.

Optimization and root finding.

fzero - Scalar nonlinear zero finding.

Numerical integration (quadrature).

quadl - Numerically evaluate integral, higher order method.  
dblquad - Numerically evaluate double integral.  
triplequad - Numerically evaluate triple integral.

Plotting.

ezplot - Easy to use function plotter.  
ezplot3 - Easy to use 3-D parametric curve plotter.  
ezpolar - Easy to use polar coordinate plotter.  
ezcontour - Easy to use contour plotter.  
ezcontourf - Easy to use filled contour plotter.  
ezmesh - Easy to use 3-D mesh plotter.  
ezmeshc - Easy to use combination mesh/contour plotter.  
ezsurf - Easy to use 3-D colored surface plotter.  
ezsurf c - Easy to use combination surf/contour plotter.  
fplot - Plot function.

Inline function object.

inline - Construct INLINE function object.

### >> help datafun

Data analysis and Fourier transforms.

Basic operations.

max - Largest component.  
min - Smallest component.  
mean - Average or mean value.  
sort - Sort in ascending order.  
sum - Sum of elements.  
prod - Product of elements.  
trapz - Trapezoidal numerical integration.  
diff - Difference and approximate derivative.  
gradient - Approximate gradient.  
char - Create character array (string).



## MATLAB - priručnik za vježbe

### >> help graph2d

Two dimensional graphs.

Elementary X-Y graphs.

plot - Linear plot.  
loglog - Log-log scale plot.  
semilogx - Semi-log scale plot.  
semilogy - Semi-log scale plot.  
polar - Polar coordinate plot.  
plotyy - Graphs with y tick labels on the left and right.

Axis control.

axis - Control axis scaling and appearance.  
zoom - Zoom in and out on a 2-D plot.  
grid - Grid lines.  
box - Axis box.  
hold - Hold current graph.  
axes - Create axes in arbitrary positions.  
subplot - Create axes in tiled positions.

Graph annotation.

plotedit - Tools for editing and annotating plots.  
legend - Graph legend.  
title - Graph title.  
xlabel - X-axis label.  
ylabel - Y-axis label.  
texlabel - Produces TeX format from a character string  
text - Text annotation.

### >> help graph3d

Three dimensional graphs.

Elementary 3-D plots.

plot3 - Plot lines and points in 3-D space.  
mesh - 3-D mesh surface.  
surf - 3-D colored surface.

### >> help graphics

Handle Graphics.

Figure window creation and control.

figure - Create figure window.  
clf - Clear current figure.

Axis creation and control.

subplot - Create axes in tiled positions.  
axes - Create axes in arbitrary positions.  
gca - Get handle to current axes.  
cla - Clear current axes.  
axis - Control axis scaling and appearance.

### >> help lang

Programming language constructs.

Control flow.

if - Conditionally execute statements.  
else - IF statement condition.  
elseif - IF statement condition.  
end - Terminate scope of FOR, WHILE, SWITCH, TRY and IF statements.  
for - Repeat statements a specific number of times.

## MATLAB - priručnik za vježbe

while	- Repeat statements an indefinite number of times.
break	- Terminate execution of WHILE or FOR loop.
continue	- Pass control to the next iteration of FOR or WHILE loop.
switch	- Switch among several cases based on expression.
case	- SWITCH statement case.
otherwise	- Default SWITCH statement case.
try	- Begin TRY block.
catch	- Begin CATCH block.
return	- Return to invoking function.
error	- Display error message and abort function.

### >> help matfun

Matrix functions - numerical linear algebra.

Matrix analysis.

norm	- Matrix or vector norm.
rank	- Matrix rank.
det	- Determinant.
trace	- Sum of diagonal elements.

### >> help ops

Operators and special characters.

Arithmetic operators.

plus	- Plus	+
uplus	- Unary plus	+
minus	- Minus	-
uminus	- Unary minus	-
mtimes	- Matrix multiply	*
times	- Array multiply	.*
mpower	- Matrix power	^
power	- Array power	.^
mldivide	- Backslash or left matrix divide	\
mrdivide	- Slash or right matrix divide	/
ldivide	- Left array divide	.\
rdivide	- Right array divide	./
kron	- Kronecker tensor product	kron

Relational operators.

eq	- Equal	==
ne	- Not equal	~=
lt	- Less than	<
gt	- Greater than	>
le	- Less than or equal	<=
ge	- Greater than or equal	>=

Logical operators.

	Short-circuit logical AND	&&
	Short-circuit logical OR	
and	- Element-wise logical AND	&
or	- Element-wise logical OR	
not	- Logical NOT	~
xor	- Logical EXCLUSIVE OR	
any	- True if any element of vector is nonzero	
all	- True if all elements of vector are nonzero	

Special characters.

colon	- Colon	:
paren	- Parentheses and subscripting	( )
paren	- Brackets	[ ]
paren	- Braces and subscripting	{ }

## MATLAB - priručnik za vježbe

punct	- Function handle creation	@
punct	- Decimal point	.
punct	- Structure field access	.
punct	- Parent directory	..
punct	- Continuation	...
punct	- Separator	,
punct	- Semicolon	;
punct	- Comment	%
punct	- Invoke operating system command	!
punct	- Assignment	=
punct	- Quote	'
transpose	- Transpose	.'
ctranspose	- Complex conjugate transpose	'
horzcat	- Horizontal concatenation	[,]
vertcat	- Vertical concatenation	[;]
subsasgn	- Subscripted assignment	( ), { }, ..
subsref	- Subscripted reference	( ), { }, ..
subsindex	- Subscript index	

### >> help polyfun

#### Polynomials.

roots	- Find polynomial roots.
poly	- Convert roots to polynomial.
polyfit	- Fit polynomial to data.
polyder	- Differentiate polynomial.
polyint	- Integrate polynomial analytically.

### >> help iofun

#### Formatted file I/O.

fgetl	- Read line from file, discard newline character.
fgets	- Read line from file, keep newline character.
fprintf	- Write formatted data to file.
fscanf	- Read formatted data from file.
input	- Prompt for user input.
textread	- Read formatted data from text file.

#### String conversion.

sprintf	- Write formatted data to string.
sscanf	- Read string under format control.
strread	- Read formatted data from text string.

#### File opening and closing.

fopen	- Open file.
fclose	- Close file.

#### Binary file I/O.

fread	- Read binary data from file.
fwrite	- Write binary data to file.

### >> help timefun

#### Time and dates.

##### Current date and time.

now	- Current date and time as date number.
date	- Current date as date string.
clock	- Current date and time as date vector.

#### Date functions.

calendar	- Calendar.
----------	-------------