

**SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

OSNOVE MATLABA

doc. dr. sc. Željko Ban

Zagreb, 2003.

SADRŽAJ

1.	UVOD	1
2.	POKRETANJE I ORGANIZACIJA MATLABA	2
2.1.	Pokretanje Matlaba	2
2.2.	Organizacija Matlaba i strukture podataka	5
3.	VARIJABLE	6
3.1.	Interne varijable	7
3.2.	Eksterne varijable	8
3.3.	Definiranje varijable	8
3.3.1.	Vektori i matrice	8
3.3.2.	Definiranje niza brojeva	10
3.3.3.	Ekstrakcija dijela matrice	11
3.3.4.	Polja cell-ova	12
3.3.5.	Strukture	13
3.4.	Brisanje varijabli	13
3.5.	Naredbe za rad s diskom, kontolu ispisa i naredbe operacijskog sustava	14
3.5.1.	Spremanje varijabli na disk i njihovo učitavanje	14
3.5.2.	Naredbe operacijskog sustava	15
3.5.3.	Naredbe za kontrolu ispisa	16
4.	Operacije u Matlabu	18
4.1.	Aritmetički operatori	18
4.2.	Relacijski operatori	19
4.3.	Logički operatori	20
4.4.	Naredbe odluke i ponavljanja	21
5.	Funkcije	23
5.1.	Elementarne matematičke funkcije	23
5.1.1.	Trigonometrijske i ciklotometrijske funkcije	24
5.1.2.	Logaritamske i hiperbolne funkcije	24
5.1.3.	Ostale funkcije	25
5.2.	Funkcije za obradu vektora i matrica	25
5.2.1.	Funkcije za definiranje matrica	25
5.2.2.	Relacijske i logičke funkcije	26
5.2.3.	Funkcije za obradu vektora	27
5.2.4.	Funkcije za obradu matrica	28
5.3.	Funkcije za obradu stringova	29
5.4.	Funkcije za rad s polinomima	29
5.5.	M funkcije	30
6.	GRAFIČKE FUNKCIJE MATLABA	32
6.1.	Dvodimenzionalni grafički prikaz	32
6.2.	Funkcije za uljepšavanje slike	33

6.3.	Funkcije za označavanje slike	34
6.4.	Trodimenzionalni grafički prikaz	34
7.	Literatura	35

1. UVOD

Potreba brze obrade laboratorijskih mjernih rezultata, kao i potreba za računanjem zasnovanom na matričnoj aritmetici dovela je do stvaranja osnovnih verzija programa za računanje u tehničkim znanostima kojima pripada i Matlab, razvijen od tvrtke MathWorks Inc. U grupi programa za računanje u tehničkim znanostima razvio se čitav niz programa, a od većih i poznatijih uz Matlab možemo spomenuti Matrix_x i Easy. Svi ovi programi nastali su iz istih potreba, pa imaju i slična korisnička sučelja i način rada, no sama izvedba i struktura su im različiti. Osnovna jezgra Matlaba izgrađena je u osnovi oko matričnog kalkulatora s mogućnošću izvršavanja niza naredbi grupiranih u *Skriptu*, odnosno funkciju. Mogućnost pisanja funkcija u Matlabu omogućila je izgradnju čitavog niza alata (Toolbox) koji danas zajedno s osnovnom jezgrom i korisničkim sučeljima čine Matlab. Upravo ova modularnost i otvorenost Matlaba za definiranje novih alata postala je velika prednost Matlaba u odnosu na ostale slične programe, koji imaju zatvorenu strukturu i nadogradnja je moguća samo od proizvođača. Otvorenost Matlaba omogućila je da praktično svi njegovi korisnici postanu u neku ruku i razvojni tim samog Matlaba. Tijekom razvoja u Matlabu su razvijeni alati za većinu djelatnosti iz područja tehničkih znanosti, a postoje i neki alati koji se primijenjuju izvan područje tehničkih znanosti.

Cilj ove skripte je prikaz osnovnih mogućnosti Matlaba, a to praktično znači njegove osnovne jezgre i nekih općenito češće korištenih alata. To znači da je skripta namijenjena početnicima u korištenju Matlaba. Prema tome, u skripti je prikazana gruba organizacija Matlaba koja korisniku omogućuje lakše shvaćanje logike rada s programom. Nadalje, navedene su strukture podataka koje se koriste u programu kao i operacije i funkcije koje su dio osnovne jezgre Matlaba. S obzirom da je rezultat proračuna često potrebno prikazati i grafički, prikazane su i grafičke funkcije Matlaba, kao i M funkcije koje omogućuju korisniku automatizaciju i ubrzanje računanja i prikaza rezultata.

Mogućnosti Matlaba ovime nisu niti približno iscrpljene. No mogućnosti kao što su simuliranje dinamičkih sustava, obrada signala, rad sa simboličkim izrazima, generiranje upravljačkog koda za procesorske sustave za rad u stvarnom vremenu i mnoge druge zahtijevaju detaljniji pristup svakom od pojedinih modula. No, kvalitetna dokumentacija i struktura *Help-a* u Matlabu, svakom korisniku nakon prikupljenih osnovnih znanja o programu omogućuju samostalno proširenje znanja i rada s Matlabom.

2. POKRETANJE I ORGANIZACIJA MATLABA

Matlab je interaktivni matrični kalkulator interpreterskog tipa, zasnovan na varijablama koji omogućuje izvođenje Matlab funkcija, i posjeduje grafičko korisničko sučelje. Naziv MATLAB dolazi od engleskih riječi MATrix LABoratory, što označava njegova osnovna svojstva, a to su rad s matričnim varijablama i primjena kod obrade mjernih rezultata u laboratoriju.

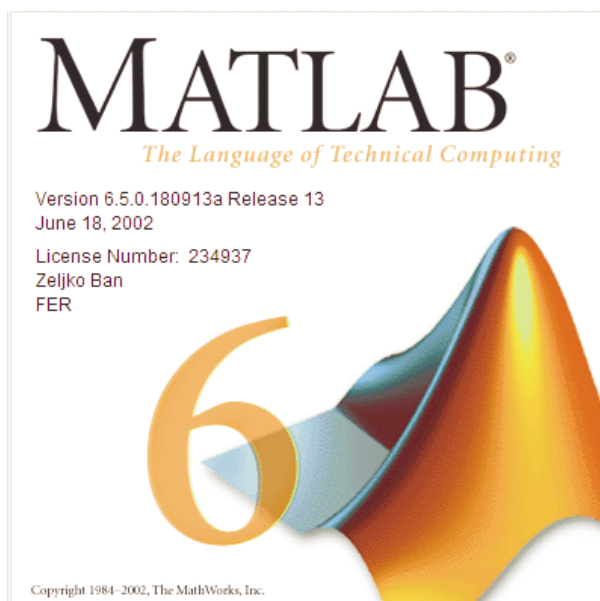
2.1. Pokretanje Matlaba

Pokretanje programa Matlab ovisno je o računalnoj platformi na kojoj se pokreće. Na PC računalima pod Windows operacijskim sustavom Matlab je moguće pokrenuti na 3 načina. To su:

- Dvostrukim klikom miša na ikonu Matlaba na *desktopu* windowsa,
- Primjenom **Start** izbornika
(Obično se Matlab nalazi na mjestu *Start*→*Programs*→*Matlab*→*Matlab*),
- Upisom naredbe **matlab** u **Run** prozor unutar *Start* izbornika.

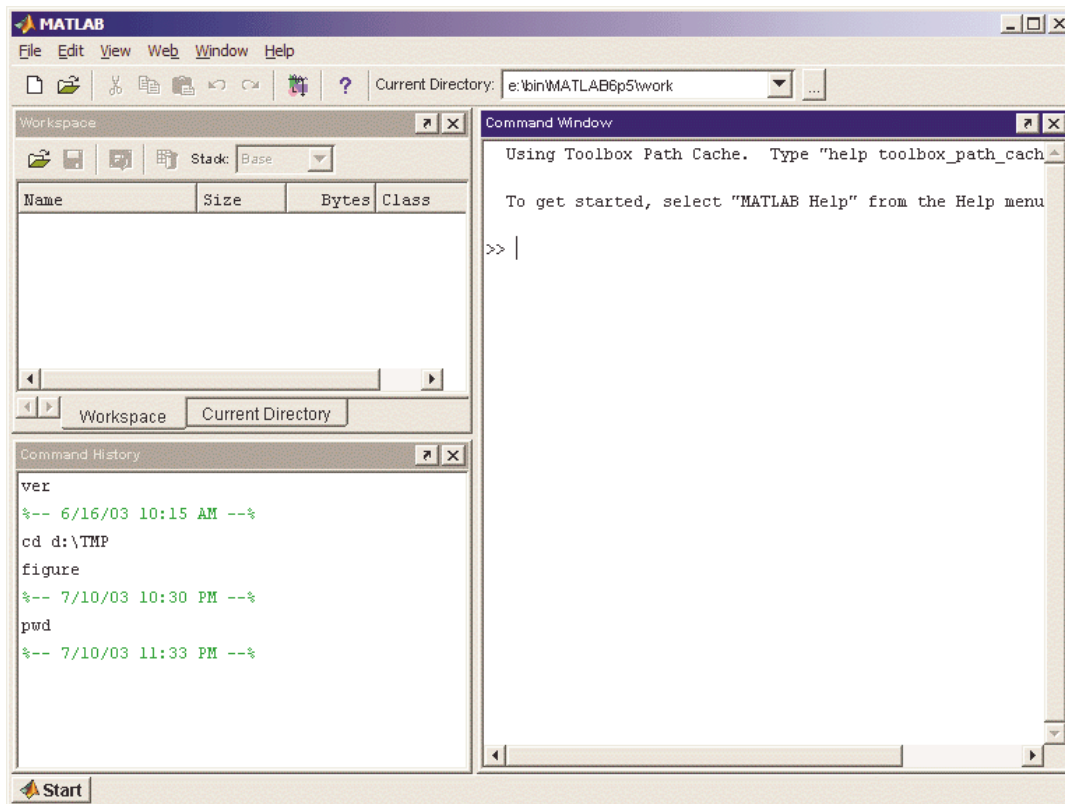


Prilikom startanja Matlab kratkotrajno prikazuje svoj znak prikazan slikom 2.1.



Sl. 2.1. Znak programskog paketa Matlab.

Nakon nestajanja Matlabova znaka pojavljuje se osnovni prozor Matlaba prikazan slikom 2.2.



Sl. 2.2. Komandni prozor programskog paketa Matlab.

Osnovni prozor Matlaba sadrži redak izbornika, redak s alatima (*toolbar*) i tri prozora. U lijevom gornjem dijelu osnovnog prozora smješten je prozor koji omogućuje uvid u memorijski prostor Matlaba (*Workspace*), odnosno u tekući direktorij. Prozor u lijevom donjem dijelu osnovnog prozora je *Command history* prozor, odnosno prozor koji sadrži popis svih upotrijebljenih naredbi. U desnom dijelu osnovnog prozora je komandni prozor. U njemu se upisuju naredbe Matlaba i dobivaju se numerički rezultati. Redak teksta ispisan u prozoru upućuje na *Help*, odnosno pomoć koju korisnik može zatražiti od programa. Znak `>>` predstavlja prompt Matlaba. To je znak koji se pojavljuje kao posljednji u prozoru i označava da je Matlab spreman za prihvatanje naredbi od korisnika. Iza prompta nalazi se kursor.

Iza prompta moguće je unositi naredbe Matlaba, pokretati funkcije i izvršavati matematičke operacije. Redak naredbe se izvršava nakon pritiska na tipku *Enter*. Važno je napomenuti da Matlab **razlikuje velika i mala slova** kod pisanja naredbi.

Završetak rada s Matlabom postiže se naredbom **quit**. Ova naredba vrijedi bez obzira na kojoj računalnoj platformi se Matlab izvodi. Kod izvođenja Matlaba na PC računalo pod Windowsima rad s Matlabom se može završiti odabirom naredbe *Exit* iz izbornika *File* ili bilo kojim od standardnih načina zatvaranja prozora u Windows okruženju.

Pomoć se u Matlabu dobiva naredbama *help* i *lookfor*. Naredba *help* sama za sebe ispiše popis svih podcjelina i toolboxova u kojima se dalje mogu naći tražene naredbe. Navođenjem naziva podcjeline iza naredbe *help* dobije se detaljniji popis mogućih naredbi. Za dobivanje sintakse i objašnjenja pojedine naredbe, potrebno je napisati:

help naredba

Gdje je *naredba* naziv naredbe za koju tražimo sintaksu. Na primjer pomoć za naredbu za crtanje bi bila: *help plot*.

Ukoliko nam nije poznat specifični naziv naredbe, za pomoć koristimo naredbu *lookfor* i to tako da iza naredbe navedemo **jednu ključnu riječ** koja označava akciju koju želimo provesti. Primjerice, potrebno je označiti osi na grafikonu, a ne zna se naredba za označavanje osi. U tom slučaju dovoljno je iza naredbe *lookfor* upisati pojam koji označava oznaku (*label*):

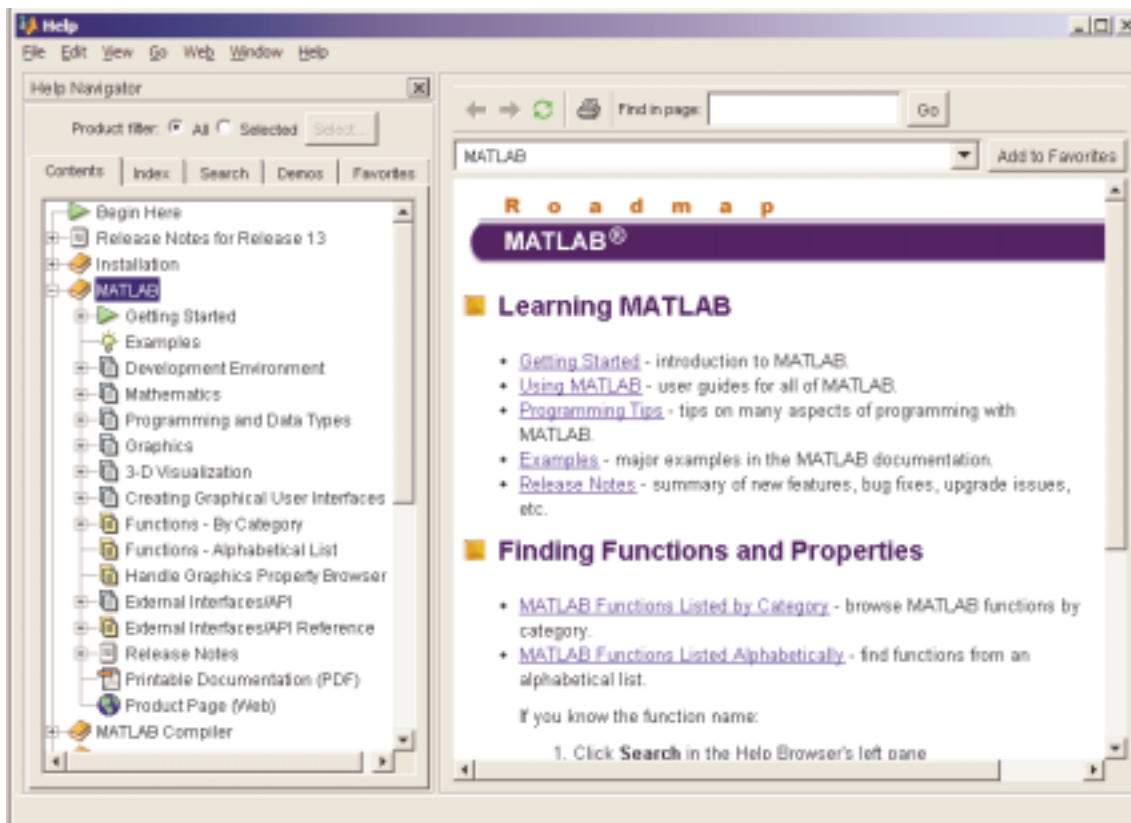
lookfor label.

Kao rezultat se dobije:

```
PLOTYY Graphs with y tick labels on the left and right.  
XLABEL X-axis label.  
YLABEL Y-axis label.  
ZLABEL Z-axis label.  
CLABEL Contour plot elevation labels...
```

Za svaku od navedenih naredbi pomoću *help* naredbe moguće je dobiti detaljnu sintaksu. Prema tome za označavanje x-osi moguće je koristiti naredbu *xlabel*, za y-os naredbu *ylabel* itd.

Naredbe *help* i *lookfor* daju sažeti prikaz sintakse naredbe i korisne su i iskusnim korisnicima Matlaba. Detaljan prikaz svih naredbi i funkcija Matlaba raspoloživ je u Matlabovoj dokumentaciji u Html i Pdf obliku. Do dokumentacije se dolazi iz *Help* izbornika odabirom opcije *Matlab Help*. Spomenuti izbornik otvara prozor prikazan slikom 2.3.



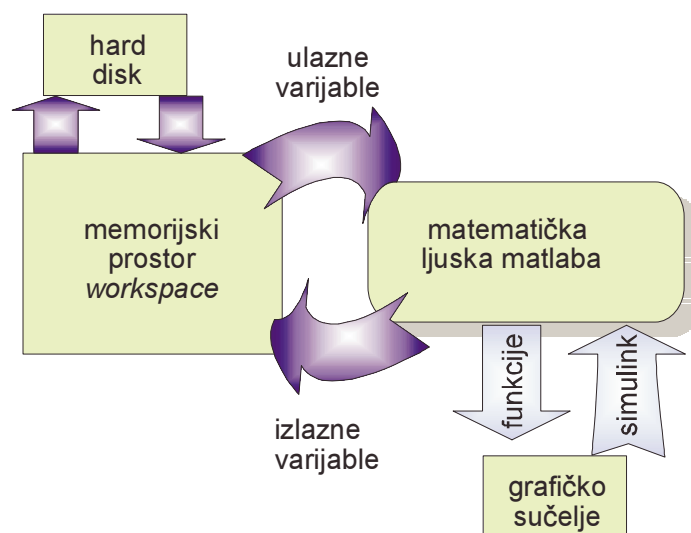
Sl. 2.3. Prozor On line dokumentacije Matlaba.

On line dokumentacija sadrži opis svih funkcija i alata koje korisnik ima na raspolaganju u svojoj instalacijskoj verziji Matlaba. U lijevom dijelu prozora (sl. 2.3.) odabire se način navigacije po dokumentaciji, dok se u desnom dijelu prikazuje izabrani sadržaj. Kao što je vidljivo, navigacija kroz dokumentaciju moguća je preko sadržaja, (*Contents*), Indeksa, pretraživanja po ključnim riječima (*Search*) ili odabirom demonstracijskih primjera. Odabirom jednog od načina navigacije i odabirom pojma u navigacijskom prozoru, dolazi se do željenog sadržaja u desnom prozoru.

2.2. Organizacija Matlaba i strukture podataka

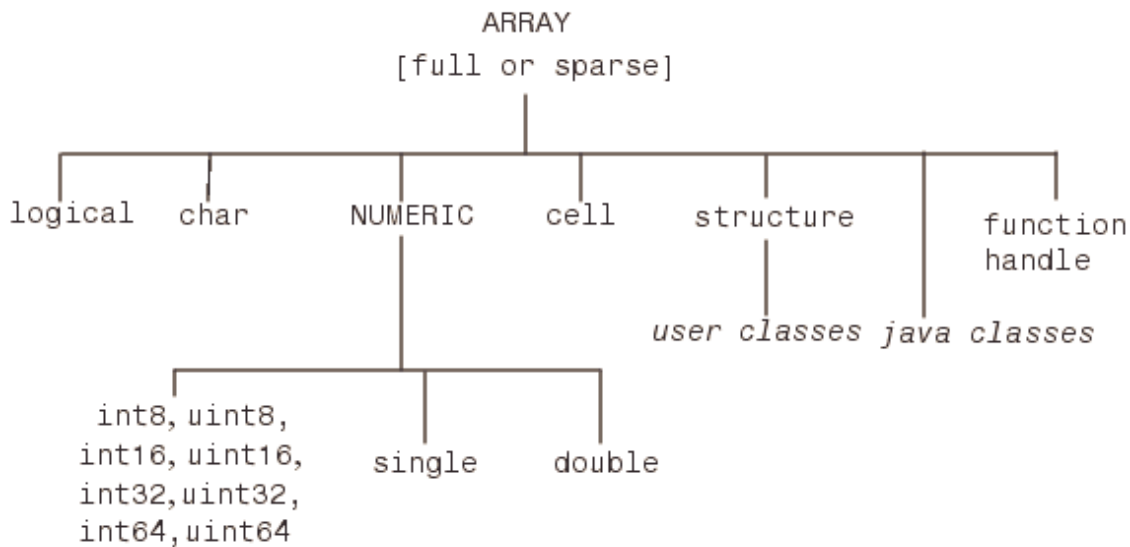
Cjelokupni rad u Matlabu zasniva se na radu s varijablama. Na definirane ulazne varijable primijenjuju se matematičke operacije i funkcije, a kao rezultat dobiju se izlazne varijable. Funkcionalna struktura Matlaba, koja lakše omogućuje predočenje načina rada u Matlabu, prikazana je slikom 2.4.

Varijable Matlaba smještaju se u memorijski prostor (*workspace*). Primjena matematičkih operacija odnosno funkcija na definirane varijable uzrokuje da matematička ljuska dohvaća varijable iz memorijskog prostora, izvodi zadanu operaciju ili funkciju i rezultat ponovno pohranjuje u izlaznu varijablu smještenu u memorijski prostor. Workspace Matlaba fizički je smješten u radnu memoriju računala, što znači da se prilikom izlaska iz programa, odnosno gašenja računala sve varijable iz workspacea gube. Zbog tog svojstva programa potrebno je sve varijable koje se žele sačuvati spremite na disk prije izlaska iz programa. Isto tako, kod pokretanja programa moguće je varijable spremljene kod prijašnjeg rada učitati s diska u radni prostor Matlaba (*workspace*).



Sl. 2.4. Funkcionalna struktura Matlaba.

Grafičko sučelje Matlaba omogućuje grafički prikaz rezultata, ali isto tako i definiranje matematičkog modela, tj. simulacijske sheme sustava pomoću grafičkih blokova unutar Simulinka. Kod grafičkog prikaza rezultata, funkcije za crtanje prikazuju sadržaj varijabli sadržanih u radnom prostoru Matlaba, u grafičkom obliku. Simulink, za razliku od njih na temelju blokova složenih od korisnika u grafičkom prozoru definira model s kojim je dalje moguće raditi u Matlabu i pohranjuje ga na disku. No rezultati operacija nad modelom u Simulinku pohranjuju se u varijable u workspaceu. Tipovi podataka u Matlabu prikazani su slikom 2.4.



Sl. 2.5. Tipovi podataka u Matlabu.

Svi podaci se temelje na matricama. Sadržaj matrica mogu biti logički, slovčani (*char*), numerički, *cell* i neki drugi specijalni tipovi podataka ovisni o funkcijama koje ih koriste. Korisnik sam može definirati i vlastite tipove struktura. Podaci su uvijek pridruženi varijablama.

3. VARIJABLE

Varijable u Matlabu su matrice, različitog sadržaja i dimenzije. Zbog takvog načina prikaza, skalarne veličine se u Matlabu tretiraju kao matrice dimenzije 1x1, dok su vektori stupčane ili redne matrice. Veličina (dimenzija) matrice ograničena je jedino raspoloživim memorijskim prostorom na računalu na kojem se Matlab izvodi. Varijable Matlaba se mogu podijeliti na više skupina, a njihova svojstva ovise o pripadnosti pojedinim skupinama. Tako varijable možemo podijeliti na slijedeće načine:

1. Prema sadržaju elemenata matrice na:

- realne,
- kompleksne,
- simboličke,
- polja cell-ova
- strukture

2. Prema dohvatu (vidljivosti) na:

- lokalne,
- globalne,

3. Prema izvoru nastanka na:

- interne,
- eksterne.

Varijable prema sadržaju spadaju u skupinu ovisno kakve elemente sadrže. Ukoliko su elementi matrice realni i varijabla se može nazvati realnom. Kompleksne varijable sadrže kompleksne brojeve čiji je imaginarni dio različit od nule, dok simboličke varijable sadrže simbole koji se ne interpretiraju numerički. Simboličke varijable koriste se kao ulazne i izlazne varijable kod simboličkog računa, kao što je simboličko rješavanje neodređenih integrala, algebarskih i diferencijalnih jednačbi i slično.

Globalne varijable su one varijable koje su vidljive iz više funkcija Matlaba. Varijabla postaje globalna tako da se deklarira naredbom:

```
>> global ime_varijable
```

Sve funkcije koje imaju napisanu definiciju vidjet će varijablu *ime_varijable* odnosno moći će dohvatiti njen sadržaj. Lokalne su varijable sve one koje nisu globalne. Prema tome ako postoji definirana varijabla u funkciji Matlaba, nakon izvršenja funkcije u *workspaceu* varijabla neće postojati. Razlog za to je što se radi o lokalnoj varijabli i ona je vidljiva samo u funkciji gdje je definirana.

Interne varijable su varijable koje definira sam Matlab, dok su eksterne varijable predstavljaju varijable definirane od korisnika ili varijable koje su proizašle kao rezultat matematičkih operacija i funkcija u Matlabu.

3.1. Interne varijable

Interne varijable služe za određivanje svojstva određenih matematičkih operacija. Interne varijable nije uputno koristiti u druge svrhe, jer se mogu polučiti nepredviđeni rezultati. Njih nije moguće izbrisati. Ukoliko se imenu interne varijable pridruži neka druga vrijednost, originalna vrijednost se restaurira brisanjem varijable naredbom *clear*.

Matlab sadrži slijedeće interne varijable:

- | | | |
|----------------------------------|---|---|
| - eps = 2.2204e-016 | - | točnost real. brojeva (razlika između 1.0 i prvog većeg), |
| - realmin=2.225073858507202e-308 | - | vrijednost najmanjeg pozitivnog realnog broja, |
| - realmax=1.797693134862316e+308 | - | vrijednost najvećeg pozitivnog realnog broja, |
| - pi = 3.14159265358979 | - | vrijednost broja π , |
| - inf = 1/0 | - | vrijednost kod dijeljenja s nulom, |
| - NaN = 0/0 | - | neodređena vrijednost, |
| - flops | - | broj floating point operacija, |
| - i, j | - | imaginarne jedinice kompleksnih brojeva. |

Varijabla flops sadrži broj floating point operacija i može se koristiti za mjerenje

efikasnosti nekog algoritma. Na početku algoritma može se postaviti na nulu naredbom $flops(0)$,

a na kraju algoritma ona sadrži broj floating point operacija koje su izvedene nakon njenog postavljanja na nulu. Sadržaj internih varijabli može se koristiti u funkcijama i operacijama u Matlabu, što znači da one mogu biti sadržane u matematičkim izrazima.

3.2. Eksterne varijable

Eksterne varijable su varijable definirane od korisnika ili varijable nastale kao rezultat matematičkih operacija i funkcija izvedenih u Matlabu. **Ime varijable** sastoji se od jednog do najviše **19 alfanumeričkih znakova** (slova engleske abecede, brojke i _ (underscore)), a počinje slovom. U imenu varijable se razlikuju velika i mala slova. To znači da na primjer *cijev1* i *Cijev1* predstavljaju imena dviju varijabli.

3.3. Definiranje varijable

Varijabla se definira i istovremeno joj se pridjeljuje vrijednost na slijedeći način:

$a=2;$

Naredba je pridružila skalarnu vrijednost 2 varijabli s imenom *a*. **Točka-zarez** na kraju naredbe označavaju da se rezultat ne ispisuje na ekran.

Na ovaj način varijabli se pridružuje skalar, vektor ili matrica. Razlika je u tome što kod vektora i matrice treba upisati i simbole za početak i kraj matrice odnosno vektora, a to su uglate zagrade. Karakteristika matrica je da joj elementi moraju biti istog tipa. To znači svi moraju biti ili numerički ili simbolički ili polja cell-ova odnosno strukture.

3.3.1. Vektori i matrice

Pridruživanje **vektora** oblika

$$b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4+2i \\ 7 \end{bmatrix}$$

varijabli *b* može se izvesti na slijedeći način:

$b=[1;2;3;4+2*i;7]$

Kao rezultat dobije se:

b =

```

1.0000
2.0000
3.0000
4.0000 + 2.0000i
7.0000

```

Otvorena uglata zagrada znači početak definicije vektora ili matrice. Retci unutar vektora odvajaju se znakom točka-zarez, a kraj definicije vektora ili matrice je zatvorena uglata zagrada. Na kraju naredbe nije stavljena točka-zarez, što je uzrokovalo ispis rezultata definicije vektora na ekran.

Za definiranje kompleksnog broja unutar vektora upotrijebljena je interna varijabla *i* koja predstavlja imaginarnu jedinicu. Isti se rezultat može postići da se umjesto znaka točka-zarez za odvajanje redaka koristi tipka *Enter*. Matlab u tom slučaju ne daje komandni prompt sve dok se ne zatvori uglata zagrada.

Matrica se definira na sličan način kao vektor, osim što uz znakove za odvajanje redaka treba dodati i znakove za odvajanje stupaca. Stupci se odvajaju praznim mjestom (*space*) ili zarezom. Elementi matrice se unose po retcima, počevši od prvog.

Primjer:

Naredba:

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

daje kao rezultat matricu:

```

c =
     3     2     1
     7     8     9
     4     5     6
     3     4     2

```

Varijable *a*, *b* i *c* definirane na opisani način pohranjene su u Matlabovoj radnoj memoriji (*workspace*). Ispis imena svih eksternih varijabli iz Matlabovog radnog prostora postiže se naredbama **who** ili **whos**. Naredba *who* daje popis imena bez opisa pojedinih varijabli, dok naredba *whos* uz ime varijable ispisuje veličinu, zauzeće u byteovima i tip sadržaja varijable.

Primjer: Nakon definirane varijable *a*, *b*, i *c* na opisani način, naredbe *who* i *whos* daju slijedeće rezultate:

```
» who
```

Your variables are:

```
a          b          c
```

```
» whos
```

Name	Size	Bytes	Class
a	1x1	8	double array
b	5x1	80	double array (complex)

```
c          4x3          96  double array
```

```
Grand total is 18 elements using 184 bytes
```

Iz primjera je vidljivo da je varijabla *a* skalarna (dakle matrica dimenzije 1x1) i da zauzima 8 byteova, dok je njen sadržaj tipa *double array*.

Varijabla *b* je vektor dimenzije 5x1, a njegov sadržaj su kompleksni brojevi spremljeni u elemente tipa *double array*. Zadnja linija izvještaja daje sumarne podatke o ukupnom broju elemenata i ukupnom zauzeću memorijskog prostora.

Na ovaj način se dobije ispis varijabli s pripadnim dimenzijama, međutim, ukoliko je potrebno dimenziju neke varijable upisati u drugu varijablu (da bi se mogle definirati granice brojača kod obrade takve varijable), to nije moguće učiniti naredbom *whos*. Za to postoji naredba *size*.

c_dim=size(c).

Ovom naredbom je stvorena varijabla *c_dim* koja ima dva elementa, od kojih prvi sadrži broj stupaca, a drugi broj redaka varijable čije je ime upotrebjeno kao argument funkcije *size*.

Za **ispis bilo koje varijable** iz radnog prostora Matlaba na ekran dovoljno je samo upisati njeno ime i pritisnuti tipku *Enter*.

3.3.2. Definiranje niza brojeva

Unos vektora i matrica opisan u prethodnom poglavlju prikladan je za upis matrica i vektora s malim brojem elemenata. Međutim, upis elemenata vektora koji predstavljaju aritmetički niz od više tisuća elemenata, predstavljao bi iscrpljujući posao. Da bi se olakšao unos takvih nizova, postoje naredbe koje to olakšavaju:

d1=[var_min:prirast:var_max]

Navedena naredba će varijabli *d1* pridružiti niz brojeva čiji će početna vrijednost biti *var_min*, konačna vrijednost će biti *var_max*, a razlika dvaju susjednih elemenata niza iznositi će *prirast*.

Primjer:

```
» d1=[0:0.5:5]
```

```
d1 =  
Columns 1 through 7  
0    0.5000    1.0000    1.5000    2.0000    2.5000    3.0000  
Columns 8 through 11  
3.5000    4.0000    4.5000    5.0000
```

Naredbom u primjeru stvoren je redni vektor od 11 elemenata (dimenzije 1x11) čija je najmanja vrijednost 0, najveća 5, a prirast elemenata iznosi 0.5.

Oblik naredbe

d1=[var_min:var_max]

stvara redni vektor čiji su elementi aritmetički niz s početnom vrijednošću *var_min*, konačnom vrijednošću *var_max* i razlikom među susjednim elementima niza jednakom 1.

Nizovi brojeva se još mogu stvoriti s dvije funkcije:

d2=linspace(min,max,br_toc)

d3=logspace(n1,n2,br_toc)

Funkcija *linspace* stvori aritmetički niz čiji je prvi element jednak prvom argumentu funkcije (*min*), zadnji element je jednak drugom argumentu funkcije (*max*), a broj elemenata niza određen je trećim argumentom funkcije *br_toc*.

Primjer:

```
d2=linspace(1,-1,5)
```

```
d2 =
```

```
1.0000    0.5000         0   -0.5000   -1.0000
```

Početna vrijednost niza je 1, konačna -1 i niz mora sadržavati 5 elemenata.

Funkcija *logspace* tvori logaritamski niz koji sadrži broj elemenata određen trećim argumentom funkcije (*br_toc*), dok je početna vrijednost niza određena iznosom 10^{n1} , gdje je *n1* prvi argument funkcije, dok konačna vrijednost niza odgovara 10^{n2} , gdje je *n2* drugi argument funkcije. Elementi tako stvorenog vektora imaju logaritamsku raspodjelu.

Primjer:

```
» d3=logspace(-1,2,7)
```

```
d3 =
```

```
0.1000    0.3162    1.0000    3.1623   10.0000   31.6228  100.0000
```

Polje *d3* iz prethodnog primjera ima početnu vrijednost 0.1 (što odgovara 10^{-1}), konačnu vrijednost 100, a sadrži 7 elemenata raspoređenih logaritamski.

3.3.3. Ekstrakcija dijela matrice

Postoje slučajevi primjene operatora i funkcija Matlaba, kad je potrebno djelovati samo na dio matrice ili vektora, to znači samo na neku submatricu unutar postojeće matrice. U tom slučaju se adresira varijabla uz naznačavanje početnog i konačnog retka, odnosno početnog i konačnog stupca na koji se naredba odnosi. Sintaksa naredbe je slijedeća:

ime_var(n1:n2,m1:m2),

gdje je *ime_var* varijabla koja sadrži matricu dimenzije *nxm*.

n1 i *n2* su brojevi početnog i konačnog retka submatrice i moraju biti u intervalu $1 \leq n1 \leq n2 \leq n$, dok *m1* i *m2* predstavljaju brojeve početnog i konačnog stupca submatrice i moraju se nalaziti u intervalu $1 \leq m1 \leq m2 \leq m$. Za adresiranje jednog stupca dovoljno je iza zareza umjesto intervala

navesti samo broj stupca koji se želi, dok se adresiranje svih stupaca matrice iza zareza unosi znak dvotočke umjesto intervala.

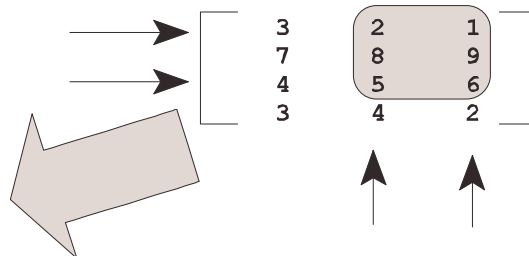
Primjer:

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

» c(1:3,2:3)

ans =
     2     1
     8     9
     5     6
```

U ovom primjeru adresirana je submatrica matrice c od 1. do 3. retka i od 2. do 3. stupca.



```
» c(:,2)

ans =
     2
     8
     5
     4
```

```
» c(3,:)

ans =
     4     5     6
```

```
» c(:,1:2)

ans =
     3     2
     7     8
     4     5
     3     4
```

3.3.4. Polja cell-ova

Polja cell-ova definiraju se slično kao i matrice samo što su znakovi za početak i kraj polja vitičaste zagrade umjesto uglatih. Karakteristika cell-ova je da mogu sadržavati različite tipove podataka. Tako se u jednom cell-u mogu naći numerički podaci, podaci tipa char, pa čak i matrice različitih dimenzija. Polje se definira na slijedeći način:

```
>> xx={1 2 3 'pero';7 'eustahije' 4 5}
```

```
xx =
```

```
 [1] [          2] [3] 'pero'
 [7] 'eustahije' [4] [ 5]
```

U navedenom primjeru, polje xx sadrži numeričke podatke kao i polja karaktera. Adresiranje pojedinog elementa polja izvodi se na isti način kao kod matrice, samo što se koriste vitičaste umjesto uglatih zagrada. Npr. podatak *pero* nalazi se u prvom retku i četvrtom stupcu polja xx. Do tog se podatka može doći naredbom:

```
xx{1,4}
```

Nakon adresiranja nekog člana polja, moguće mu je pridružiti novu vrijednost.

3.3.5. Strukture

Strukture podataka omogućuju kompaktnije združivanje podataka koji trebaju pripadati određenoj cjelini. Strukture nastaju kao rezultati određenih funkcija Matlaba, ili definiranjem od strane korisnika.

Struktura se može definirati naredbom:

```
S = STRUCT('field1',VALUES1,'field2',VALUES2,...)
```

gdje su:

- field - imena polja u strukturi, a definiraju se kao *string*,
- values - vrijednosti koje se pridružuju poljima strukture.

Primjer definiranja strukture dan je naredbom:

```
>> a3=struct('imena',{ 'pero', 'eustahije'; 'brzic', 'janko' }, 'brojevi', [1 2 3 4])  
a3 =  
    imena: {2x2 cell}  
    brojevi: [1 2 3 4]
```

Struktura *a3* sadrži dva polja: polja imena koje sadrži polje cell-ova i polje brojevi koje sadrži rednu matricu. Adresiranje pojedinog dijela strukture vrši se na način da se iza imena strukture stavi točka i ime polja, a elementi unutar polja se adresiraju na način kako je to već za dani tip podataka određeno. Prema tome ispis polja *imena* iz strukture *a3* dobije se naredbom:

```
a3.imena  
  
ans =  
    'pero'      'eustahije'  
    'brzic'     'janko'
```

Ispis elementa *eustahije* iz polja *imena* strukture *a3* koji se nalazi u prvom retku i drugom stupcu polja, dobit će se naredbom:

```
>> a3.imena{1,2}  
  
ans =  
eustahije
```

Pošto je polje *imena* tipa *cell*, elementi unutar njega adresiraju se pomoću vitičastih zagrada. Promjena sadržaja pojedinog elementa strukture postiže se tako da se željeni element adresira i pridruži mu se nova vrijednost.

3.4. Brisanje varijabli

Brisanje varijable iz radnog prostora Matlaba postiže se naredbom *clear*. Naredba *clear* bez argumenata briše sve varijable iz radnog prostora. Za brisanje samo određenih varijabli potrebno je iza naredbe navesti listu varijabli koje se žele obrisati. Brisanjem varijabli oslobađa se zauzeti radni prostor (memorija računala).

Primjer:

`clear a b c` - briše varijable a, b i c iz radnog prostora.

Za brisanje samo globalnih varijabli iz radnog prostora postoji naredba

clear global.

U tom slučaju lokalne varijable ostaju nedirnute. Osim toga, u imenu varijable je moguće koristiti i zvijezdicu (isto vrijedi i za naredbu `whos`).

Naredba

clear d*

briše sve varijable u radnom prostoru čije ime počinje s d (npr.: d, d1, d12, d3, ...).

Varijable se mogu brisati i u prozoru *Workspace* tako da se selektiraju i odabere se opcija delete na tipkovnici ili pomoću desne tipke miša za vrijeme kad je kursor miša unutar spomenutog prozora iznad selektirane varijable ili grupe varijabli.

3.5. Naredbe za rad s diskom, kontolu ispisa i naredbe operacijskog sustava

Varijable Matlaba sadržane u radnom prostoru (*workspace*) mogu se koristiti u matematičkim izrazima i funkcijama, no izlaskom iz programa njihov se sadržaj gubi. Da bi se sačuvale za rad kod ponovnog pokretanja Matlaba, potrebno ih je spremati na disk. Matlab sadrži nekoliko naredbi za rad s diskom, u koje su uključene naredbe za spremanje varijabli na disk i učitavanje varijabli s diska u radni prostor, naredbe za prikaz i promjenu tekućeg direktorija, ispis sadržaja direktorija ispis sadržaja datoteka, brisanje datoteka, odnosno izvršenje naredbi operacijskog sustava.

3.5.1. Spremanje varijabli na disk i njihovo učitavanje

Spremanje podataka na disk izvodi se naredbom *save*. Naredba *save* bez daljnjih argumenata sprema sve varijable iz radnog prostora Matlaba u datoteku *matlab.mat* koja se smješta u tekući direktorij.

Postoji još nekoliko načina upotrebe naredbe *save*:

save ime_datoteke - sprema sve varijable u datoteku pod nazivom *ime_datoteke*. Ako ime datoteke ne sadrži ekstenziju, Matlab dodaje ekstenziju *mat*. Ukoliko je u imenu datoteke sadržan kompletan put (ime diska, ime direktorija i ime datoteke), datoteka se sprema na specificirano mjesto umjesto u tekući direktorij.

save ime_datoteke var1 var2 var3 - u datoteku *ime_datoteke* spremaju se samo varijable *var1*, *var2* i *var3*.

Podaci zapisani u datoteku na ovaj način su u binarnom Matlab formatu. Ukoliko se žele zapisati tako da ih može pročitati bilo koji program koji ne podržava Matlabov format zapisa podataka

potrebno ih je zapisati u ASCII obliku. To se postiže naredbom:

save ime_datoteke var1 var2 var3 -ascii

Opcija na kraju određuje da se ne radi o standardnom Matlabovom zapisu, nego o ASCII formatu.

Učitavanje varijabli iz datoteke na disku postiže se naredbom ***load***. Pri tome naredba *load* bez argumenata učitava sve varijable iz datoteke *Matlab.mat* iz tekućeg direktorija.

Naredba

load ime_datoteke

čita sve varijable iz datoteke pod imenom *ime_datoteke*. Ako ekstenzija datoteke nije navedena, Matlab automatski podrazumijeva ekstenziju *mat*. Ukoliko se u nastavku naredbe doda i lista varijabli (kao kod naredbe *save*), iz specificirane datoteke će biti učitane samo navedene varijable.

Učitavanje podataka iz ASCII datoteka koje su dobivene mjerenjem ili nekim drugim programom moguće je također naredbom *load*. Jedini je uvjet da datoteka ima stupce odvojene praznim mjestima (space) i da ima u svim stupcima isti broj redaka (da izgleda kao matrica).

Nakon učitavanja takve datoteke, Matlab će njen sadržaj pridružiti varijabli s imenom jednakim imenu datoteke bez ekstenzije.

3.5.2. Naredbe operacijskog sustava

Kod naredbi za spremanje varijabli na disk, spominje se tekući direktorij. Tekući direktorij je direktorij iz kojeg je program pokrenut, odnosno koji je postavljen iz samog programa da bude tekući.

Ispis imena tekućeg direktorija postiže se naredbom ***pwd***. Tekući direktorij vidljiv je i u lijevom gornjem prozoru osnovnog prozora Matlaba odabirom opcije *Current directory*. U istom prozoru moguće je odabrati željeni direktorij i postaviti ga da bude tekući.

Primjer:

```
» pwd
ans =
C:\usr\phd\matlab\opti
```

Promjena tekućeg direktorija postiže se naredbom

cd ime_direktorija.

Ime direktorija se piše onako kako je to uobičajeno za operacijski sustav na kojem se program izvodi (na PC računalima pod Windowsima koristi se DOS sintaksa imena direktorija).

Sadržaj direktorija moguće je ispisati naredbom ***dir*** ili ***ls***. Bilo koja od navedenih naredbi ispiše imena svih datoteki u tekućem direktoriju. Specificiranje dijela imena datoteke u kombinaciji sa zvjezdicom omogućuje definiranje ispisa imena datoteka koje imaju dio imena zajednički. Naredba se koristi isto kao i u operacijskom sustavu.

Slična naredbi *dir* je i naredba ***what***. Ona ispisuje imena (bez ekstenzije) svih datoteka s ekstenzijom *m*, *mat* i *mex* u tekućem direktoriju.

Ispis sadržaja ASCII datoteke postiže se naredbom

type ime_datoteke.

Brisanje cjelokupne datoteke sa diska postiže se naredbom

delete ime_datoteke.

Izvođenje bilo koje naredbe operacijskog sustava bez izlaska iz Matlaba postiže se tako da se na komandnom promptu Matlaba upiše uskličnik, a iza njega naredba operacijskog sustava.

Na primjer:

!md vježba

Zbog uskličnika se naredba *md vježba* prosljeđuje operacijskom sustavu koji će (ako se radi o DOS-u) u tekućem direktoriju otvoriti poddirektorij *vježba*.

3.5.3. Naredbe za kontrolu ispisa

Sadržaj varijabli, odnosno rezultate matematičkih operacija moguće je ispisivati u nekoliko standardnih načina. Sprečavanje ispisa rezultata na ekran postiže se postavljanjem znaka točka-zarez iza naredbe (korisno je kad je rezultat vektor od više tisuća redaka, koji će se crtati).

Sam način ispisa određuje se naredbom *format*. Ta naredba ima niz argumenata koji određuju oblik zapisa. Naredba zajedno s argumentima i primjerom ispisa vrijednosti 0.2 prikazana je tablicom.

Naredba	Opis	Primjer
format short	Zapis s 4 decimalna mjesta	0.2000
format long	Zapis s 14 decimalnih mjesta	0.2000000000000000
format short e	Eksponecijalni zapis s 4 decimalna mjesta	2.0000e-001
format long e	Eksponecijalni zapis s 14 decimalna mjesta	2.0000000000000000e-001
format hex	Heksadecimalni zapis	
format bank	Zapis primjenjiv kod novčanih transakcija (dvije decimale)	0.20
format rat	Aproksimacija omjerom malih cijelih brojeva	1/5
format +	Ispisuje + odnosno - za brojeve	0.2 -----> + -0.2 -----> - 0 ----->
format compact	Izbacivanje dodatnih praznih linija iz rezultata	
format loose	Dodavanje praznih linija u rezultat	

Kod eksponencijalnog zapisa slovo *e* označava brojku 10, a broj iza nje eksponent broja 10. Prema tome vrijedi: $2.0000e-1 = 2.0 \cdot 10^{-1}$.

Brisanje sadržaja Matlabova komandnog prozora (ne varijabli iz radnog prostora), postiže se naredbom *clc*.

4. Operacije u Matlabu

Matlab je interaktivni program interpreterskog tipa, što znači da naredbe upisane iza komandnog prompta neposredno izvodi. Isto tako, *m funkcije* upisane editorom na disk se mogu direktno izvoditi bez prethodnog prevođenja. Operacije Matlabamogu se podijeliti una nekoliko grupa:

1. Aritmetički operatori,
2. Relacijski operatori,
3. Logički operatori,
4. Naredbe odluke i ponavljanja,
5. Funkcije,
6. Simulink.

Rezultat operacija se sprema u naznačenu varijablu. Ako se ne naznači varijabla za spremanje rezultata, Matlab stvara varijablu pod nazivom *ans* (dolazi od engleske riječi *answer*) i u nju pohranjuje rezultat. Među navedenim grupama operacija, simulink po načinu izvedbe pripada u grupu funkcija, međutim zbog značaja kod simulacija u području automatike, izdvojen je kao posebna cjelina.

4.1. Aritmetički operatori

Aritmetički operatori omogućuju aritmetičke operacije nad skalarnim odnosno matičnim varijablama. Popis operatora prema prioritetu izvođenja dan je u tablici.

Prioritet	Operator	Opis	Primjer
1.	()	Zagrade grupiraju izraz i daju najveći prioritet	$a*(b+c)$
2.	' '	Konjugiranje i transponiranje matrice (vektora) Transponiranje matrice (vektora)	a' $a.'$
3.	^ .^	Potenciranje Potenciranje među elementima matrice	a^3 $a.^b$
4.	* .* / \ ./	Množenje skalara ili matrica Množenje među elementima matrice Desno dijeljenje matrica lijevo dijeljenje matrica Dijeljenje među elementima	$a*b$ $a.*b$ b/a $b\backslash a$ $b./a$
5.	+ -	Zbrajanje Oduzimanje	$a+b$ $a-b$

Prema tablici, najviši prioritet (br. 1.) imaju zagrade, a najniži zbrajanje. Operacije unutar izraza

se izvršavaju počevši od najvišeg prioriteta prema najnižem, a izrazi istog prioriteta izvršavaju se od lijeva prema desno.

*Primjer: $11+a(x+y)^3*2$*

redosljed operacija:

1. $rez=x+y$
2. $rez=rez^3$
3. $rez=a*rez$
4. $rez=rez*2$
5. $rez=11+rez$

Potenciranje matrice moguće je izvesti samo kod kvadratnih matrica, dok potenciranje među elementima matrica postavlja potrebu da matrice budu istih dimenzija, ali ne moraju nužno biti kvadratne.

Množenje matrica provodi se po zakonima množenja matrica, što znači da matrice moraju imati odgovarajuće dimenzije. Dijeljenje matrica odgovara množenju inverznom matricom (odnosno pseudo inverznom matricom kod nekvadratnih matrica). Desno dijeljenje matrica:

$\underline{x}=\underline{b}/\underline{A}$ daje rješenje jednadžbe $\underline{x}*\underline{A}=\underline{b}$, dok

lijevo dijeljenje

$\underline{x}=\underline{A}\backslash\underline{b}$ daje rješenje jednadžbe $\underline{A}*\underline{x}=\underline{b}$.

Ukoliko je kod množenja jedan od operandi skalar, tad ne postoje ograničenja na dimenzije matrice, a kao rezultat se dobije matrica čiji je svaki element pomnožen navedenim skalarom.

Množenje, dijeljenje i potenciranje među elementima matrica zahtijeva da matrice budu istih dimenzija, osim ukoliko je jedan od operandi skalar.

Element rješenja kod množenja među elementima matrica dobije se kao

$c(i,j)=a(i,j)*b(i,j) \forall i=1,\dots,m, j=1,\dots,n$, kod matrica dimenzije $m \times n$.

Za uspješno zbrajanje i oduzimanje matrica matrice moraju biti istih dimenzija ili jedan od operandi mora biti skalar. U slučaju zbrajanja skalara i matrice rezultat je matrica čiji svaki element odgovara sumi elementa početne matrice i skalara.

4.2. Relacijski operatori

Relacijski operatori su operatori odnosa među dvjema varijablama, a kao rezultat daju logičku varijablu. Logička varijabla je realna varijabla koja ima dvije moguće vrijednosti i to: nulu ako operator nije zadovoljen i jedinicu ako operator je zadovoljen. Ovakav način definiranja logičke varijable omogućuje da se rezultat relacijskih operatora direktno koristi kako u logičkim tako i u aritmetičkim izrazima. Relacijski operatori mogu se primijenjivati među matricama i među skalarima, odnosno između matrice i skalara.

Kod primjene relacijskih operatora između dviju matrica, matrice moraju biti istih dimenzija, operator se primjenjuje među odgovarajućim elementima matrice, a rezultat je matrica istih dimenzija kao matrice između kojih je operator primijenjen. Operator između matrice i skalara daje kao rezultat matricu dimenzija jednakih ulaznoj matrici, a dobije se primjenom operatora između skalara i svakog elementa matrice. Matlab podržava slijedeće operatore:

Operator	Opis	Primjer
<	Manje	a<b
<=	manje i jednako	a<=b
>	veće	a>b
>=	veće i jednako	a>=b
==	jednako	c==2
~=	različito	a~=c

Primjer:

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \end{bmatrix}$$

<pre> » a<=b ans = 1 1 0 1 1 0 </pre>	<pre> » a<=3 ans = 1 1 1 0 0 0 </pre>	<pre> » a~=b ans = 1 0 1 1 0 1 </pre>
--	--	---

4.3. Logički operatori

Logički operatori primjenjuju se između logičkih varijabli i kao rezultat daju logičku varijablu. Logički se operatori mogu primijeniti i između realnih varijabli, no tad se realne varijable interno pretvaraju u logičke prije samih operacija i to na način da realna 0 prelazi u logičku 0 (*false*), a svi ostali realni brojevi prelaze u logičku jedinicu. Ukoliko su varijable među kojima se provode logički operatori matrice ili kombinacije matrica i skalara, vrijede ista pravila kao i kod relacijskih operatora. To znači da matrice moraju biti istih dimenzija, rezultatna matrica je rezultat operacija među elementima matrica, kod operacija između skalara i matrica elementi rezultatne matrice su rezultat operacija između skalara i odgovarajućeg elementa matrice. Logički operatori raspoloživi u Matlabu prikazani su tablicom.

Operator	Opis	Primjer
&	logičko I	a&b
	logičko ili	a b
~	logički komplement (ne)	~a
xor	logičko ekskluzivno ili	xor(a,b)

Logički operatori su istog prioriteta, ali im je prioritet viši od relacijskih operatora. Kod izraza logičkih operatora, odnosno miješanih izraza logičkih relacijskih i aritmetičkih operatora dozvoljeno je koristiti zagrade za točno određivanje prioriteta izvođenja.

Primjer (za matrice a i b iz prethodnog primjera)

<pre> » a<=2&b==3 ans = 1 0 0 0 0 0 </pre>	<pre> » ~(b>=3) ans = 0 1 1 0 0 0 </pre>	<pre> » a&b ans = 1 1 1 1 1 1 </pre>
--	--	---

Pošto su a i b realne matrice sa svim elementima različitim od nule, rezultat & operatora između njih je matrica čiji su svi elementi jedinice.

4.4. Naredbe odluke i ponavljanja

Aritmetički, logički i relacijski operatori omogućuju uspješno zadavanje pojedinačnih naredbi na komandnom promptu. Međutim, za programiranje potrebne su osim navedenih naredbi i naredbe odluke i ponavljanja koje omogućuju tvorbu višestruko izvršavajućih petlji i grananje unutar toka izvođenja programa. Naredbe za ponavljanje moguće je koristiti i na komandnom promptu i u *m-funkcijama*, dok upotreba naredbi odluke ima smisla samo u funkcijama. Korištenje naredbe ponavljanja na komandnom promptu uzrokuje da matlab ne daje novi prompt sve do završetka petlje za ponavljanje.

Naredbe odluke imaju slijedeći oblik:

```

if logički_izraz
    naredbe;
elseif logički_izraz
    naredbe;
else
    naredbe;
end

```

Naredbe odluke uvijek počinju naredbom if, a završavaju naredbom end. Naredbe elseif i else ne

moraju nužno postojati u naredbama odluke. Naredbe *if*, *elseif*, *else* i *end* predstavljaju ključne riječi naredbi odluke i odjeljuju blokove naredbi koji se u pojedinom slučaju izvode. Logički izraz je skalarni izraz logičkih i relacijskih operatora ili logička skalarna varijabla. Ukoliko je logički izraz iza naredbe *if* istinit, u tom slučaju se izvode naredbe koje slijede u retcima između naredbi *if* i *elseif* i nakon toga se perskaču sve naredbe do naredbe *end*. Ako izraz nije istinit, ispituje se izraz iza *elseif* naredbe (ako ona postoji). Istinitost toga logičkog izraza omogućuje izvođenje bloka naredbi između *elseif* i slijedeće ključne naredbe bloka odluke i odlazak na naredbu *end*. Nezadovoljenje niti jednog logičkog uvjeta unutar naredbi odluke uzrokuje izvođenje naredbi iza naredbe *else* i odlazak na naredbu *end*.

Primjeri:

```
if a>=b & a<c
    a=0;
end
```

```
if a>=b & a<c
    a=0;
elseif a>c
    a=c^2
end
```

```
if a>=b & a<c
    a=0;
elseif a>c
    a=c^2;
else
    a=c*2;
end
```

```
if a>=b & a<c
    a=0;
elseif a>c
    a=c^2;
elseif a<=b
    a+c/2;
else
    a=3;
end
```

Naredbe za ponavljanje imaju oblik:

```
for varijabla=izraz,
    naredbe;
end
```

Kod *for* petlje naredbe između *for* i *end* naredbi izvode se za svaku vrijednost *varijable* koja je definirana izrazom. Obično se izrazom definira početna i konačna vrijednost na način kako se definiraju nizovi brojeva u vektorima, no moguće je specificirati i inkrement. Moguće je definirati više petlji jedne unutar druge. Dvije petlji ponavljanja prikazane su primjerom:

```
for i=1:10,
    for j=1:2:14,
        a(i,j)=10*i+j;
    end
end
```

U ovom primjeru vanjska petlja kontrolirana je varijablom *i* koja će poprimiti vrijednosti 1, 2, ... 10, dok je unutarnja petlja kontrolirana varijablom *j* koja poprima vrijednosti 1, 3, 5, ... 13. Za jednu vrijednost varijable vanjske petlje izvedu se sva ponavljanja unutarnje petlje, pa se onda prelazi na novu vrijednost varijable vanjske petlje. (U ovom primjeru su kao

brojači iskorištene varijable s istim imenima kao interne varijable).

Drugi način ostvarenja petlje ponavljanja je naredbom *while*. Sintaksa naredbe je slijedeća:

```
while varijabla,
    naredbe;
end
```

Petlja se izvršava sve dok je vrijednost varijable različita od nule, tj. logička jedinica.

5. Funkcije

Funkcije predstavljaju uz operatore bitan dio matlaba, a njihovo mnoštvo razvrstano po različitim toolbox-ovima upravo čini Matlab upotrebljivim i moćnim alatom. Funkcije se prema svom porijeklu mogu svrstati u 3 kategorije:

- interne funkcije,
- funkcije u toolbox-ima,
- funkcije definirane od korisnika.

Porijeklo funkcije može se odrediti naredbom *which* iza koje se upiše ime funkcije. Ako je funkcija interna, matlab to i napiše, dok za funkcije iz toolbox-ova i definirane od korisnika ispiše mjesto na disku gdje je smještena. Za upotrebu funkcija njihovo porijeklo nije bitno, jer se sve pozivaju na sličan način:

ime_funkcije(arg1,arg2,...argn).

Ovisno o funkciji, u zagradi se navodi jedan ili više argumenata, od kojih svaki mora biti ili konstanta ili varijabla definirana u *workspace-u*. Na taj se način mogu pozvati sve funkcije koje spadaju u grupu internih funkcija matlaba ili su spremljene na disku računala u direktoriju koji je naveden u Matlabovom putu pretraživanja ili su smještene u tekućem direktoriju.

Definirani put pretraživanja može se dobiti naredbom **path**.

Ukoliko se želi dodati novi direktorij u put pretraživanja, potrebno je provesti slijedeće naredbe Matlaba:

```
p=path;  
path(p, 'novi_put');
```

Prva naredba postojeći put pretraživanja pridružuje varijabli *p*. Druga naredba definira novi put pretraživanja tako da starom putu iz varijable *p* doda novi put pretraživanja definiran stringom *'novi_put'*. Definirani put pretraživanja ostaje aktivan do izlaska iz matlaba. Ukoliko se taj put želi trajno pohraniti u računalo, potrebno je navedene naredbe dodati u *startup.m* proceduru u direktoriju gdje je instaliran Matlab.

Broj i vrsta raspoloživih funkcija u Matlabu ovisi o broju instaliranih toolbox-ova. No, već i standardni toolboxovi sadrže velik broj funkcija među kojima možemo izdvojiti samo neke kao što su: Elementarne matematičke funkcije, Funkcije za obradu vektora i matrica, Funkcije za obradu stringova i funkcije za rad s polinomima. U ovoj skripti će biti spomenute samo osnovne funkcije, a sintaksu i namjenu ostalih korisnik može saznati iz matlabova helpa.

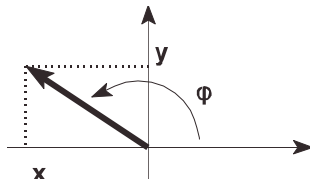
5.1. Elementarne matematičke funkcije

Elementarne matematičke funkcije definirane su na području kompleksnih brojeva, a rezultate daju također u skupu kompleksnih brojeva. Argumenti mogu biti tipa skalara, vektora i matrica, a rezultat je varijabla istog tipa i dimenzija kao i ulazni argument. Prema tome, primjena

funkcije na matricu, kao rezultat će dati matricu istih dimenzija kao što je matrica argumenta čiji su elementi rezultat primjene funkcije na pojedini element matrice argumenta.

5.1.1. Trigonometrijske i ciklometrijske funkcije

Domena i kodomena funkcija je skup kompleksnih brojeva. Sintaksa i opis funkcija dani su tablicom:

Funkcija	Sintaksa	Opis
sin	$y=\sin(x)$	Sinus funkcija kuta u radijanima
cos	$y=\cos(x)$	Cosinus funkcija kuta u radijanima
tan	$y=\tan(x)$	Tangens funkcija kuta u radijanima
asin	$y=\text{asin}(x)$	Arcus sinus funkcija (uz realni argument u području -1 do 1 rezultat je u području $-\pi/2$ do $\pi/2$)
acos	$y=\text{acos}(x)$	Arcus cosinus funkcija (uz realni argument u području -1 do 1 rezultat je u području $-\pi$ do 0)
atan	$y=\text{atan}(x)$	Arcus tangens funkcija (uz realni argument u području $-\infty$ do ∞ rezultat je u području $-\pi/2$ do $\pi/2$)
atan2	$\text{phi}=\text{atan2}(y,x)$	 <p>Arcus Tangens definiran u 4 kvadranta.</p>

5.1.2. Logaritamske i hiperbolne funkcije

Logaritamske i hiperbolne funkcije, zajedno sa sintaksom i opisom dane su tablicom:

Funkcija	Sintaksa	Opis
exp	$y=\exp(x)$	Eksponencijalna funkcija $y = e^x$
log	$y=\log$	Funkcija prirodnog logaritma $y=\ln(x)$
log10	$y=\log_{10}$	Funkcija dekatskog logaritma
sinh	$y= \sinh(x)$	Funkcija sinus hiperbolni
cosh	$y=\cosh(x)$	Funkcija cosinus hiperbolni
tanh	$y=\tanh(x)$	Funkcija tangens hiperbolni
asinh	$y=\text{asinh}(x)$	Inverzna funkcija sinus hiperbolni
acosh	$y=\text{acosh}(x)$	Inverzna funkcija cosinus hiperbolni
atanh	$y=\text{atanh}(x)$	Inverzna funkcija tangens hiperbolni

5.1.3. Ostale funkcije

Među ostalim elementarnim funkcijama mogu se navesti funkcije zaokruživanja, apsolutne vrijednosti, kao i funkcije za određivanje realnog odnosno imaginarnog dijela kompleksnog broja. Opis i sintaksa funkcija prikazani su tablicom:

Funkcija	Sintaksa	Opis
abs	y=abs(x)	Apsolutna vrijednost od argumenta $ x $
sign	y=sign(x)	Signum funkcija
round	y=round(x)	Zaokruživanje prema najbližem cijelom broju
fix	y= fix(x)	Zaokruživanje prema najbližem cijelom broju prema nuli (odbacuju se decimalna mjesta iza decimalne točke broja)
ceil	y=ceil(x)	Zaokruživanje prema najbližem cijelom broju u smjeru $+\infty$
floor	y=floor(x)	Zaokruživanje prema najbližem cijelom broju u smjeru $-\infty$
angle	y=angle(x)	Kut kompleksne varijable u radijanima
real	y=real(x)	Realni dio kompleksne varijable
imag	y=imag(x)	Imaginarni dio kompleksne varijable
conj	y=conj(x)	Konjugirano kompleksna vrijednost argumenta
rem	y=rem(x,y)	Daje ostatak cjelobrojnog dijeljenja varijabli x i y
sqrt	y=sqrt(x)	Kvadratni korjen argumenta

5.2. Funkcije za obradu vektora i matrica

Grupu funkcija za obradu vektora i matrica predstavljaju funkcije čiji su argumenti vektori ili matrice, a kao rezultat daju logičku varijablu, ili vektor ili matricu različitih dimenzija od ulazne matrice, odnosno matricu indeksa elemenata argumenta koji zadovoljavaju dani kriterij. To su funkcije za stvaranje matrica, relacijske i logičke funkcije nad matricama, te funkcije za određivanje različitih veličina karakterističnih za matrice i vektore.

5.2.1. Funkcije za definiranje matrica

Funkcije za definiranje matrica omogućuju definiranje matrice dimenzije $m \times n$ sa svojstvima određenim funkcijom. Za sve funkcije ove skupine vrijedi da mogu imati jedan ili dva skalarna argumenta tipa integer. Ukoliko se radi o funkciji s **jednim argumentom, on određuje**

dimenziju izlazne kvadratne matrice. Ako su funkcije pozvane s **2 argumenta**, oni određuju **broj redaka i broj stupaca izlazne matrice.** Funkcije za definiranje matrica su prikazane tablicom:

Funkcija	Sintaksa	Rezultat funkcije
zeros	y=zeros(m) y=zeros(m,n)	Matrica čiji su svi elementi jednaki nuli.
ones	y=ones(m) y=ones(m,n)	Matrica čiji su svi elementi jednaki jedinici.
eye	y=eye(m) y=eye(m,n)	Jedinična matrica (elementi glavne dijagonale jednaki 1, a ostali elementi jednaki nuli)
rand	y=rand(m) y=rand(m,n)	Matrica slučajnih elemenata u intervalu [-1, 1]
randn	y=rand(m) y=rand(m,n)	Matrica slučajnih elemenata generiranih prema normalnoj razdiobi

5.2.2. Relacijske i logičke funkcije

Relacijske i logičke funkcije za rad s matricama ispituju sadržaj matrica koje su dane kao ulazni argumenti funkcije i vraćaju logičku varijablu ili matricu logičkih varijabli koji imaju vrijednost 1 ako je uvjet ispunjen, odnosno nulu ako nije.

Funkcija	Sintaksa	Opis funkcije
any	y=any(x)	Za vektor (stupac) vraća jedinicu ako je barem jedan element različit od nule. Primijenjena na matricu funkcija daje rezultat za svaku kolonu matrice posebno pa je ukupni rezultat redni vektor.
all	y=all(x)	Za vektor (stupac) vraća jedinicu ako su svi elementi različiti od nule. Primijenjena na matricu funkcija daje rezultat za svaku kolonu matrice posebno pa je ukupni rezultat redni vektor.
find	y=find(log_iz)	Funkcija vraća vektor indeksa kod kojih je logički izraz (<i>log_iz</i>) zadovoljen. Npr. $y=find(x>0)$, daje indekse elemenata vektora x koji su veći od nule. Primijenjena na matricu funkcija vraća dva indeksa za svaki element pa je sintaksa: $[m,n]=find(z>0)$.

Funkcija	Sintaksa	Opis funkcije
isnan isinf finite	y=isnan(x) y=isinf(x) y=finite(x)	Sve tri funkcije daju izlaznu matricu istih dimenzija kao što je matrica argumenta ($size(y)=size(x)$). Funkcije daju jedinicu na mjestima elemenata ulazne matrice koji zadovoljavaju ispitivano svojstvo matrice, a nulu na ostalim mjestima. isnan ==> da li je element ulazne matrice jednak NaN isinf ==> da li je element ulazne matrice jednak Inf finite ==> da li je element ulazne matrice konačan
isempty	y=isempty(x)	Funkcija vraća skalarnu vrijednost 1 ako je matrica (vektor) prazan (<i>ima veličinu 0</i>)
exist	m=exist('ime')	Traži da li postoji entitet s imenom <i>ime</i> i vraća: 0 - ako ne postoji 1 - postoji varijabla u radnom prostoru 2 - postoji m-funkcija u Matlabovom putu traženja 3 - postoji MEX-funkcija u Matlabovom putu traženja 4 - postoji MDL datoteka u putu traženja 5 - interna funkcija matlaba 6 - P funkcija 7 - direktorij

5.2.3. Funkcije za obradu vektora

Funkcije za analizu vektora mogu imati kao argument vektor ili matricu. Ako je argument vektor, rezultat je skalarna veličina proizašla kao obrada vektora. Ako je argument matrica, rezultat je redni vektor čiji svaki element predstavlja rezultat operacije nad odgovarajućim stupcem matrice. Sintaksa i opis funkcija dani su tablicom:

Funkcija	Sintaksa	Rezultat funkcije
min	y=min(x)	Minimum vektora x
max	y=max(x)	Maksimum vektora x
mean	y=mean(x)	Srednja vrijednost vektora x
median	y=median(x)	Median vektora x
std	y=std(x)	Standardna devijacija
cov	y=cov(x)	Kovarijanca vektora
sum	y=sum(x)	Suma elemenata vektora
cumsum	y=cumsum(x)	Kumulativna suma elemenata vektora (dimenzija jednaka ulaznom vektoru)

Funkcija	Sintaksa	Rezultat funkcije
diff	y=diff(x)	Vektor razlike susjednih elemenata ulaznog vektora $y(i)=x(i+1)-x(i)$
prod	y=prod(x)	Produkt elemenata vektora
cumprod	y=cumprod(x)	Kumulativni produkt elemenata vektora (dimenzija jednaka ulaznom vektoru)
sort	y=sort	Vraća vektor istih dimenzija kao ulazni vektor, kojem su elementi sortirani po rastućem redu. Kod ulaznog argumenta tipa matrice rezultat je matrica kojoj su stupci sortirani po rastućem redu.

5.2.4. Funkcije za obradu matrica

Funkcije za obradu matrica kao argument imaju matricu, a kao rezultat daju matricu ili vektor. Sintaksa i opis funkcija dan je tablicom:

Funkcija	Sintaksa	Rezultat funkcije
trace	y=trace(x)	Trag matrice (suma elemenata glavne dijagonale)
rank	y=rank(x)	Rang matrice
det	y=det(x)	Determinanta matrice
eig	y=eig(x)	Vektor karakterističnih vrijednosti kvadratne matrice
poly	y=poly(x)	Karakteristični polinom kvadratne matrice
diag	y=diag(x)	Vektor koji sadrži elemente glavne dijagonale matrice
inv	y=inv(x)	Inverzna matrica kvadratne matrice
rot90	y=rot90(x)	Rotacija matrice za 90 stupnjeva u smjeru obrnutom od kazaljke na satu
fliplr	y=fliplr(x)	Zrcalna zamjena stupaca matrice
flipud	y=flipud(x)	Zrcalna zamjena redaka matrice
triu	y=triu(x)	Gornja trokutasta matrica matrice x
tril	y=tril(x)	Donja trokutasta matrica matrice x

5.3. Funkcije za obradu stringova

String je niz ASCII karaktera pridružen varijabli. U Matlabu string se definira jednostrukim navodnicima na početku i kraju željenog niza znakova.

Primjer:

```
s='Ovo je poruka'
```

U ovom primjeru varijabli *s* je pridružen niz znakova.

Matlab sadrži veliki broj funkcija za rad sa stringovima, a njihov je popis moguće dobiti upisom naredbe *lookfor string*. Za svaku pojedinu funkciju detaljno objašnjenje i sintaksa dobiju se upisom naredbe *help* i imena pojedine funkcije. Ovdje je dan pregled samo nekoliko naredbi za rad sa stringovima. Sintaksa i opis funkcija dani su u tablici:

Funkcija	Sintaksa	Rezultat funkcije
blanks	s=blanks(n)	String od <i>n</i> praznih mjesta (ulazni argument integer)
findstr	x=findstr(s1,s2)	Funkcija traži string <i>s2</i> u stringu <i>s1</i> i vraća indekse svih početaka stringa <i>s2</i> u <i>s1</i> .
isstr	x=isstr(s1)	Vraća logičku jedinicu ako je <i>s1</i> string. Inače vraća nulu.
lower	s=lower(s1)	Sva slova stringa <i>s1</i> pretvara u mala slova i sprema u string <i>s</i> .
upper	s=upper(s1)	Sva slova stringa <i>s1</i> pretvara u velika slova i sprema u string <i>s</i> .
strcat	s=strcat(s1,s2,s3)	Spaja sve stringove navedene kao argumente u jedan string.
num2str	s=num2str(x)	Pretvara numeričku varijablu <i>x</i> u string.

5.4. Funkcije za rad s polinomima

Polinomi se u matlabu prikazuju kao redni vektori kod kojih prvi element predstavlja koeficijent uz najvišu potenciju vektora, dok posljednji koeficijent predstavlja koeficijent uz nultu potenciju vektora.

Prim:

Polinom: $x^4 + 3x^2 + 2x$

u Matlabu se opisuje vektorom:

```
[1 0 3 2 0].
```

Prema tome, broj elemenata vektor mora biti za jedan veći od reda polinoma.

Sintaksa i opis funkcija za rad s polinomima dan je tablicom:

Funkcija	Sintaksa	Rezultat funkcije
roots	y=roots(x)	Vektor koji sadrži korjene polinoma x
poly	x=poly(y)	Inverzna funkcija funkciji <i>roots</i> . Iz vektora korjena polinoma stvara polinom.
conv	x=conv(m,n)	Rezultat je polinom nastao množenjem polinoma <i>m</i> i <i>n</i>
deconv	[q,r] = deconv(b,a)	Rezultat su dva polinoma nastala dijeljenjem polinoma <i>b</i> s polinomom <i>a</i> . Polinom <i>q</i> je rezultat dijeljenja, a polinom <i>r</i> je polinom ostatka.
polyder	y=polyder(x)	Polinom nastao derivacijom polinoma <i>x</i> .
polyfit	p=polyfit(x,y,n)	Računa polinom p(x) n-tog reda za skup ulaznih točaka određenih vektorima x i y metodom najmanjih kvadrata. (Elementi x(i) i y(i) formiraju uređene parove točaka krivulje koja se aproksimira polinomom).

5.5. M funkcije

M funkcije su funkcije definirane od korisnika predviđene za izvođenje pomoću Matlaba. Funkcije se sastoje od standardnog zaglavlja iza kojeg slijede naredbe Matlaba. Funkcije se zapisuju kao ASCII datoteke na disk, a njihovo ime obavezno mora imati **ekstenziju m**.

Ime funkcije treba odgovarati imenu datoteke bez ekstenzije, u koju se funkcija sprema. Prema tome, duljina imena, kao i razlikovanje velikih i malih slova ovisno je o operacijskom sustavu na kojem se Matlab izvodi. Da bi se izbjegla kolizija s postojećim funkcijama Matlaba, korisničkim je funkcijama potrebno davati različita imena od imena internih funkcija i funkcija u toolboxovima. Da bi se funkcija mogla izvoditi, mora biti smještena u tekući direktorij ili u direktorij koji je u putu pretraživanja Matlaba.

Funkcija mora imati slijedeći oblik:

```
function izl_var=ime_funkcije(ul_var_1,ul_var_2, ... ul_var_n)
%komentar koji se ispisuje u helpu

operatori i druge funkcije

izl_var= ....
```

Zaglavlje funkcije počinje propisanom riječju *function* iza koje slijedi ime izlazne varijable ili niza varijabli koje funkcija vraća kao rezultat. Ako vraća više varijabli, tada se umjesto imena jedne

izlazne varijable postavlja definicija izlaznog vektora. Na primjer vektor od 3 izlazne varijable imao bi oblik:

```
[iz1, iz2, iz3].
```

Iza imena izlazne varijable slijedi znak jednakosti i ime funkcije (isto kao ime datoteke u koju se funkcija sprema), a zatim u okruglim zagradama slijedi popis ulaznih varijabli (jedne ili više) koje će služiti kao argumenti funkcije kod njenog poziva. U novom retku slijedi jedan ili više redaka komentara s opisom funkcije. Komentar počinje znakom %. Komentar neposredno iza definicije funkcije prikazuje se kod naredbe help za tu funkciju. Iza toga slijedi niz naredbi i operatora matlaba kojima se na temelju ulaznih varijabli određuje izlazna. Izlaznoj varijabli mora u funkciji biti pridružena vrijednost.

M funkcija se poziva iz komandnog prozora Matlaba ili iz druge funkcije tako da se napiše njeno ime, a u zagradi se navedu potrebne vrijednosti ulaznih funkcija.

6. GRAFIČKE FUNKCIJE MATLABA

Matlab omogućuje grafički prikaz rezultata, a osim toga posjeduje i niz funkcija za opis slike, osi i prikazanih krivulja. Matlabom je moguće crtati dvodimenzionalne i trodimenzionalne grafičke prikaze.

6.1. Dvodimenzionalni grafički prikaz

Za crtanje dvodimenzionalnih grafičkih prikaza koriste se naredbe **plot**, **bar**, **stairs** i **stem**. Naredba *plot* spaja susjedna točke grafičkog prikaza ravnom linijom. *Bar* daje stupčasti prikaz, dok se kod *stairs* prikaza dobije prikaz sličan rubno obrisu stupčastog prikaza (bez tijela štapića točke povezane horizontalnim i vertikalnim linijama). *Stem* način prikaza svaki podatak prikazuje vertikalnom linijom čija dužina odgovara iznosu podatka, a na vrhu linije se nalazi kružić. Sintaksa i opcije svih četiriju naredbi je ista pa će biti prikazana samo za naredbu *plot*.

Sintaksa naredbe je:

plot(x,y,'opcije').

Ukoliko se vektor *x* i string opcija izostave, tada se prikazuju točke vektora *y* u ovisnosti o njihovom rednom broju. Ako su prisutni i vektor *x* i vektor *y*, tada vektori tvore niz uređenih parova točaka. Vektori *x* i *y* moraju imati isti broj redaka. Ako je *y* matrica umjesto vektora, koja ima isti broj redaka kao vektor *x*, naredba *plot* crta po jednu krivulju za svaki stupac vektora *y*.

Opcijama se određuje boja i tip linije, prema slijedećoj tablici:

Oznaka	Boja	Oznaka	Tip linije
y	žuta	-	puna linija
m	ljubičasta	:	točkasta linija
c	svjetlo plava	-.	točka crta
r	crvena	--	isprekidana
g	zelena		
b	plava		
w	bijela		
k	crna		

Ostale opcije moguće je naći primjenom funkcije *help plot*.

Naredba *plot* otvara novi grafički prozor i u njemu crta sliku. Ako je grafički prozor već postojao, slika se crta u njemu.

Crtanje na logaritamskoj umjesto na linearnoj skali postiže se primjenom funkcija **semilogx**, **semilogy** ili **loglog** umjesto funkcije *plot*, uz istu sintaksu.

Funkcija *semilogx* daje logaritamsku os apscise i linearnu ordinatu, funkcija *semilogy* daje prikaz na logaritamskoj ordinati i linearnoj apscisi, dok funkcija *loglog* daje logaritamski prikaz na obje osi.

6.2. Funkcije za uljepšavanje slike

Da bi grafički prikaz bio funkcionalniji za korisnika, u Matlabu postoje funkcije za uljepšavanje i kontrolu grafičkog prikaza. Naredbe su prikazane tablicom.

Naredba	Funkcija
clf	Brisanje slike iz aktivnog grafičkog prozora
grid	Crtanje koordinatne mreže na slici
zoom	Omogućenje povećavanja dijela grafikona pomoću miša
hold	Zadržavanje tekuće slike u aktivnom grafičkom prozoru i omogućenje da novi plot ne briše postojeći crtež nego da budu vidljiva oba crteža.

Sve naredbe iz tablice odnose se na aktivni grafički prozor. Naredbe *grid*, *zoom* i *hold* kod prvog poziva postavljaju svojstvo koje kontroliraju, a kod drugog poziva ga poništavaju. Dodavanjem opcije **on** (*grid on*) iza naredbe, svojstvo se uvijek postavlja, a dodavanjem naredbe **off** svojstvo se uvijek uklanja.

Otvaranje novog grafičkog prozora postiže se naredbom **figure**, dok naredba **figure(broj)** aktivira grafički prozor čiji je broj upotrebljen kao argument.

Crtanje više koordinatnih sustava unutar istog grafičkog prozora postiže se naredbom **subplot**.

Sintaksa naredbe *subplot*:

subplot(n,m,i), plot(...).

Argumenti *n* i *m* određuju broj redaka i broj stupaca u koje će koordinatni sustavi biti posloženi, dok broj *i* određuje broj aktivnog elementa u koji naredna naredba *plot* crta.

Primjer:

```
subplot(2,1,1), plot(x,y)
```

```
subplot(2,1,2), plot(x1,y1)
```

Prva naredba *subplot* određuje da će se koordinatni sustavi poredati u 2 retka i jednom stupcu, i da je aktivan prvi od njih. U taj se koordinatni sustav crta krivulja određena vektorima *x* i *y*. Druga *subplot* naredba određuje da je aktivan drugi koordinatni sustav (koordinatni sustav u drugom retku) i u njega se crta krivulja određena vektorima *x1* i *y1*.

6.3. Funkcije za označavanje slike

Grafičkim prikazima u Matlabu moguće je dodati naslov, oznake osi, tekst unutar grafičkog prikaza i legendu. Za to postoje sljedeće naredbe:

- title('string')** - upisuje specificirani string na mjesto naslova,
- xlabel('string')** - upisuje specificirani string na mjesto oznake *x* osi,
- ylabel('string')** - upisuje specificirani string na mjesto oznake *y* osi,
- zlabel('string')** - upisuje specificirani string na mjesto oznake *z* osi, kod 3D grafičkog prikaza
- gtext('string')** - omogućuje upis teksta na mjesto koje se određuje mišem
- legend('str1','str2','str3')** - upisuje se legendana aktivni grafički prikaz i to tako da se uz tip linije prve krivulje ispisuje string *str1*, uz tip 2. krivulje string *str2* itd.

6.4. Trodimenzionalni grafički prikaz

Za trodimenzionalni prikaz potrebno je definirati dva vektora dimenzija $n \times 1$ i $m \times 1$ i jednu matricu dimenzija $n \times m$. Na taj se način dobije skup točaka određenih uređenim trojkama, koje čine plohu. Crtanje trodimenzionalnih grafičkih prikaza izvodi se naredbom **mesh**.

Sintaksa naredbe je:

mesh(x,y,z).

Rezultat prikaza je površina u trodimenzionalnom koordinatnom sustavu, koja je za svaku kombinaciju točaka iz *x-y* ravnine određenih vektorima *x* i *y* definirana vrijednošću na *z* osi definiranom matricom *z*.

7. Literatura

Engineer's toolkit, The Benjamin/Cummings Publishing Company, Inc. Redwood City, USA 1995.,

Hanselman D, Littlefield B.: *Mastering Matlab - A Comprehensive Tutorial and Reference*, Prentice Hall, Upper Saddle River, 1996.

<http://www.yale.edu/secf/software/matlab/matlab-primer.html>