

**Univerzitet u Beogradu  
Mašinski fakultet**

**Aleksandar S. Cvetković  
Slobodan Lj. Radojević**

# **MATLAB**

# **I**

**Beograd, 2012.**



# Matlab I

prof. dr Aleksandar S. Cvetković  
prof. dr Slobodan Lj. Radojević

Univerzitet u Beogradu  
Mašinski fakultet  
Katedra za Matematiku  
Beograd, 2012. godine

Dr Aleksandar S. Cvetković, vanredni profesor  
Mašinskog fakulteta Univerziteta u Beogradu

Dr Slobodan Lj. Radojević, vanredni profesor  
Mašinskog fakulteta Univerziteta u Beogradu

## **Matlab I**

*Osnovni udžbenik*

I izdanje

Recenzenti:

Dr Ivan Arandelović, vanredni profesor  
Mašinskog fakulteta Univerziteta u Beogradu  
Dr Nenad Cakić, vanredni profesor  
Elektrotehničkog fakulteta Univerziteta u Beogradu

Izdavač:

Univerzitet u Beogradu, Mašinski fakultet  
Kraljice Marije 16, 11120 Beograd 35, Srbija

Za izdavača:

Dekan dr Milorad Milovančević, redovni profesor  
Mašinskog fakulteta Univerziteta u Beogradu

Glavni i odgovorni urednik:

Dr Aleksandar Obradović, redovni profesor  
Mašinskog fakulteta Univerziteta u Beogradu

Odobreno za štampu:

Odlukom Dekana Mašinskog fakulteta, br. odluke 218/12 od 25.10.2012.

Beograd 2012. godine

Tiraž: 600 primeraka

Štampa: Planeta print

ISBN 978-86-7083-771-3

---

Preštampanje, umnožavanje, fotokopiranje  
ili reprodukcija cele knjige ili nekih njenih delova nije dozvoljeno

*Angelini i Veri*



# Sadržaj

<b>1. Uvod u Matlab</b>	<b>15</b>
1.1. Radna površina prozora Matlab-a . . . . .	16
1.1.1. Komandni prozor i interpretator komandi . . . . .	17
1.1.2. Prozor radnog direktorijuma . . . . .	19
1.1.3. Istorija unesenih komandi . . . . .	21
1.1.4. Radni prostor interpretatora komandi . . . . .	22
1.2. Editor m fajlova . . . . .	25
<b>2. Tipovi podataka</b>	<b>27</b>
2.1. Numerički tipovi . . . . .	27
2.1.1. Celobrojni tipovi . . . . .	27
2.1.2. Konverzija osnove brojnog sistema . . . . .	30
2.1.3. Brojevi u pokretnom zarezu . . . . .	31
2.1.4. Kompleksni brojevi . . . . .	34
2.1.5. Specijalne vrednosti standarda IEEE-754 . . . . .	35
2.1.6. Funkcije koje identifikuju numeričke tipove . . . . .	36
2.1.7. Prikazivanje numeričkih vrednosti . . . . .	36
2.1.8. Aritmetičke operacije nad skalarima . . . . .	38
2.1.9. Relacione operacije nad skalarima . . . . .	44
2.1.10. Operacije koje deluju nad bitovima . . . . .	45
2.2. Logički tip podataka . . . . .	47
2.2.1. Logički operatori . . . . .	48
2.2.2. Aritmetičke operacije . . . . .	50
2.2.3. Relacioni operatori . . . . .	50
2.3. Znakovni tip . . . . .	51
2.3.1. Aritmetičke operacije, relacioni i logički operatori . . . . .	53
2.3.2. Funkcije povezane sa znakovnim tipom . . . . .	54

<b>3. Strukture podataka</b>	<b>57</b>
3.1. Matrice . . . . .	57
3.1.1. Kreiranje jednostavnih matrica . . . . .	58
3.1.2. Konkatencija - spajanje matrica . . . . .	60
3.1.3. Kreiranje aritmetičkih nizova . . . . .	62
3.1.4. Kreiranje matrica sa različitim tipovima podataka . . . . .	63
3.1.5. Adresiranje elemenata matrica . . . . .	64
3.1.6. Funkcije za prikupljanje informacija o matricama . . . . .	68
3.1.7. Promena tipa matrice . . . . .	69
3.1.8. Operacije sa matricama . . . . .	71
3.1.9. Sortiranje elemenata matrice . . . . .	82
3.1.10. Operacije nad dijagonalnim matricama . . . . .	85
3.1.11. Skalari, vektori i prazne matrice . . . . .	86
3.1.12. Pune i retke matrice . . . . .	87
3.2. Višedimenzionalni nizovi . . . . .	90
3.2.1. Kreiranje višedimenzionalnih nizova . . . . .	90
3.2.2. Funkcije za kreiranje višedimenzionalnih nizova . . . . .	93
3.2.3. Pristupanje elementima . . . . .	97
3.2.4. Linearno indeksiranje . . . . .	100
3.2.5. Promena tipa . . . . .	101
3.2.6. Operacije sa višedimenzionalnim nizovima . . . . .	103
3.3. Nizovi karaktera . . . . .	105
3.3.1. Komparacija - poređenje stringova . . . . .	108
3.3.2. Pretraga stringa i zamena podstringova . . . . .	109
3.3.3. Izvršavanje stringova . . . . .	110
3.3.4. Konverzija podataka i tip string . . . . .	111
3.4. Datum i vreme . . . . .	116
3.4.1. Merenje vremenskih intervala . . . . .	118
3.5. Nizovi ćelija . . . . .	118
<b>4. Programiranje u Matlab-u</b>	<b>123</b>
4.1. M fajlovi . . . . .	123
4.1.1. Putanje u fajl sistemu . . . . .	123
4.2. Osnovni elementi funkcijskog fajla . . . . .	124
4.2.1. Imena funkcija i m fajlova . . . . .	126
4.3. Naredba uslovnog grananja i petlje . . . . .	127
4.3.1. Naredba if . . . . .	127
4.3.2. Naredba return . . . . .	129
4.3.3. Naredba switch . . . . .	129
4.3.4. Petlja for . . . . .	131
4.3.5. Petlja while . . . . .	134
4.3.6. Naredbe continue i break . . . . .	135
4.3.7. Funkcija error . . . . .	136
4.3.8. Funkcija find . . . . .	136
4.4. Ulazni i izlazni argumenti funkcije . . . . .	137



4.4.1.	Promenljivi broj ulaznih i izlaznih argumenata . . . . .	139
4.5.	Globalne i statičke promenljive . . . . .	141
4.5.1.	Globalne promenljive . . . . .	141
4.5.2.	Statičke promenljive . . . . .	142
4.6.	Podfunkcije . . . . .	143
4.7.	Anonimne funkcije i pokazivači na funkcije . . . . .	144
4.8.	Ugnježđenje funkcije . . . . .	146
4.9.	Privatne funkcije . . . . .	147
4.10.	Skript fajlovi . . . . .	147
4.11.	Elementarna matematika . . . . .	148
4.11.1.	Apsolutna vrednost i znak . . . . .	148
4.11.2.	Polinomi . . . . .	149
4.11.3.	Linearna algebra . . . . .	152
4.11.4.	Trigonometrijske i hiperboličke funkcije . . . . .	153
4.11.5.	Eksponencijalna, logaritamska i stepena funkcija . . . . .	153
4.11.6.	Statistika . . . . .	156
4.11.7.	Konverzija koordinata . . . . .	157
<b>5.</b>	<b>Ulaz i izlaz podataka</b>	<b>159</b>
5.1.	Binarni fajlovi . . . . .	159
5.1.1.	Identifikator fajla . . . . .	160
5.1.2.	Pisanje u fajl i čitanje iz fajla . . . . .	161
5.1.3.	Format mašine . . . . .	166
5.1.4.	Čitanje-upisivanje karaktera . . . . .	169
5.1.5.	Dobijanje informacija o otvorenom fajlu . . . . .	172
5.1.6.	Sekvencijalno čitanje . . . . .	172
5.1.7.	Čitanje-pisanje sa slučajnim pristupom . . . . .	173
5.2.	Tekstualni fajlovi . . . . .	177
5.2.1.	Funkcija <code>fprintf</code> . . . . .	178
5.2.2.	Funkcija <code>fscanf</code> . . . . .	183
5.2.3.	Čitanje redova . . . . .	185
5.2.4.	Značenje simbola <code>'t'</code> pri otvaranju tekstualnog fajla . . . . .	186
5.2.5.	CSV fajlovi . . . . .	187
5.2.6.	DLM fajlovi . . . . .	188
<b>6.</b>	<b>Grafika</b>	<b>191</b>
6.1.	Dvodimenzionalni grafici . . . . .	191
6.1.1.	Kreiranje prozora za crtanje . . . . .	193
6.1.2.	Crtanje više grafika u jednom prozoru . . . . .	193
6.1.3.	Ostali argumenti funkcije <code>plot</code> . . . . .	195
6.1.4.	Razmera . . . . .	198
6.1.5.	Polarni i logaritamski dijagrami . . . . .	198
6.1.6.	Podgrafici . . . . .	201
6.1.7.	Razne vrste grafika . . . . .	203
6.2.	Trodimenzionalni grafici . . . . .	204

6.2.1.	Crtanje linija . . . . .	204
6.2.2.	Crtanje mrežastih grafika . . . . .	205
6.2.3.	Crtanje površi . . . . .	212
6.2.4.	Konturni grafici . . . . .	218
6.3.	Slike . . . . .	219
<b>7.</b>	<b>Simbolička izračunavanja</b>	<b>225</b>
7.1.	Simbolički objekti . . . . .	226
7.1.1.	Simboličke promenljive i izrazi . . . . .	226
7.1.2.	Promenljive sa specifičnim vrednostima . . . . .	228
7.1.3.	Kreiranje matrice simboličkih promenljivih . . . . .	229
7.1.4.	Funkcija sym sa numeričkim argumentom . . . . .	231
7.2.	Manipulacija simboličkim izrazima . . . . .	233
7.2.1.	Pojednostavljivanje izraza . . . . .	233
7.2.2.	Racionalni izrazi . . . . .	235
7.2.3.	Zamene . . . . .	237
7.3.	Analiza . . . . .	240
7.3.1.	Diferenciranje . . . . .	240
7.3.2.	Integracija . . . . .	241
7.3.3.	Nalaženje graničnih vrednosti . . . . .	243
7.3.4.	Sumiranje . . . . .	244
7.3.5.	Taylorov polinom . . . . .	244
7.4.	Rešavanje jednačina . . . . .	245
7.4.1.	Algebarske jednačine . . . . .	245
7.4.2.	Diferencijalne jednačine . . . . .	246
7.5.	Linearna algebra . . . . .	248
7.5.1.	Rad sa matricama . . . . .	248
7.5.2.	Determinanta i inverzna matrica . . . . .	249
7.5.3.	Redukcija i rang . . . . .	250
7.5.4.	Karakteristični polinom, sopstvene vrednosti i sopstveni vektori	250
7.6.	Konverzija simboličkih izraza u Matlab tipove . . . . .	252
7.6.1.	Konverzija u stringove . . . . .	252
7.6.2.	Konverzija u numeričke tipove . . . . .	252
7.7.	Grafika . . . . .	253

# Slike

1.1. Izgled radnog okruženja Matlab-a u operativnom sistemu <i>Microsoft Windows</i> . . . . .	16
1.2. Osnovni izgled editora Matlab-a. . . . .	26
6.1. Grafik funkcije sin konstruisan tačkama $(x_i, \sin x_i)$ , gde je $x_i = 0.1 \cdot i$ za $i = -100 : 100$ , korišćenjem funkcije plot, bez oznaka. . . . .	192
6.2. Grafik funkcije sin konstruisan tačkama $(x_i, \sin x_i)$ , gde je $x_i = 0.1 \cdot i$ za $i = -100 : 100$ , korišćenjem funkcije plot, sa oznakama. . . . .	192
6.3. Grafici funkcija sin i cos prikazani u jedinstvenom prozoru za crtanje slika. . . . .	194
6.4. Grafik funkcije peaks sa argumentom 200. . . . .	194
6.5. Grafici funkcija sin i cos prikazni stilovima $- - +c$ i $^{\wedge}b$ . . . . .	196
6.6. Grafik funkcije $\sin ax$ , $a = 1, 2, 3$ , sa legendom i tekstom. . . . .	196
6.7. Grafik funkcije sin na intervalu $(-10, 10)$ sa razmerom $x_{\min} = -100$ , $x_{\max} = 100$ i $y_{\min} = -5$ , $y_{\max} = 5$ . . . . .	199
6.8. Archimedesova spirala. . . . .	199
6.9. Grafik funkcije $y = 10^x$ u linearnoj razmeri. . . . .	200
6.10. Grafik funkcije $y = 10^x$ sa logaritamskom podelom na $y$ -osi. . . . .	200
6.11. Grafik funkcije $y = x^3$ , $x \in [10, 100]$ , u raznim podelama. . . . .	202
6.12. Razne mogućnosti vizuelizacije podataka. . . . .	202
6.13. Grafik elipse. . . . .	204
6.14. Mrežasti dijagram ravni $z = y$ , sa pravilnim susedstvom tačaka. . . . .	206
6.15. Mrežasti dijagram ravni $z = y$ sa problematičnim susedstvom. . . . .	206
6.16. Grafik funkcije $z = x^2 + y^2$ . . . . .	209
6.17. Grafik elipsoida $x^2 + \frac{y^2}{2} + \frac{z^2}{3} = 1$ . . . . .	209
6.18. Ilustracija koncepta različitih pravaca pogleda na trodimenzionalni grafik. . . . .	211
6.19. Površ ravni $z = y$ , sa pravilnim susedstvom tačaka. . . . .	213

6.20. Površ ravni $z = y$ , sa nepravilnim susedstvom tačaka. . . . .	213
6.21. Elipsoid. . . . .	214
6.22. Cilindar. . . . .	214
6.23. Razni oblici senčenja. . . . .	216
6.24. Razni načini bojenja. . . . .	216
6.25. Grafik funkcije $z = e^{-x^2-y^2} \sin 3\pi x \sin 4\pi y$ , sa konturnim grafikom. . .	219
6.26. Lena. . . . .	222
6.27. Lena u crveno plavoj varijanti. . . . .	222
6.28. Zastava Republike Srbije. . . . .	223
7.1. MuPAD <i>notebook</i> . . . . .	226
7.2. Ilustracija funkcije ezplot. . . . .	254
7.3. Ilustracija funkcije ezpolar. . . . .	254
7.4. Ilustracija funkcije ezplot3. . . . .	256
7.5. Ilustracija funkcije ezsurf. . . . .	256
7.6. Ilustracija funkcija ezmesh. . . . .	258
7.7. Ilustracija funkcije ezcontour. . . . .	258

# Tabele

2.1. Celobrojni tipovi podataka u Matlab-u. . . . .	28
2.2. Funkcije za konvertovanje osnove brojnog sistema. . . . .	30
2.3. Osnovne funkcije za ispitivanje numeričkih tipova. . . . .	36
2.4. Razni tipovi formata ispisa brojeva. . . . .	37
2.5. Osnovne aritmetičke operacije sa celobrojnim tipovima i beskonačnošću gde $N$ može biti 8, 16, 32 ili 64. . . . .	40
2.6. Operacije sa podacima u pokretnom zarezu sa beskonačnostima. . . . .	44
2.7. Relacioni operatori. . . . .	45
2.8. Operacije koje dejstvuju nad bitovima podataka. . . . .	46
2.9. Osnovne funkcije za rad sa logičkim tipom. . . . .	47
2.10. Operacije sa logičkim tipom. . . . .	48
2.11. Osnovne funkcije koje se mogu koristiti sa znakovnim tipom . . . . .	55
3.1. Neke funkcije koje se koriste za kreiranje specijalnih matrica. . . . .	59
3.2. Funkcije koje se koriste za kreiranje matrica iz već postojećih matrica. . . . .	61
3.3. Promena tipova pri kreiranju matrice sa elementima koji su različitih tipova. . . . .	64
3.4. Funkcije koje daju informacije o strukturi matrice. . . . .	68
3.5. Funkcije koje se koriste za rad sa matricama. . . . .	69
3.6. Funkcije koje menjaju oblik matrice. . . . .	70
3.7. Operacije sa matricama. . . . .	72
3.8. Aritmetičke operacije nad elementima matrice. . . . .	77
3.9. Relacioni operator za rad sa matricama. . . . .	78
3.10. Logičke operacije sa matricama. . . . .	80
3.11. Prioriteti operacija u Matlab-u poređani od najvišeg ka najnižem. . . . .	82
3.12. Funkcije koje se koriste pri sortiranju matrica. . . . .	82
3.13. Osnovne funkcije koje se koriste pri radu sa dijagonalnim matricama. . . . .	85
3.14. Funkcije za rad sa retkim matricama. . . . .	88

3.15. Operacije sa višedimenzionalnim nizovima. . . . .	104
3.16. Funkcije koje vrše komparaciju stringova. . . . .	109
3.17. Funkcije koje se koriste za pretraživanje i zamenu. . . . .	109
3.18. Eksplicitna konverzija tipova podataka sa tipom string. . . . .	111
3.19. Neki od mogućih formata datumskog stringa. . . . .	117
4.1. Trigonometrijske i hiperboličke funkcije. . . . .	154
4.2. Eksponencijalna, logaritamska i stepena funkcija. . . . .	155
4.3. Funkcije za transformaciju koordinata tačke. . . . .	157
5.1. Načini pristupa fajlu prilikom otvaranja funkcijom fopen. . . . .	160
5.2. Tipovi podataka, preciznost i broj bitova prilikom pisanja ili čitanja iz fajla. . . . .	162
5.3. Podržani formati mašine prilikom operacija pisanja/čitanja . . . . .	167
5.4. Moguće vrednosti kodovanja karaktera prilikom otvaranja fajla. . . . .	169
5.5. Konvezione niske za funkciju fprintf. . . . .	179
5.6. Vrednosti polja flegova koje se koriste prilikom konverzija. . . . .	181
5.7. Parovi karaktera sa specijalnim značenjem. . . . .	181
5.8. Konverzije niske funkcije funkcije fscanf. . . . .	184
5.9. Mogući atributi funkcije dlmwrite i njihove moguće vrednosti. . . . .	189

# Predgovor

Knjiga *Matlab I* je prvenstveno namenjena studentima Mašinskog fakulteta, Univerziteta u Beogradu, kao udžbenik iz predmeta Računarski alati koji se sluša u drugom semestru prve godine studija. S obzirom na namenu, izvršena je selekcija tema koje knjiga pokriva, ali knjigu mogu koristiti svi koji su zainteresovani za uvodni kurs `Matlab-a`.

Prvo poglavlje sadrži upoznavanje sa korisničkim interfejsom `Matlab-a`. Objasnjeni su pojmovi poput sesija, radni prostor, komandni prozor, radni direktorijum i slično. To će čitaocu - korisniku olakšati savladavanje osnovnih koncepata koji omogućavaju siguran rad u `Matlab-u`. Dodatnu sigurnost će mu doneti i objašnjenje o pribavljanju pomoći o naredbama `Matlab-a`.

Drugo poglavlje objašnjava osnovne tipove podataka koje `Matlab` podržava. Poglavlje je posebno detaljno pisano, a obuhvaćeni su tipovi podataka u pokretnom zarezu, celobrojni, znakovni i logički tip podataka. Detaljno je proučen međusobni odnos tipova podataka.

Treće poglavlje je posvećeno osnovnim strukturama podataka - matricama, zatim višedimenzionim nizovima, nizovima znakova, datumu i nizovima ćelija. Matrice su opisane sa najviše detalja, zatim slede višedimenzionalni nizovi. Nizovi ćelija su urađeni na elementarnom nivou, a strukture nisu pomenute.

Četvrto poglavlje se bavi pisanjem programa u `Matlab-u`. Objasnjeni su koncepti funkcija u `Matlab-u`, naredbe uslovnog grananja, petlje, naredbe za kontrolu izvršenja petlji, globalne i statičke promenljive. Velika pažnja je posvećena argumentima funkcija, i raznim tipovima funkcija. Uveden je i tip podataka pokazivača na funkcije. Posebna sekcija ovog poglavlja sadrži objašnjenje funkcija koje se koriste za osnovna izračunavanja u linearnoj algebri, analizi i statistici.

Peto poglavlje sadrži sve detalje o ulazu i izlazu podataka niskog nivoa. Osobine i rad sa binarnim i tekstualnim fajlovima prikazani su nizom primera. Razmatrano je sekvencijalno čitanje i pisanje u fajl, kao i čitanje i pisanje sa slučajnim pristupom sa različitim konceptima kodiranja karaktera i formata mašina. Obradene su i mogućnosti `Matlab`-a za kreiranje i čitanje CSV i DLM fajlova.

Šesto poglavlje sadrži objašnjenje o postupcima crtanja dvodimenzionalnih i trodimenzionalnih grafika. Veliki broj funkcija za kreiranje grafika je objašnjen i njihova primena je pokazana primerima, tako da ovo poglavlje sadrži i veliki broj slika. Poslednji deo poglavlja posvećen je, na elementarnom nivou, obradi slika u `Matlab`-u.

Sedmo poglavlje, kroz primere, objašnjava simbolička izračunavanja u `Matlab`-u. Ovo poglavlje sadrži gotovo sve što Symbolic Math Toolbox ima da ponudi. Deo poglavlja sadrži i načine crtanja grafika funkcija zadatih simboličkim izrazima.

Na kraju knjige je dat detaljni indeks pojmova i funkcija `Matlab`-a.

Knjiga je prepuna primera. Svi koncepti su prikazani i ilustrovani konkretnim naredbama `Matlab`-a, sa rezultatima njihovog izvršavanja. Preporučujemo aktivno čitanje knjige uz računar, što znači da svaki od primera treba samostalno uneti u računar, proveriti njegov rad i rezultat izvršenja.

Tekst u knjizi objašnjava ponašanje `Matlab`-a i omogućava razumevanje rezultata obrade u `Matlab`-u. Ideja nam je bila da prezentujemo koncepte `Matlab`-a, tako da ih korisnik može razumeti i jednostavno samostalno primeniti. To će pomoći da čitalac - korisnik razume, zatim i savlada osnovne zamisli koje su opredelile današnji `Matlab`. Ovo će omogućiti formiranje kritičke svesti i ojačati kreativnost čitaoca - korisnika u odnosu na predmet proučavanja `Matlab`. Sa te pozicije svaki čitalac - korisnik može sa sigurnošću da upravlja `Matlab`-om u svojoj matičnoj oblasti.

`Matlab` se neprestano razvija. Godišnje se pojavljuju bar dve verzije, što je u skladu sa napretkom Računarskih nauka i Informacionih tehnologija. Svaka želja ili htenje da se da konačna forma teksta knjige ili da se postigne njena bezvremenost je nepotrebna sujeta. Dosadašnji razvoj `Matlab`-a karakteriše nepromenljivost osnovnih koncepata, a kako je u ovoj knjizi insistirano na njihovom razumevanju i formiranju kritičke svesti o njima, to je i njena osnovna vrednost.

Posebno se zahvaljujemo recenzentima na učinjenom naporu da rukopis dobije formu koja je pred čitaocem. Pažljivim čitanjem recenzenti su ukazali na mnogobrojne greške koje su i otklonjene, ali kako to uvek biva, jedan deo je zaostao. Nadamo se da će u nekim budućim izdanjima, i uz pomoć čitalaca ovog udžbenika, i zaostale greške biti otklonjene.



# 1.

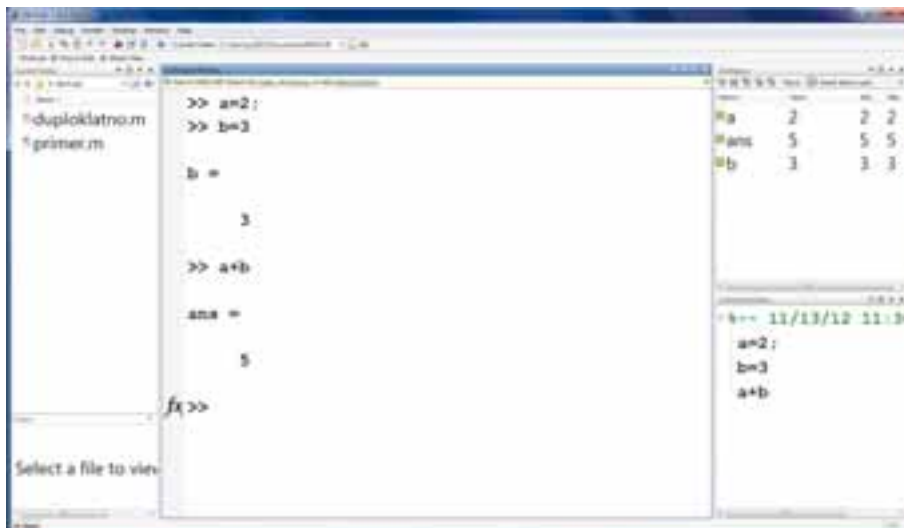
## Uvod u Matlab

Ime programskog paketa `Matlab` je skraćenica od *Matrix Laboratory*, koji je zamišljen kao skup funkcija koje bi pojednostavile i ubrzale rad sa matricama. `Matlab` je samo jedan od brojnih programskih paketa koji se mogu koristiti za izračunavanja. Neki od postojećih su i *Maple*, *Mathematica*, *MathCad* i slični. Nijedan od pobrojanih, ili nekih koji će se tek pojaviti, nije najbolji, već svi imaju prednosti i nedostatke.

`Matlab` je zamišljen kao alat koji sprovodi numerička izračunavanja, tako da je u tom području primene prilično kvalitetan. Preciznije, možemo reći da je `Matlab` vrlo razvijen softverski paket za izvođenje raznih numeričkih izračunavanja. Njegov razvoj tokom godina uključio je i druge mogućnosti, kao što su razni paketi koje `Matlab` podržava. Na primer, trenutna verzija `Matlab`-a uključuje pakete za podršku simboličkim izračunavanjima, za izračunavanja u aerodinamici, bioinformatički, automatičkim, finansijama, za konstruisanje filtera, za primene sa neuronskim mrežama i obradi signala, za numeričko rešavanje parcijalnih diferencijalnih jednačina, za izračunavanja u tehnici visokih frekvencija... Na engleskom se, standardno, ovi paketi nazivaju *toolbox*. `Matlab` podržava pisanje novih funkcija ili izgradnju novih paketa, tako da spisak paketa nije konačan i u budućnosti će sigurno biti proširen.

Programi pisani u `Matlab`-u se interpretiraju, odnosno naredba po naredba se analizira i izvršava. Posledica ovog procesa je sporije izvršenje algoritama u odnosu na iste algoritme implementirane u `C/C++`-u ili nekom drugom programskom jeziku visokog nivoa opšte namene.

`Matlab` je programski paket koji je osmišljen za aplikacije sa matričnim izračunavanjima, i prirodno je da `Matlab` ima slabije performanse, na primer za algoritme za obradu nizova karaktera, u odnosu na `C/C++`, Javu... Ono što `Matlab` čini jedinstvenim, je lakoća implementacije algoritama, jer ima velike biblioteke gotovih implementacija pojedinih algoritama, koje možemo da koristimo u procesu razvoja. Gotove funkcije `Matlab`-a koje se jednostavno koriste za vizuelizaciju predstavljaju izvanredan alat,



Slika 1.1: Izgled radnog okruženja Matlab-a u operativnom sistemu *Microsoft Windows*

a to programski jezici visokog nivoa obično nemaju, ili je bar nemaju na ovakvom nivou.

## 1.1. Radna površina prozora Matlab-a

Pokretanjem *Matlab*-a, u operativnom sistemu *Microsoft Windows*, dobija se prozor koji, sem ako nije izmenjen od strane korisnika, izgleda kao što je prikazano na slici 1.1. Uočljivo je da pojedini elementi radnog prozora služe različitim namenama.

U sredini se nalazi komandni prozor koji prima naredbe koje izvršava interpretator komandi. Gore desno se nalazi prozor koji prikazuje radni prostor interpretatora komandi. Dole desno se nalazi prozor u kome je moguće sagledati hronološki sve komande koje su izdate interpretatoru komandi. Gore levo se nalazi prozor koji prikazuje sadržinu radnog direktorijuma. Ovak prozor ima istu namenu kao *Windows Explorer*, jer prikazuje sadržaj odgovarajućeg direktorijuma u fajl sistemu. Dole levo se nalazi prozor koji prikazuje detalje selektovanog fajla u prozoru radnog direktorijuma.

Pojedini delovi menija se razmatraju u daljem tekstu, a podrazumeva se da postoji meni kao i u svakoj *Microsoft Windows* aplikaciji. Naslovna linija aplikacije, obično pokazuje i verziju *Matlab*-a koja se koristi. Slika 1.1. prikazuje verziju 2010a, što znači da je verzija nastala u prvom polovini 2010. godine.

### 1.1.1. Komandni prozor i interpretator komandi

Komandni prozor, nalazi se u sredini radne površine prozora `Matlab-a`, i najvažniji je prozor radne površine. Služi za unos komandi - naredbi, i u njemu se saopštavaju rezultati interpretacije unesenih komandi.

Primetićete prompt `>>`, oznaka da je interpretator spreman za rad, kojim se korisniku saopštava da se čeka na unos njegove komande. Kad komandni prozor primi fokus, kursor operativnog sistema trepće iza znaka prompta `>>`.

U komandnom prozoru, na slici 1.1, već su unesene neke jednostavne komande. Prva je `a=2;`, dok je druga `b=3`, i kao što vidimo druga komanda proizvodi izlaz:

```
>>b=3
b =
    3
```

Prva komanda ne proizvodi izlaz jer se završava ; karakterom. Ovde se niske `a` i `b` koriste kao promenljive, a u pomenutim naredbama vrši se dodela vrednosti odgovarajućim promenljivama. Dakle, `a=2` je naredba dodeljivanja koja dodeljuje vrednost 2 promenljivoj `a`. Slično, naredba `b=3` dodeljuje vrednost 3 promenljivoj `b`. Napomenimo da u nastavku ove knjige, sve ključne - rezervisane reči `Matlab-a`, kao i naredbe, promenljive, i delove programa pišemo *ovako*. Sledeća naredba je operacija sabiranja, koja sabira promenljive `a` i `b` i ona daje rezultat:

```
>>a+b
ans =
    5
```

Poslednja naredba je samo operacija sabiranja, bez naredbe dodeljivanja. Posledica toga je da se vrednost dodeljuje automatski kreiranoj promenljivoj `ans`, skraćeno od *answer*. Ako naredba, koja se izvršila, nije eksplicitno u sebi sadržala deo za dodelu rezultata izvršenja, automatski se stvara promenljiva `ans` koja prihvata rezultat izvršenja naredbe.

Razlog postojanja promenljive `ans` može se objasniti sledećim primerom. Neka je potrebno izvršiti neka izračunavanja koja su vremenski veoma zahtevna. Pretpostavimo da ste pokrenuli izračunavanja, i posle nekoliko sati primetite da niste nijednoj promenljivoj dodelili rezultat. Posledica ovakvog scenarija je gubitak vremena, jer bi izračunavanja bila izvršena i istog trenutka po završetku rezultati izračunavanja bi nestali, jer nisu dodeljeni nijednoj promenljivoj. Promenljiva `ans` onemogućava ovakav scenario po automatizmu, nezavisno od pažnje korisnika.

Pokrenuta i izvršena naredba ne može biti povučena, ali jednom pokrenuta naredba može biti prekinuta tokom izvršenja. Dok se naredba izvršava dovoljno je pritisnuti kombinaciju tastera `Ctrl-C`. Ova mogućnost je od posebnog značaja u slučaju kad želimo da prekinemo neko vremenski dugo izračunavanje, a da pri tom ne izgubimo

sadržaj prethodnih izračunavanja. Gubitak izračunavanja bi se desio kad bi, na primer, prekinuli izračunavanja prostim prekidom izvršenja procesa `Matlab-a`.

Programski paket `Matlab` se sastoji iz najmanje dva funkcionalna dela. Jedan koji se 'vidi' je zadužen za prikazivanje rezultata obrade i omogućava unos komandi i upravljanje. Drugi deo, koji se 'ne vidi', je deo zadužen za interpretaciju komandi koje komandni prozor prihvata. Rezultat interpretacije komandi je promena stanja odgovarajućih promenljivih, promena stanja u kojoj se nalazi interpretator komandi ili promene načina na koji se prikazuju rezultati obrade. Komanda se interpretira čim je unesena, posle pritiska na taster `Enter` na tastaturi, tako da je nemoguće povući neku naredbu kad je jednom pokrenuto izvršenje, ali, kao što je rečeno, moguće je prekinuti izvršenje naredbe.

Uobičajeno je da se komande unose u komandni prozor jedna po jedna. Međutim, moguće je poslati na izvršenje i sekvence komandi. Na primer, u komandnom prozoru, moguće je uneti niz komandi međusobno razdvojenih ; karakterom. Posledica pritiska tastera `Enter` je izvršavanje sekvence naredbi, naravno, jedna po jedna, redosledom kojim su unete.

```
>>a=20;b=a+3;c=b+3
c =
    26
```

Drugi način je korišćenje karaktera `,` za razdvajanje komandi u nizu. Na primer, prethodna sekvenca izgleda ovako:

```
>>a=20, b=a+3, c=b+3
a =
    20
b =
    23
c =
    26
```

Ovakav unos se razlikuje od prethodnog, jer sve komande proizvode izlaz. Kada bi sekvenca naredbi izgledala `a=20, b=a+3, c=b+3;`, izlaz bi proizvele naredbe `a=20` i `b=a+3`, dok poslednja naredba `c=b+3;` ne bi dala izlaz. Ovakav unos komandi je slabo pregledan i, pri tome su vrlo skromne mogućnosti obrade teksta u komandnom prozoru. Da bi se ovo izbeglo, moguće je uneti sekvencu komandi u tekstualni fajl koristeći editor, recimo *Notepad*, i pozvati ga na izvršenje.

Da bi prethodno ostvarili, neophodno je da fajl u kome se snima niz komandi, ima ekstenziju `m`. Na primer, niz komandi:

```
a = 1;
b = a + 1;
c = b + 1;
c
```

unesemo u fajl `primer.m`, a sam fajl stavimo u radni direktorijum, čiji sadržaj prikazuje prozor radnog direktorijuma `Matlab-a`. U komandnom prozoru, koristeći naredbu `primer`, pozovimo fajl na izvršenje:

```
>>primer
c =
    3
```

Radno okruženje `Matlab-a` omogućava jednostavnije kreiranje fajlova koji sadrže sekvence komandi. U meniju `File` nalazi se podmeni `New` koji omogućava kreiranje ovakvih fajlova u originalnom editoru teksta `Matlab-a`. O ovim mogućnostima će biti više reči kasnije.

Funkcija `clc` briše celokupnu sadržinu komandnog prozora, ali ne proizvodi povlačenje izvršenja komandi koje su već izvršene, već samo, eventualno korisniku nepregledni, komandni prozor dovodi u inicijalno stanje. Efekat naredbe `clc` je sličan naredbama `clear` u *Linux*-u ili `cls` u *Microsoft Windows*-u, kada se koristi komandni režim ovih operativnih sistema.

`Matlab`, kao *Linux*, za razliku od *Microsoft Windows-a*, pravi razliku između velikih i malih slova. Kaže se da je `Matlab` *case sensitive*. To znači da promenljiva `a` nije isto što i promenljiva `A`, niti je `CLC` naredba `Matlab-a`, a `clc` to jeste.

Veliki broj funkcija je bogatstvo `Matlab-a` koje možemo koristiti u izračunavanjima, ali problem nastaje kada je potrebno proučiti format funkcije, kao i opis delovanja ili izvršavanja. Informacije o bilo kojoj funkciji možemo dobiti korišćenjem komande `help`. Na primer, opis funkcije `clc`, dobija se na sledeći način:

```
>>help clc
CLC      Clear command window.
         CLC clears the command window and homes the cursor.

         See also home.

         Reference page in Help browser
         doc clc
```

Poslednji red sadrži link na dokument u kome je detaljnije opisana funkcija `clc` u pretraživaču pomoći. Prozor pretraživača pomoći može se dobiti i iz menija `help` u stavci `help product`.

### 1.1.2. Prozor radnog direktorijuma

Prozor radnog direktorijuma se nalazi gore levo u prozoru `Matlab-a`. Pomoću njega možemo posmatrati sadržinu pojedinih direktorijuma na disku ili lokalno mapiranih diskova i lokacija na mreži. Posebno je važna činjenica da prikazani direktorijum, prikazan u prozoru radnog direktorijuma, interpretator pretražuje i pokušava da pronađe

komandu koja je kreirana od strane korisnika.

Postoji radni direktorijum `Matlab`-a, nazovimo ga *osnovni radni direktorijum*, čiji je sadržaj, uglavnom, prikazan prilikom prvog startovanja `Matlab`-a. On se uvek pretražuje kada se pokušava naći korisnički definisana komanda. Preciznije, ako prozor radnog direktorijuma pokazuje neki direktorijum, različit od osnovnog radnog direktorijuma, onda se prvo pretražuje njegova sadržina pa tek onda sadržina osnovnog radnog direktorijuma. Početnicima ova činjenica može predstavljati problem, zato navodimo jedan primer.

Fajl `primer1.m` postavimo u osnovni radni direktorijum, a `primer2.m` u neki direktorijum - radni direktorijum. Ako je prozor radnog direktorijuma pozicioniran na osnovni radni direktorijum, i ako pokušamo da izvršimo naredbu `primer2`, dobićemo poruku o grešci. Jednostavno, `Matlab` neće pronaći fajl `primer2.m`. Naravno, izvršavanje `primer1.m` je uspešno. Pozicioniranje prozora radnog direktorijuma tamo gde se nalazi fajl `primer2.m` omogućava izvršavanje oba fajla.

Ovakvo pretraživanje, ima uticaj i na to koji od dva istoimena fajla će biti pozvan na izvršenje. Na primer, neka prozor radnog direktorijuma prikazuje sadržinu direktorijuma, nazovimo ga `dir_x`, u kome se nalazi fajl `primer.m` sa sadržinom:

```
a=2
```

dok u osnovnom radnom direktorijumu takođe postoji `primer.m` sa sadržinom:

```
a=3
```

Tada, u komandnom prozoru pokretanje izvršavanja fajla `primer` daje:

```
>>primer  
a =  
    2
```

Ako prozor radnog direktorijuma pokazuje sadržaj direktorijuma različitog od `dir_x`, onda prethodna naredba proizvodi sledeći izlaz:

```
>>primer  
a =  
    3
```

Kroz fajl sistem na disku se možemo kretati funkcijom `cd`. Naredba `cd` bez argumentata pokazuje sadržaj radnog direktorijum, dok `cd ..` pomera radni direktorijum direktorijum iznad. Pomeranje radnog direktorijuma na unapred zadatu lokaciju u fajl sistemu obezbeđuje naredba `cd(direktorijum_na_disku)`, gde je lokacija zadata argumentom `direktorijum_na_disku`. U ovom slučaju je `direktorijum_na_disku` tipa `string`. O tipu `string` će biti više reči u trećoj glavi. Moguće je pisati i direktno `cd direktorijum_na_disku`, u ovom slučaju `direktorijum_na_disku` nije tipa

`string`, već predstavlja nisku. Naredba `cd` je slična naredbi `cd` u *Microsoft Windows*-u i *Linux*-u. Sadržaj prozora radnog direktorijuma sinhronizovan je sa promenama koje unosimo kroz komandni prozor. To znači, da upotreba naredbe `cd` u komandnom prozoru menja direktorijum koji prikazuje prozor radnog direktorijuma. Da bi se promenio disk, na primer, neka je potrebno preći na `d` disk, dovoljno je otkucati `cd('d:')`. Ekvivalentna naredba bez upotrebe tipa `string` bi bila `cd d:`.

Naredbe `dir` i `ls` prikazuju sadržaj radnog direktorijuma. Naredba `mkdir` se koristi za kreiranje poddirektorijuma u direktorijumu u kojem se trenutno nalazimo. Brisanje nekog fajla na disku je moguće naredbom `delete`, a `recycle` služi za postavljanje stanja radnog okruženja Matlab-a u stanje u kome se fajl ne briše sa diska u potpunosti, nego se ostavlja kopija koja se po potrebi može reciklirati.

Za prikaz sadržaja direktorijuma najčešće se koristi komanda `what` koja je slična komandama `dir` i `ls`, ali prikazuje fajlove koji imaju ekstenzije povezane sa Matlab-om. Sledeći primer ilustruje razliku:

```
>>dir
.
..
a.bat
a.m
>>what
a.m
```

Fajl `a.m` je jedini koji prikazuje komanda `what`, jer ima ekstenziju `m` koja je povezana sa Matlab-om.

### 1.1.3. Istorija unesenih komandi

Prozor dole desno u radnoj površini prozora Matlab-a sadrži spisak svih komandi koje je korisnik uneo. Postojanje liste svih komandi koje su unesene, omogućava da se neka prethodno unesena komanda može pozvati na ponovno izvršenje. U komandnom promptu komandnog prozora, korisnik se može kretati kroz listu unetih komandi koristeći kursorke strelice na gore i na dole na tastaturi. Ova je slično funkcionalnosti koju nude razne vrste radnog okruženja u *Linux*-u ili *Microsoft Windows*-u.

U vezi sa istorijom unesenih komandi je naredba `diary`. Sesija Matlab-a se kreira onoga trenutka kad je startovan Matlab. Moguće je deo sesije, neku sekvencu naredbi, snimiti u tekstualni fajl. Početak sekvence naredbi je određen izvršenjem komande `diary on`, a kraj sekvence naredbi je određen izvršenjem komande `diary off`. Po izvršenju komande `diary on` u radnom direktorijumu kreira se fajl `diary`. Ovaj fajl će sadržati izveštaj o aktivnosti unutar Matlab-a, dok se ne izvrši naredba `diary off`. Tokom snimanja izvršenja sekvence komandi, dakle, između komandi `diary on` i

`diary off`, moguće je izvršiti komandu `diary ime_fajla`, koja preusmerava pisanje izveštaja u fajl `ime_fajla`, počev od trenutka njenog izvršenja. To znači da se dešavanja u `Matlab`-u ne upisuju u fajl `diary`, a upisuju se u fajl `ime_fajla`. Prekid upisivanja je obezbeđen pozivanjem naredbe `diary off`. Zanimljivo je da ponovno pokretanje naredbe `diary on`, ne uništava stari fajl `diary` već se nastavlja pisanje u fajl jednostavnim dodavanjem na prethodni sadržaj. Izvršenje sledeće komande `diary ime_fajla`, bez prethodnog izvršenja `diary on`, kreira fajl `ime_fajla` u radnom direktorijumu i počinje snimanje aktivnosti unutar sesije `Matlab`-a. Komanda `diary off` i u ovom slučaju prekida snimanje u fajl `ime_fajla`.

Po startovanju `Matlab`-a pokreće se sesija `Matlab`-a. Izlaz iz trenutne sesije se dešava onog trenutka kada se prekine izvršenje procesa `Matlab`. Prekidanje procesa `Matlab` možemo izazvati pritiskom na dugme za zatvaranje prozora, kao i za bilo koju aplikaciju koja radi u `Windows` okruženju, ali i izvršenjem komandi `quit` ili `exit`.

#### 1.1.4. Radni prostor interpretatora komandi

Sve promenljive koje održava interpretator nalaze se smeštene u radnom prostoru - *workspace*. `Matlab` stvara promenljivu na sledeći način:

- u komandi mora postojati niska koja zadovoljava uslove da je ime promenljive;
- u radnom prostoru ne postoji promenljiva sa imenom koje je već utvrđeno;
- ne postoji fajl po direktorijumima sa imenom koje je već utvrđeno.

Vizuelno radni prostor je predstavljen prozorom gore desno u radnoj površini prozora `Matlab`-a. Kao što vidimo prozor radnog prostora sadrži imena svih promenljivih koje postoje u radnom prostoru. Za svaku od promenljivih prozor radnog prostora prikazuje još neke dodatne informacije. Na primer, na slici 1.1. prikazane su vrednost i tip matrice koje promenljiva predstavlja. Naše promenljive `a` i `b` su brojevi, pa su predstavljene matricama tipa  $1 \times 1$ .

Informacija o imenima svih promenljivih koje se nalaze u radnom prostoru može se dobiti upotrebom funkcije `who`. Ako odmah po startovanju `Matlab`-a izvršimo sledeću sekvencu naredbi:

```
>>a=2;
>>who
Your variables are:
a
```

dobićemo izlaz funkcije `who` kako je to i prikazano. Detaljnije informacije o promenljivama koje se nalaze u radnom prostoru mogu se dobiti korišćenjem funkcije `whos`.

```
>>a=2;
>>whos a
  Name      Size      Bytes  Class    Attributes
  a         1x1         8    double
```



Kao što vidimo, funkcija `whos` daje informacije o promenljivoj koje sadrže ime, tip matrice koju promenljiva reprezentuje, količinu zauzete memorije u bajtovima, tip - klasu kojoj vrednost promenljive pripada i attribute. Komanda `whos`, bez argumenta, prikazuje sve promenljive u radnom prostoru `Matlab`-a. Prikaz je organizovan slično kako je organizovan u prozoru radnog prostora. Vrednost neke promenljive možemo sagledati korišćenjem funkcije `disp`. Naredba štampa vrednost promenljive bez navođenja njenog imena. U svemu ostalom funkcija `disp` se ponaša kao naredba koja nije zatvorena ; karakterom, ali ne štampa sadržinu prazne matrice.

```
>>disp([])
>>[]
ans =
     []
>>a=3;disp(a)
     3
```

### Funkcije za upravljanje radnim prostorom

Radni prostor možemo osloboditi postojanja svih promenljivih upotrebom funkcije `clear`. Posle izvršenja funkcije `clear`, u radnom prostoru nema promenljivih. Funkcija `clear`, takođe, omogućava brisanje pojedinačne promenljive iz radnog prostora `Matlab`-a. Ako odmah po startovanju `Matlab`-a izvršimo sledeću sekvencu naredbi:

```
>>a=2;
>>who
Your variables are:
a
>>clear a;
>>who
```

dobićemo izlaz kakav je prikazan. Ukoliko funkcija `who` kao rezultat ne prikaže tekst to znači da radni prostor nema u sebi nijednu promenljivu.

Po završetku rada sa `Matlab`-om, dakle, posle završetka rada programa, sve promenljive iz radnog prostora programa nestaju. Prilikom startovanja `Matlab`-a, nema ni jedne promenljive u radnom prostoru, nema čak ni promenljive `ans`. Vrednosti promenljivih između prestanka rada i ponovnog pokretanja `Matlab`-a, mogu se preneti koristeći funkciju `save`. Ova funkcija za argument ima ime fajla u kome će, pored ostalog, biti smeštena imena promenljivih i njihovih vrednosti. Po ponovnom pokretanju `Matlab`-a, funkcijom `load`, može se ponovo restaurirati radni prostor. Funkcija `load` ima za argument ime fajla koji sadrži informacije o sadržini radnog prostora koji se učitava.

Funkcija `save` ne mora da snima ceo radni prostor. Na primer, pretpostavimo da radni prostor `Matlab`-a sadrži promenljive `a`, `b` i `c`. Naredba:

```
>>save wSab a b
```

sačuvaće vrednost promenljivih `a` i `b` u fajl `wSab`, dok vrednost promenljive `c` neće biti sačuvana. Slično, naredba `load` ne mora da učita sve promenljive iz fajla:

```
>>load wSab a
```

učitaće u radni prostor samo vrednost promenljive `a` kao i samu promenljivu `a`, dok promenljiva `b` neće biti učitana.

Funkcija `uisave` ima isti format kao i funkcija `save`. Jedina razlika je u tome je što pokreće grafičko okruženje za snimanje fajla. Funkcije `who` i `whos` mogu se koristiti za prikaz promenljivih koje se nalaze u nekom fajlu koji čuva neki radni prostor. Na primer:

```
>>who -file wSab
```

```
Your variables are:
```

```
a b
```

```
>>whos -file wSab a
```

Name	Size	Bytes	Class	Attributes
a	1x1	8	double	

Potrebno je navesti opciju `-file` da bi funkciji naznačili da radimo sa fajlom, kao i samo ime fajla, u našem slučaju `wSab`. Iza ovih argumenata može slediti spisak promenljivih koje nas interesuju.

## Imena promenljivih i funkcija

Niske koje predstavljaju imena promenljivih su nizovi karaktera, uz napomenu da bilo koja niska ne može biti ime promenljive. Recimo, nema mnogo smisla u radnom prostoru imati promenljive sa nekim od imena `clc`, `clear` ili `who`. Postojanje promenljivih sa ovim imenima je u principu moguće, ali nije preporučljivo. Navedimo primer.

Matlab neće prijaviti grešku ako izvršimo sledeću naredbu: `who=4`, ali ako posle izvršenja ove naredbe, izvršimo `who` dobićemo:

```
>>who
```

```
who =
```

```
4
```

Ovo nije ono što očekujemo od funkcije `who`. Naravno, ovo je posledica toga što smo kreiranjem promenljive `who`, praktično preklapili definiciju funkcije `who`. Razlog za to je rad samog interpretatora komandi Matlab-a. Kad interpretator prepozna nisku u nekoj naredbi, on prvo pokušava da nađe interpretaciju niske u radnom prostoru. Ako ne postoji interpretacija u radnom prostoru, interpretator pretražuje sadržinu direktorijuma koji je prikazan u prozoru radnog direktorijuma, zatim pretražuje osnovni radni direktorijum `i`, na kraju, systemske delove koji su deo instalacije Matlab-a u kojima se nalaze definisane funkcije koje dolaze sa instalacijom. Na primer, moguće je u radnom direktorijumu Matlab-a kreirati fajl `who.m`, sadržine `a=2`. Rezultat izvršenja naredbe `who`, u ovom slučaju, će biti

```
>>who
a =
    2
```

Ako na ovaj način prekrijemo funkciju `whos`, recimo, promenljivom, prilikom izvršenja naredbe `whos a`, interpretator će detektovati grešku, jer smo naveli imena dve promenljive. Vidimo da izraz `whos a` nema smisla ako su obe niske promenljive. Poruka o grešci sadrži detaljan opis da smo definiciju funkcije prekrili definicijom promenljive. Naredba `which` može se koristiti da bi se dobila informacija o pojedinim niskama. Na primer:

```
>>which sin
built-in
(C:\Program Files\Matlab\R2010b\toolbox\Matlab\elfun\@double\sin)
    % double method
>>which if
built-in (C:\Program Files\Matlab\R2010b\toolbox\Matlab\lang\if)
>>which a
a is a variable
```

Znači da je `sin` funkcija koja dolazi sa instalacijom `Matlab`-a, koja se nalazi u direktorijumu koji je prikazan. Preciznije, na pomenutoj lokaciji se ne nalazi stvarna implementacija funkcije `sin`, već samo fajl koji je njena reprezentacija u fajl sistemu. Funkcija `sin` je implementirana u interpretatoru komandi. Ključne reči `Matlab`-a nemoguće je prekrivati definicijama promenljivih, za razliku od funkcija. Spisak svih ključnih reči `Matlab`-a dobija se izvršenjem funkcije `iskeyword`.

Imena promenljivih u `Matlab`-u moraju početi slovom. Pored ovog prvog slova, ime promenljive može sadržati slova, brojeve i `_` karakter. Naredba `isvarname` proverava da li je neka niska ispravno ime promenljive. Na primer:

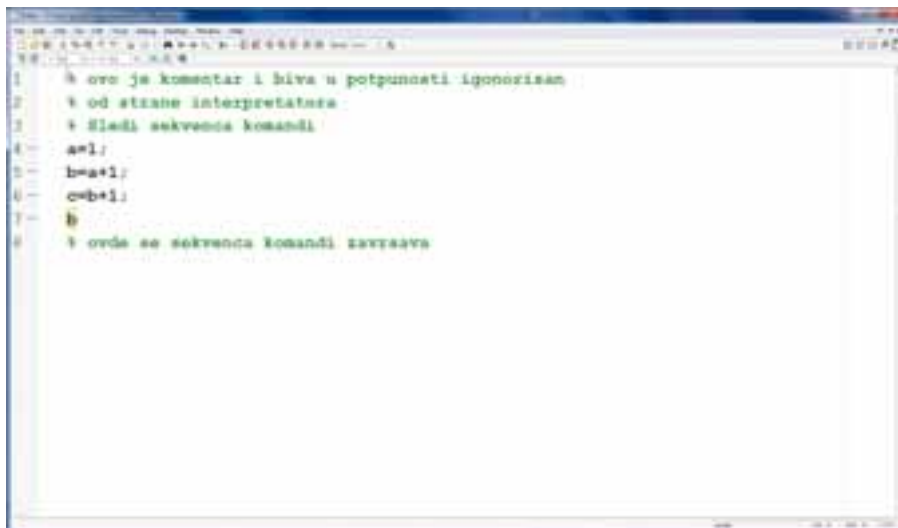
```
>>isvarname masinski_fakultet
ans =
    1
```

znači da je moguće promenljivoj dati ime `masinski_fakultet`.

Imena promenljivih ne mogu biti proizvoljne dužine. Naredba `namelengthmax` daje maksimalnu dozvoljenu dužinu imena promenljive u karakterima. Ova vrednost na tridesetdvobitnoj mašini i verziji `Matlab`-a 2010b iznosi 63.

## 1.2. Editor m fajlova

Zbog prilično ograničenih mogućnosti komandnog prozora za unošenje nizova komandi, `Matlab` poseduje daleko moćniji i primereniji editor teksta. Ovaj editor je



Slika 1.2: Osnovni izgled editora Matlab-a.

poznat kao editor m fajlova, jer fajlovi koje Matlab poziva na izvršenje imaju ekstenziju m.

Komanda `edit ime_fajla` otvara fajl `ime_fajla.m` u direktorijumu koji je prikazan u prozoru radnog direktorijuma, pri čemu ste pitani da li želite da kreirate fajl `ime_fajla.m`. Ako fajl `ime_fajla.m` postoji u direktorijumu koji prikazuje prozor radnog direktorijuma, osnovnom radnom direktorijumu ili u direktorijumima koji dolaze sa instalacijom Matlab-a, onda komanda `edit ime_fajla` otvara u editoru već postojeći fajl.

Postoje različiti tipovi fajlova koje možemo kreirati u ovom editoru teksta. Na primer, moguće je kreirati skript fajlove koji predstavljaju jednostavne nizove komandi, funkcije koje predstavljaju osnovni instrument kreiranja programa u Matlab-u, ili kreirati novu klasu - tip podataka. Slika 1.2. prikazuje osnovni izgled prozora editora. Editujemo fajl `primer.m` i sadržaj fajla je prikazan u radnoj površini prozora. Linije koje počinju karakterom `%` predstavljaju komentare, i njih interpretator jednostavno ignoriše prilikom interpretiranja fajla. Moguće je fajl pozvati na izvršenje direktno iz editora, i to pritiskom tastera `F5`, fajl se šalje na izvršenje. Interpretiranjem našeg skripta, u komandnom prozoru, dobija se izlaz:

```
b =
    2
```

Detaljnije o editoru i njegovom korišćenju za editovanje ostalih tipova fajlova biće reči dalje u knjizi.

Slika 6.18, na pozicijama 3. i 4. sadrži grafike istog elipsoida, ali sada sa pogledima koji se zadaju preko azimuta i elevacije:

- azimut je ugao koji zaklapa projekcija duži koja spaja tačku iz koje se posmatra sa centrom prozora za crtanje sa negativnim delom  $y$ -ose;
- elevacija je ugao koja zaklapa duž koja spaja tačku iz koje se posmatra sa centrom prozora za crtanje sa  $xy$  ravni, a koji se meri u ravni kojoj pripada  $z$ -osa i tačka iz koje se posmatra.

Uglovi azimuta i elevacije zadaju se u stepenima. Pozicija 3, slike 6.18, je pogled sa azimutom od  $90^\circ$  i elevacijom  $45^\circ$ , a pozicija 4 je pogled sa azimutom od  $-90^\circ$  i elevacijom  $-45^\circ$ . Povratak u podrazumevanu poziciju za posmatranje grafika, ostvaruje se pozivom funkcije `view`, i to `view(3)` za trodimenzionalne grafike i `view(2)` za dvodimenzionalne grafike.

### 6.2.3. Crtanje površi

Nekad je potrebno prikazati grafike funkcija i kao jasne površi. Ovakav način prikazivanja je elegantniji i pregledniji od mrežastih grafika. Za ove namene koristimo funkciju `surf`, koja ima iste argumente, njihov oblik, i probleme sa susedstvom, kao i funkcija `mesh`.

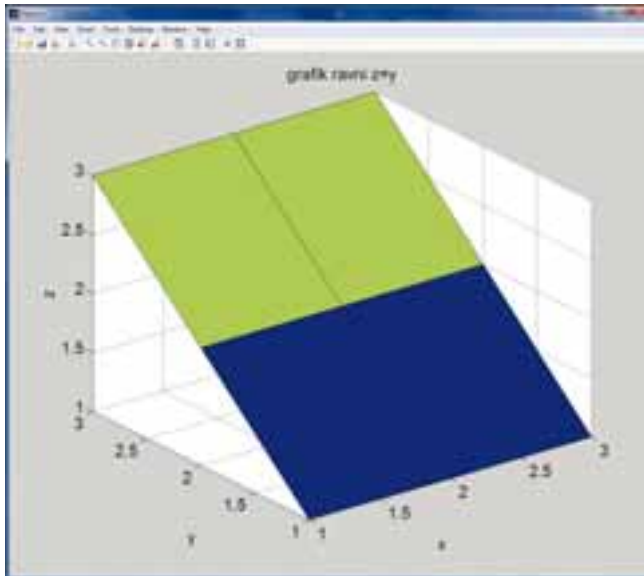
Iskoristimo isti primer kao u prethodnoj sekciji, ravan  $z = y$ ,  $x, y \in [1, 3]$ . Slike 6.19. i 6.20, prikazuju crtež ravni, sa pravilnim susedstvom tačaka, i sa nepravilnim susedstvom tačaka. Sekvenca naredbi korišćena za kreiranje ovih grafika je:

```
>>x=[1 2 3; 1 2 3; 1 2 3];
>>y=[1 1 1; 2 2 2; 3 3 3];
>>z=y;
>>surf(x,y,z);
>>xlabel('x'), ylabel('y'), zlabel('z');
>>title('grafik ravni z=y');
>>figure(2);
>>y=[1 3 1; 2 2 2; 3 1 3];
>>z=y;
>>surf(x,y,z);
>>xlabel('x'), ylabel('y'), zlabel('z');
>>title('grafik ravni z=y, los raspored tacaka');
```

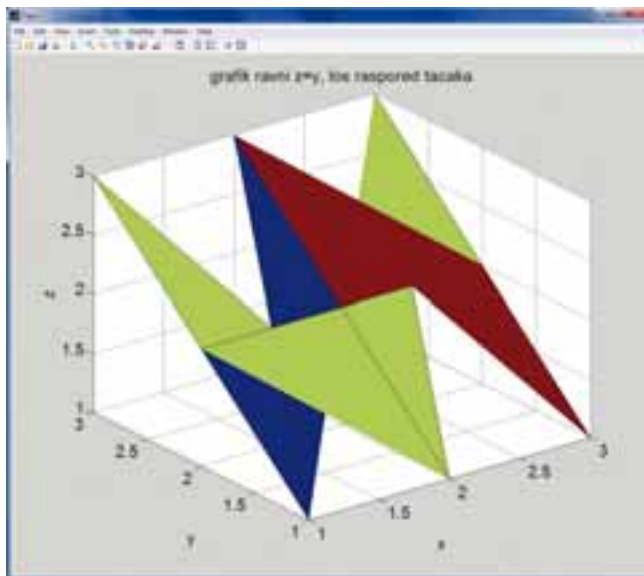
Kako vidimo problem je isti kao u prethodnoj sekciji. Pažljiv izbor susednih tačaka je od izuzetnog značaja.

### Sfera, elipsoid i cilindar

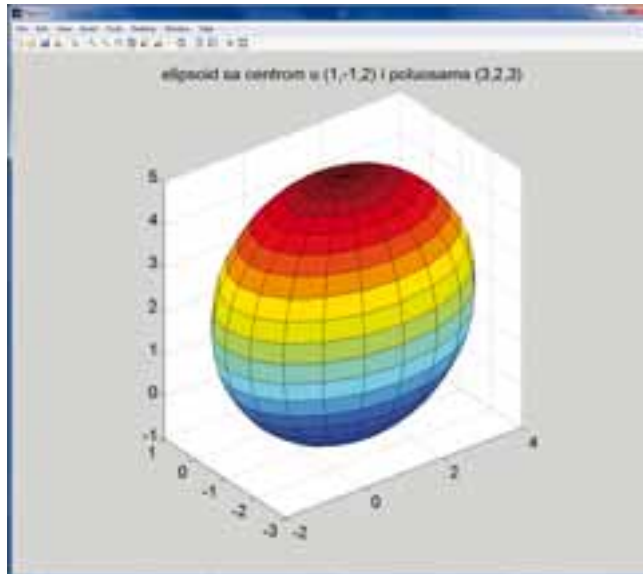
Funkcije `sphere`, `ellipsoid` i `cylinder` generišu meševe kojima se mogu crtati, uz pomoć funkcije `surf`, jedinična sfera, elipsoid sa zadatim centrom i zadatim poluosama i jedinični cilindar. Pozivaju se na sledeći način:



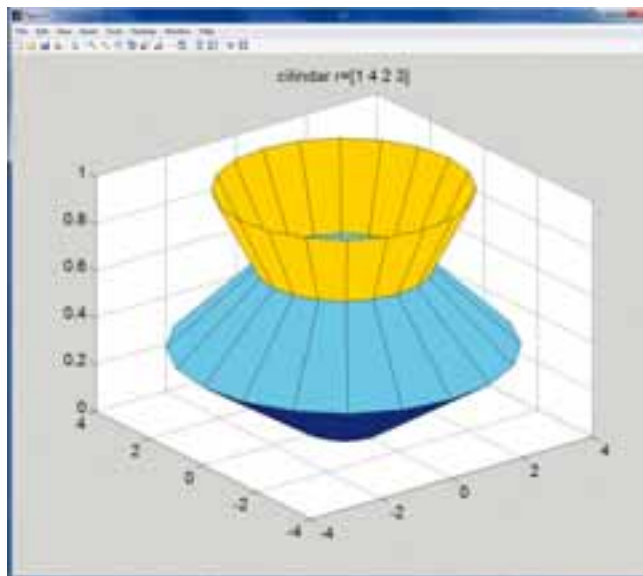
Slika 6.19: Površ ravni  $z = y$ , sa pravilnim susedstvom tačkaka.



Slika 6.20: Površ ravni  $z = y$ , sa nepravilnim susedstvom tačkaka.



Slika 6.21: Elipsoid.



Slika 6.22: Cilindar.

# Bibliografija

*Getting Started with Matlab, Version 2012a*, The MathWorks, Inc., 2012.

*Matlab Programming Fundamentals, Version 2012a*, The MathWorks, Inc., 2012.

J.R. Gilbert, C. Moler, R. Schreiber, *Sparse matrices in Matlab: design and implementation*, SIAM J Matrix Anal App, 33 (1996), 333–356.

B.D. Hahn, B. Hahn, D.T. Valentine, *Essential Matlab for Engineers and Scientists*, Academic Press, 2010.

D.J. Higham, N.J. Higham, *Matlab Guide*, SIAM, 2005.

J.K. Korpela, *Unicode Explained*, O'Reilly Media, 2006.



CIP - Каталогизација у публикацији  
Народна библиотека Србије, Београд

004.42MATLAB(075.8)

**ЦВЕТКОВИЋ, Александар С., 1972-**

Matlab I : [osnovni udžbenik] /  
Aleksandar S. Cvetković, Slobodan Lj.  
Radojević. - 1. izd. - Beograd : Mašinski  
fakultet, 2012 (Beograd : Planeta print). -  
269 str. : ilustr. ; 24 cm

Na nasl. str.: Univerzitet u Beogradu,  
Katedra za Matematiku. - Tiraž 600. -  
Bibliografija: str. 261. - Registar.

ISBN 978-86-7083-771-3

1. Радојевић, Слаободан Љ., 1962- [аутор]  
а) Апликативни програм "MATLAB"

COBISS.SR-ID 195447820



9 788670 837713

