

Katedra za telekomunikacioni saobraćaj i mreže
Statistička teorija telekomunikacija

M A T L A B

- kratko uputstvo za korišćenje -

Oktober 2018.

Osnovni koncept

Matlab je programski paket koji se koristi za rešavanje matematičkih problema i izračunavanja u različitim oblastima nauke i tehnike. Matlab obuhvata programiranje, izračunavanje i vizuelizaciju. Osnovna forma za podatke je niz ili matrica, koja ne zahteva dimenzionisanje. Naziv Matlab potiče od *Matrix laboratory*. Matlab sadrži alate za rešavanje specifičnih problema. Ovi dodatni programski paketi se nazivaju *toolbox*. Toolbox predstavlja kolekciju Matlab funkcija (M-fajlova) koje pružaju mogućnost da se pomoću Matlab-a reše problemi iz različitih oblasti: obrada signala, upravljanje procesima, telekomunikacije, simulacije i dr. Ovo kratko uputstvo bazira se na verziji Matlab 2009a. Nove verzije Matlab-a imaju moderniji grafički korisnički interfejs i širi spektar funkcija, ali osnovna funkcionalnost programa Matlab koja je u ovom u putstvu prezentovana ostala je neizmenjena.

Razvojno okruženje

Nakon startovanja Matlab programa dobija se *Matlab Desktop* – radna površina.

Current Direcory

Prikaz svih fajlova u tekućem direktorijumu

Command Window

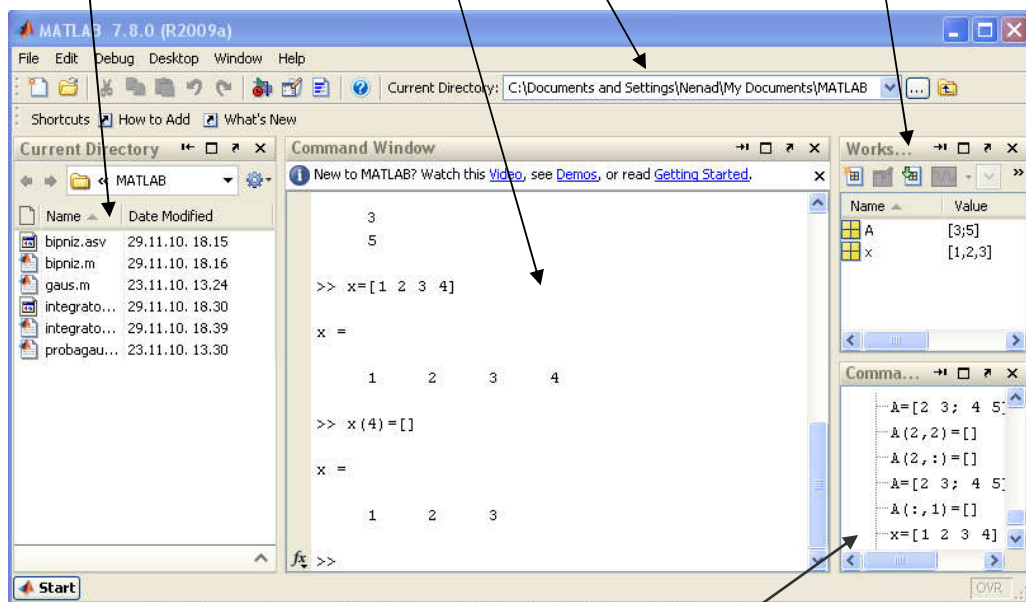
Ovde se unose Matlab komande i funkcija

Current Direcory

Tekući direktorijum

Workspace

Radni prostor prikazuje promenljive koje su prethodno kreirane



Command History Prozor u kome su prikazane prethodno korišćene komande i funkcije

Slika 1. Glavni prozor ili radna površina Matlab-a

Izgled desktopa se može menjati otvaranjem, zatvaranjem, pomeranjem i promenom dimenzije alata koji se na njemu nalaze. Alati se mogu pomerati izvan destopa i vraćati nazad. Standardni (*default*) izgled prozora (slika 1) se uvek može dobiti pozivanjem opcije iz menija: *View* → *Desktop Layout* → *Default*.

Završetak rada: *File* → *Exit MATLAB*.

Delovi Matlaba

Razvojno okruženje - skup alata koji omogućuju korišćenje Matlab funkcija i fajlova. Ovi alati su grafički orjentisani (*desktop*, *command window*, *command history* i drugi).

Matematičke funkcije - ogromna zbirka funkcija počev od najjednostavnijih (*sum*, *sin*) do kompleksnih (*fft*, *ifft*, ...).

Matlab jezik – programski jezik visokog nivoa, sadrži upravljačke funkcije, ulaze/izlaze, objektno orjentisano programiranje.

Grafika – komande visokog nivoa za dvo i trodimenzionalnu grafiku i vizuelizaciju, obradu slike i animacije.

Matlab API (Application program interface) – biblioteka koja omogućava povezivanje C i Fortran programa sa Matlabom.

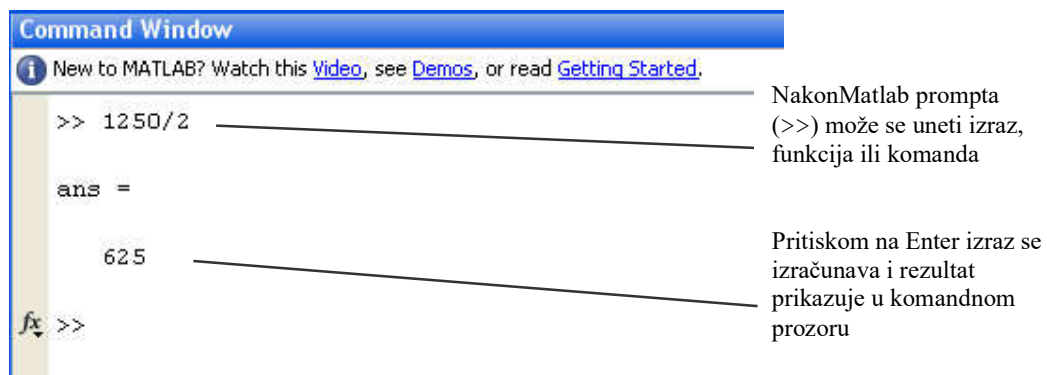
Alati desktopa

U nastavku su razmotreni osnovni elementi radnog okruženja.

Komandni prozor - *Command window*

Komandni prozor je deo radnog okruženja namenjen za interaktivno izvršavanje komandi i pozivanje Matlab funkcija. Nakon Matlab prompta (*>>*) može se uneti proizvoljan izraz, komanda ili ime funkcije (slika 2). Pritiskom na taster Enter komanda se izvršava i rezultat ispisuje u komadnom prozoru.

Command window se ovara/zatvara iz menija: *Desktop* → *Command Window*. Brisanje sadržaja komandnog prozora postiže se opcijom: *Edit* → *Clear Command Window*.

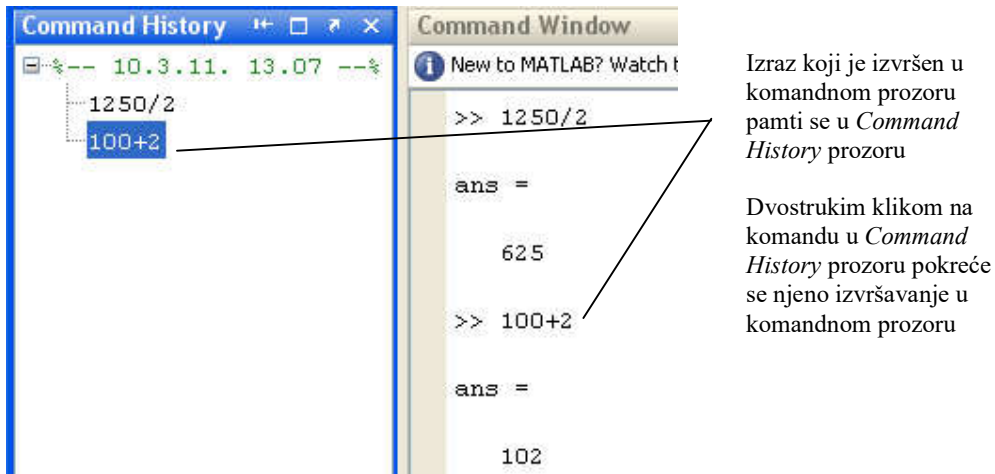


Slika 2. Izgled dela komandnog prozora

Istorija komandi - *Command History*

Linije unete u komandni prozor se memorišu u *command history* prozoru. Preko ovog prozora mogu se podgledati prethodno korišćene funkcije, kopirati ili izvršavati selektovane linije.

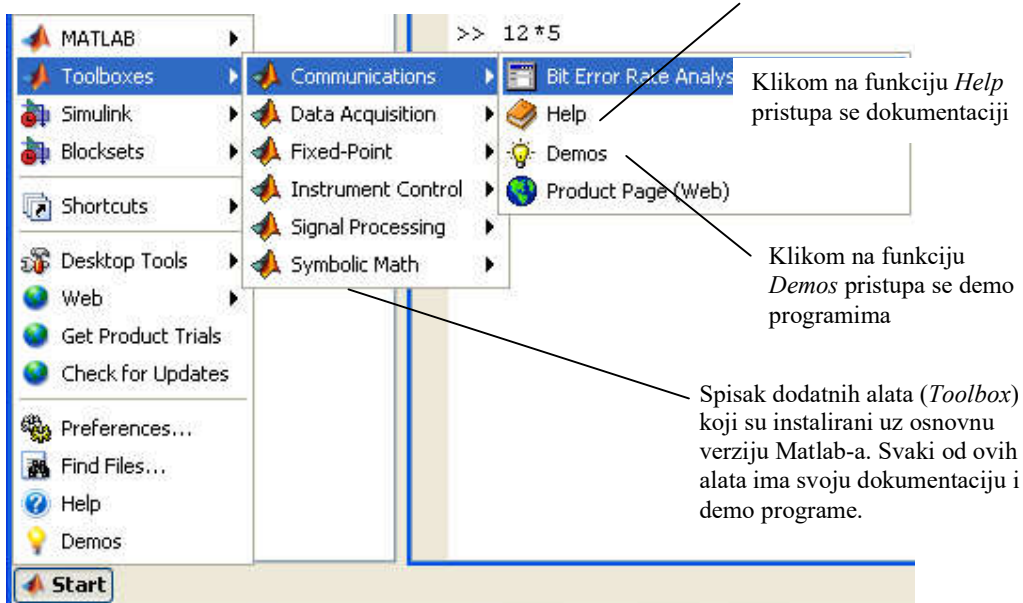
Command History se otvara/zatvara iz menija: *Desktop* → *Command History*.
Brisanje sadržaja ovog prozora postiže se opcijom: *Edit* → *Clear Command History*.



Slika 3. Izgled dela *Command History* prozora

Dodatni alati - *Toolbox*

Start opcija u donjem levom uglu *Desktop*-a omogućava brz pristup alatima, demo programima i dokumentaciji. Sadržaj ovog prozora se može razlikovati u zavisnosti od dodatnih alata (*Toolbox*) koji su instalirani (slika 4).

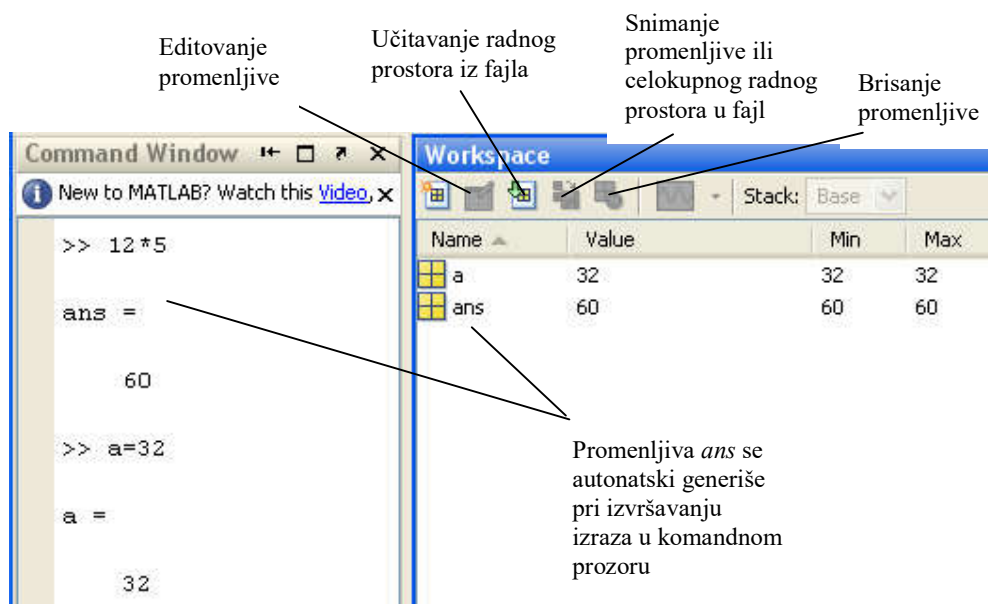


Slika 4. Opcije za brz pristup dokumentaciji i demo programima

Radni prostor - *Workspace*

Radni prostor Matlaba sadrži skup varijabli koje su generisane tokom rada i sačuvane u memoriji (slika 5). Variable iz ovog prostora se mogu editovati dvostrukim klikom. Korišćenjem funkcija, pokretanjem Matlab fajlova dodaju se nove promenljive u ovom prostoru. Radni prostor se može pogledati iz komandnog prozora komandom *who*.

Workspace Browser se otvara/zatvara iz menija: *Desktop* → *Workspace*.
Brisanje sadržaja ovog prozora postiže se opcijom: *Edit* → *Clear Workspace*.



Slika 5. Izgled dela *Workspace* prozora

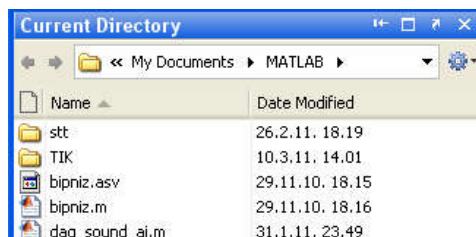
Editor promenljivih - *Variable Editor*

Dvoklikom na varijablu iz *Workspace Browser* prozora dobija se *Variable Editor*. Ovaj prozor je pogodan za editovanje nizova i matrica koje su predstavljene u vidu tabela.

Tekući radni direktorijum - *Current Directory*

Prozor tekućeg radnog direktorijuma (*Current Directory*) omogućava manipulaciju fajlovima u okviru Matlab okruženja. Tekući direktorijum predstavlja standardnu putanju pri snimanju i učitavanju fajlova iz Matlab okruženja.

Current Directory se otvara/zatvara iz menija: *Desktop* → *Current Directory*.



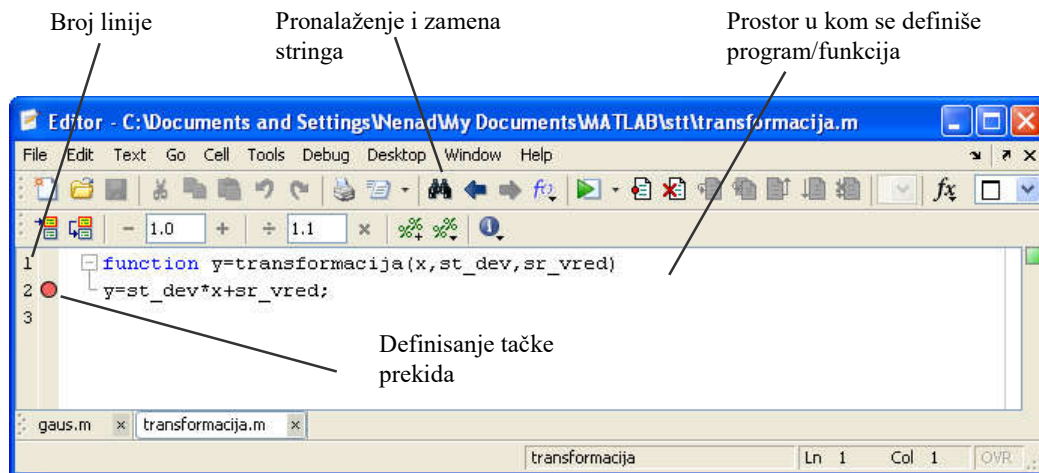
Slika 6. Izgled dela *Current Directory* prozora

Pisanje programa - *Editor/Debugger*

Proširenje funkcionalnosti Matlab paketa može se ostvariti pisanjem korisničkih programa i funkcija (M-fajlova). Za kreiranje i debugovanje M-fajlova može se koristiti *Editor/Debugger* program koji je deo Matlab okruženja. On sadrži funkcije za editovanje teksta i debugovanje koje su neophodne u radu sa M-fajlovima.

Editor/Debugger se ovara iz menija izborom opcije: *File* → *New* → *M-file*.

Editor se može otvoriti i iz komandnog prozora komandom: `>> edit`.



Slika 7. Prozor za editovanje i debugovanje Matlab programa

Za kreiranje M-fajlova može se koristiti bilo koji tekst editor (na primer *Notepad*), a korišćeni editor se može iz *File* menija (*File* → *Preferences*) specificirati kao standardni (*default*) editor.

Sadržaj M-fajla se može pogledati i u komandnom prozoru, ako unesemo:

```
>> type ime_fajla
```

gde je "ime_fajla" ime M-fajla čiji se sadržaj želi prikazati na ekranu.

Matlab kao kalkulator

Posle Matlab prompta (`>>`) u prozoru *Command Window* mogu se unositi promenljive, izračunavati izrazi ili izvršavati komande i funkcije. Na primer,

izračunavanje izraza $5 + 4 \cdot \frac{6^2}{0.3}$ se ostvaruje unošenjem u komandni prozor:

```
>> 5+4*(6^2)/0.3
```

Pritiskom na taster Enter dobija se rezultat:

```
ans =  
485
```

U okviru Matlaba postoje neke unapred definisane često korišćene konstante kao što je konstanta π . Vrednost ove konstante se lako može proveriti unošenjem:

```
>> pi
```

Pritiskom na taster Enter dobija se:

```
ans =  
3.1416
```

Ukoliko je potrebno prikazati rezultate sa većom tačnošću može se koristiti komanda *format*:

```
>> format long  
>> pi  
ans =  
3.14159265358979
```

Detaljno uputstvo za korišćenje ove komande može se naći u dokumentaciji ili pozivom komande *help* u komandnom prozoru:

```
>> help format
```

Osim ugrađenih konstanti, u Matlabu postoje i unapred definisane elementarne matematičke funkcije:

```
>> cos(pi/3)  
ans =  
0.5000
```

Spisak ugrađenih elementarnih funkcija može se dobiti komandom:

```
>> help elfun
```

Tipovi podataka i promenljive

Promenljive se u Matlabu mogu jednostavno definisati korišćenjem operatora dodele (=):

```
>> a=1.234  
a =  
1.2340  
  
>> b=2^15;  
>> c=log2(b)  
c =  
15
```

Prilikom dodele vrednosti nova vrednost promenljive se prikazuje u komandnom prozoru. Ukoliko nije potrebno prikazivati dodeljenu vrednost promenljive *b* na kraju komande dodaje se tačka-zarez (;). Ako treba proveriti koja je trenutna vrednost promenljive *b* dovoljno je uneti:

```
>> b  
b =  
32768
```

Vrednosti promenljivih se pamte u radnom prostoru i mogu se pogledati u okviru *Workspace Browser* prozora i editovati korišćenjem *Array Editor*.

U Matlabu je moguće koristiti različite tipove promenljivih: realne, kompleksne, nizove, matrice i druge. Promenljive nije potrebno deklarirati već se njihov tip i dimenzija određuje implicitno na osnovu rezultata izraza/funkcije kojom se promenljiva kreira.

Kompleksne promenljive se mogu kreirati unošenjem:

```
>> z1=sqrt(-1)
z1 =
    0 + 1.0000i
```

```
>> z2=z1+0.5*(1+2j)
z2 =
    0.5000 + 2.0000i
```

Kao imaginarna jedinica ravnopravno se mogu koristiti oznake i i j .

Najčešće korišćene funkcije pri radu sa kompleksnim brojevima su su unapred definisane, a elementarne matematičke funkcije podržavaju rad sa kompleksnim brojevima:

```
>> abs(z2);      %Moduo
>> angle(z2);   %Argument
>> exp(j*pi/4)
ans =
    0.7071 + 0.7071i
```

Postoje dva osnovna načina za kreiranje nizova u Matlabu:

1. koristeći srednje zagrade []

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

2. koisteći operator :

```
>> x=0:5      %Vektor čije su vrednosti od 0 do 5 sa korakom 1
x =
     0     1     2     3     4     5
>> x=0:0.5:2  %Vektor čije su vrednosti od 0 do 2 sa korakom 0.5
x =
     0     0.5000     1.0000     1.5000     2.0000
```

Matrice se mogu uneti na sledeći način:

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

Elementarne matematičke funkcije podržavaju rad sa nizovima i matricama:

```
>> log10(A) %Logaritam sa osnovom 10
ans =
     0     0.3010     0.4771
    0.6021     0.6990     0.7782
    0.8451     0.9031     0.9542
```

Nizovi karaktera (stringovi) se definišu pomoću apostrofa ('):

```
>> text='ovo je string'
text =
    ovo je string

>> novi_text=['i ' text ' takodje'] %Spajanje više stringova
novi_text =
    i ovo je string takodje
```

Osim ovih osnovnih tipova promenljivih u Matlabu je moguće definisati i složenije tipove: višedimenzione nizove, nizove ćelija (*Cell Array*), strukture i kalase.

Generisanje i grafički prikaz signala

Pretpostavimo da je potrebno nacrtati grafik funkcije $y = 1 + \frac{1}{2}\sin(x)$ u opsegu $0 \leq x \leq 2\pi$. U Matlabu se dati zadatak može rešiti tako što se najpre generiše vektor x čiji će elementi biti realni brojevi u opsegu $[0, 2\pi]$:

```
>> x=0:pi/20:2*pi;
```

a zatim izračuna vrednost funkcije y u tačkama koje su definisane vektorom x :

```
>> y=1+0.5*sin(x);
```

Izračunate vrednosti funkcije y u tačkama koje su definisane vektorom x se jednostavno prikazuju grafički korišćenjem ugrađene Matlab funkcije:

```
>> plot(x,y)
```

Vrednost funkcije y računa se samo u tačkama vektora x , dok se ostale tačke grafika dobijaju linearnom interpolacijom. Vektor x u prethodnom primeru sadrži 41 tačku u opsegu $[0, 2\pi]$. Ako se želi precizniji grafički prikaz potrebno je uzeti veći broj tačaka odnosno generisati vektor x sa više elemenata.

Ukoliko za grafički prikaz podataka interpolacija nije potrebna može se koristiti funkcija *plot* u obliku:

```
>> plot(x,y, '.')
```

Više informacija o upotrebi funkcije *plot* može se naći u dokumentaciji ili pozivom pomoći iz komandnog menija:

```
>> help plot
```

Za prikaz diskretnih funkcija ubičajeno se koristi funkcija:

```
>> stem(x,y)
```

Programski paket Matlab sadži izuzetno bogatu biblioteku funkcija za dvodimenzionalni i trodimenzionalni prikaz podataka. O gračkim mogućnostima Matlabu biće više reči kasnije.

Zadatak 1

Nacrtati grafik funkcije $f(t) = \sin(\omega t) + \frac{1}{2}\cos(2\omega t)$ u opsegu $t \in [0, 0.4]$ sekunde ako je $\omega = 10\pi$ rad/s. Grafički prikazati signal $f(t)$ nakon procesa odabiranja ako je učestanost odabiranja $F_s = 200$ Hz.

Rešenje:

```
>> w=10*pi;
>> Fs=200;
>> t=0:1/Fs:0.4;
>> f=sin(w*t)+0.5*cos(2*w*t);
>> plot(t,f)
>> stem(t,f)
```

Detaljnije upoznavanje sa tipovima podataka

Matrice - osnovne operacije

Osnovni tip podataka u Matlabu su matrice i vektori. Zbog toga treba obratiti posebnu pažnju na alate za rad sa matricama koji su na raspolaganju u ovom programskom paketu.

Generisanje specijalnih matrica

U Matlabu postoji veliki broj unapred definisanih funkcija koje služe za generisanje specijalnih vektora i matrica. Na primer matrica A dimenzija 2×5 čiji su svi elementi nule može se generisati funkcijom:

```
>> A=zeros(2,5)
A =
     0     0     0     0     0
     0     0     0     0     0
```

Slično se mogu generisati i druge vrste matrica:

```
>> B=ones([2 5]) %Matrica dimenzija 2x5 čiji su elementi jedinice
B =
     1     1     1     1     1
     1     1     1     1     1
```

```
>> I=eye(3) %Jedinična matrica dimenzija 3x3
I =
     1     0     0
     0     1     0
     0     0     1
```

```
>> C=randn(3,2) %Matrica dimenzija 3x2. Elementi su slučajni brojevi
%sa Gausovom raspodelom  $\mu_c=0$  i  $\sigma_c=1$ 
C =
    -0.4326    0.2877
    -1.6656   -1.1465
     0.1253    1.1909
```

```
>> D=rand(1,6) %Vektor vrsta dužine 6. Elementi su slučajni brojevi
%sa uniformnom raspodelom u opsegu (0,1)
D =
    0.9501    0.2311    0.6068    0.4860    0.8913    0.7621
```

Spisak osnovnih funkcija za generisanje i manipulaciju matricama može se dobiti pozivom funkcije:

```
>> help elmat
```

Detaljnije informacije o svakoj funkciji dobijaju se unošenjem:

```
>> help ime_funkcije
```

Osnovne funkcije za manipulaciju matricama

Ranije je već rečeno da se spisak osnovnih funkcija za manipulaciju matricama može dobiti pozivom funkcije `help elmat`. Svakako najčešće korišćene jesu

funkcije za određivanje dimenzija nizova (vektora) i matrica. Dimenzije ranije definisane matrice A određuju se pozivom Matlab funkcije:

```
>> size(A)
ans =
     2     5
```

Rezultat poziva funkcije $size(A)$, gde je A matrica dimenzija $m \times n$, je vektor od dva elementa čiji je prvi element broj redova (m) a drugi broj kolona (n) matrice A . Funkcija $size$ se može pozvati u obliku:

```
>> [m n]=size(A)
m =
     2
n =
     5
```

ukoliko se želi da se broj redova i kolona čuvaju u posebnim promenljivima.

Funkcija $size$ može se pozvati i za vektore. Na primer za prethodno uneti vektor D se dobija:

```
>> size(D)
ans =
     1     6
```

Za vektore se, međutim, češće koristi funkcija:

```
>> length(D)
ans =
     6
```

Pristup elementima matrice

Pojedinim elementima matrice pristupa korišćenjem zagrada (). Na primer vrednost elementa u drugom redu i trećoj koloni matrice A dobija se:

```
>> A(2,3)
ans =
     0
```

dok se vrednost ovog elementa može promeniti unošenjem:

```
>> A(2,3)=1
A =
     0     0     0     0     0
     0     0     1     0     0
```

S obzirom da se u Matlabu dimenzije matrice određuju implicitno, dodela vrednosti elementu u trećem redu i šestoj koloni matrice dimenzija 2×5 ne generiše grešku već se matrica proširi potrebnim brojem redova i kolona. Novi elementi koji nisu eksplicitno definisani dobijaju vrednost nula:

```
>> A(3,6)=1
A =
     0     0     0     0     0     0
     0     0     1     0     0     0
     0     0     0     0     0     1
```

Pri radu sa matricama vrlo često se koristi operator : da bi se formirala submatrica koja sadrži samo pojedine kolone ili vrste polazne matrice:

```
>> A(:,3) %Treća kolona
ans =
     0
```

```

1
0

>> A(2,:) %Druga vrsta
ans =
    0    0    1    0    0    0

>> A(2:3,1:2:6) %Druga i treća vrsta; prva, treća i peta kolona
ans =
    0    1    0
    0    0    0

>> A(1:2:3,1:2)=[2 2; 2 2] %Promena vrednosti elemenata u submatrice
A =
    2    2    0    0    0    0
    0    0    1    0    0    0
    2    2    0    0    0    1

>> A(:) '
ans =
Columns 1 through 9
    2    0    2    2    0    2    0    1    0
Columns 10 through 18
    0    0    0    0    0    0    0    0    1

```

Ukoliko je potrebno ukloniti neki element vektora x ili red/kolonu matrice A , to se može uraditi na sledeći način:

```

>> x=[1 2 3 4]
x =
    1    2    3    4

>> x(4)=[] %Brisanje 4. elementa vektora x
x =
    1    2    3

>> A=[2 3; 4 5]
A =
    2    3
    4    5

>> A(2,:)=[] %Brisanje drugog reda matrice A
A =
    2    3

>> A(:,1)=[] %Brisanje prve kolone matrice A
A =
    3
    5

```

Osnovne operacije

Matlab u potpunosti podržava matrice operacije vodeći računa da budu ispunjeni potrebni uslovi za izvođenje date operacije:

```

>> A=rand(2,3)
A =
    0.1763    0.9355    0.4103
    0.4057    0.9169    0.8936

>> B=A' %Transponovanje

```

```

B =
    0.1763    0.4057
    0.9355    0.9169
    0.4103    0.8936

>> C=1+A %Sabiranje sa konstantom
C =
    1.1763    1.9355    1.4103
    1.4057    1.9169    1.8936

>> A+B %Matrice A i B nisu istih dimenzija
??? Error using ==> +
Matrix dimensions must agree.

>> A+C
ans =
    1.3525    2.8709    1.8205
    1.8114    2.8338    2.7873

>> A*C %Matrice A i C nemaju odgovarajuće dimenzije za množenje
??? Error using ==> *
Inner matrix dimensions must agree.

>> A*B
ans =
    1.0745    1.2959
    1.2959    1.8039

>> B*A
ans =
    0.1957    0.5369    0.4349
    0.5369    1.7158    1.2032
    0.4349    1.2032    0.9669

>> D=randn(2)
D =
    1.1892    0.3273
   -0.0376    0.1746

>> D^2 %Kvadriranje matrice. Matrica mora biti kvadratna
ans =
    1.4018    0.4464
   -0.0513    0.0182

>> exp(D)/D %Matrično deljenje
ans =
    2.8446    2.6123
    0.9682    5.0042

>> inv(D) %Inverzna matrica
ans =
    0.7938   -1.4877
    0.1711    5.4055

```

Vrlo često je potrebno da se umesto matričnog množenja (*), deljenja (/) ili stepenovanja (^) izvrši množenje (.*), deljenje ./) ili stepenovanje (.^) matrica član po član. Operatori za rad član po član se dobijaju dodavanjem tačke (.) ispred odgovarajućeg matričnog operatora.

Na primer, pretpostavimo da treba grafički predstaviti funkciju:

$$f(x) = \left(\frac{\sin(x)}{x} \right)^2$$

u opsegu $(-3\pi, 3\pi)$.

Najpre se definiše vektor x :

```
>> x=-3*pi:0.2:3*pi; %Korak 0.2 je uzet proizvoljno
```

Zatim se definiše funkcija $f(x)$:

```
>> f=(sin(x)./x).^2; %Koriste se operatori (./) i (.^)
```

pri čemu se deljenje elemenata vektora $\sin(x)$ odgovarajućim elementima vektora x obavlja operatorom za deljenje član po član ($./$). Na taj način dobija se rezultujući vektor iste dužine kao polazni vektor x . Kvadriranje elementa rezultujućeg vektora obavlja se operatorom za stepenovanje član po član ($.^$).

Rezultat se sada može prikazati grafički:

```
>> plot(x,f)
```

Dotadne informacije o operatorima mogu se dobiti pozivom funkcije za pomoć:

```
>> help ops
```

Zadatak 2

Slučajni proces $S(t)$ pokorava se Gausovoj raspodeli sa srednjom vrednošću 0 i varijansom 1. Generisati vektor S dužine 50000 čiji elementi predstavljaju odbirke datog procesa. Na istom grafiku nacrtati normalizovani histogram vektora S u 100 tačaka i raspodelu procesa $S(t)$. Normalizaciju histograma uraditi tako da se maksimum histograma poklapa sa maksimumom raspodele procesa.

Rešenje:

```
>> S=randn(1,50000);
>> [H X]=hist(S,100); %Histogram u 100 tačaka
>> H=H/(max(H)*sqrt(2*pi)); %Normalizacija histograma
>> bar(X,H,1) %Crtanje histograma
>> s=-5:0.01:5;
>> fs=1/sqrt(2*pi)*exp(-s.^2/2); %Gausova raspodela
>> hold on %Omogućava crtanje više grafika "jedan preko drugog"
>> plot(s,fs,'r') %Grafički prikaz raspodele crvenom bojom
```

Nizivi ćelija - *Cell Array*

Pojedini podaci treba da imaju formu nizova ili matrica ali podaci u matrici ne mogu da budu istog tipa ili podaci nisu iste "dužine". Na primer, ako posmatramo niz stringova primerićemo da se svaki string sastoji iz više karaktera ali dužina stringa nije uvek ista. Ovakav tip podatak može se smestiti u matricu karaktera dimenzija $m \times n$, gde je m broj stringova a n maksimalni broj karaktera. Ali pri tome se zauzima memorijski prostor za svaki string koji je maksimalne dužine n iako dati string može biti značajno kraći.

Niz ćelija je tip podataka sa indeksiranim kontejnerima podataka koji se zovu ćelije. Svaka ćelija može sadržavati bilo koju vrstu podataka. Nizovi ćelija obično sadrže tekstualne podatke, kombinacije teksta i brojeva iz tabelarnih ili tekstualnih datoteka ili numeričkih nizova različitih veličina.

Niz ćelija se kreira na sledeći način:

```
>> student = {'Ime','Petar'; 'Prezime','Petrovic'; 'Smer', 'TS'; 'Br. indeksa', '160007'}
```

```
student =  
    'Ime'          'Petar'  
    'Prezime'     'Petrovic'  
    'Smer'        'TS'  
    'Br. indeksa' '160007'
```

Na ovaj način dobijen je niz ćelija dimenzija 4×2 koji sadrži stringove različite dužine.

Postoje dva načina za upućivanje na elemente ćelijskog polja. Za pristup ćeliji bez obzira na njen tip tj. sadržaj koriste se glatke zagrade, (), kao i kod matrica:

```
>> student(1,2)
```

```
ans =  
    'Petar'
```

```
>> student(:,2)
```

```
ans =  
    'Petar'  
    'Petrovic'  
    'TS'  
    '160007'
```

Za pristup sadržaju ćelije (brojevima tekstu i slično) koriste se vitičaste zagrade, {}:

```
>> student{1,2}
```

```
ans =  
    Petar
```

Kada se pristupi sadržaju ćelije moguće je menjati njene pojedine elemente. Pošto je sadržaj ćelije na koordinati {2,2} string 'Petar', jednostavnom zamenom prva dva slova možemo promeniti ime u 'Mitar':

```
>> student{2,2}(1,1:2)='Mi'
```

```
student =
```

```
    'Ime'          'Petar'  
    'Prezime'     'Mitrovic'  
    'Smer'        'TS'  
    'Br. indeksa' '160007'
```

To se ne može postići glatkim zagradama jer koordinate (2,2) pristupaju ćeliji ali ne njenom sadržaju. Zapravo rezultat indeksiranja nije string nego ćelija pa dalji pristup elementima stringa nije validan:

```
>> student(2,2)(1,1:2)='Mi'
```

```
Error: ()-indexing must appear last in an index expression.
```

Niz ćelija ne mora da sadrži samo tekst već sadržaj ćelije može biti bilo koji tip podatka uključujući matrice ili niz ćelija. Recimo, broj indeksa se ne mora čuvati kao tekstualni podatak već kao celobrojni podatak (matrica dimenzije 1x1):

```
>> student{4,2}=16007
```

```
student =
```

```
    'Ime'          'Petar'  
    'Prezime'     'Mitrovic'  
    'Smer'        'TS'  
    'Br. indeksa' [ 16007]
```

Niz ćelija može imati i veoma složenu strukturu kao u sledećem primeru:

```
>> myCell = {1, 2, 3;  
            'text', rand(5,10,2), {11; 22; 33}}
```

```
myCell =
```

```
    [ 1]    [          2]    [          3]  
    'text' [5x10x2 double] {3x1 cell}
```


Programiranje u Matlabu

Matlab programi predstavljaju skup komandi koje su memorisane u tekstualnom fajlu (M-fajlu). Komande se unose u M-fajl slično kao što bi se unosile u komandni prozor.

Kreiranje programa

Kreiranje i editovanje M-fajlova objašnjeno je u odeljku Editor/Debugger.

Izvršavanje programa

Izvršavanje M-fajlova vrši se na isti način kao izvršavanje funkcija u *Command window*:

Nakon `>>` ispisuje se ime fajla i pritisne taster Enter.

```
>> ime_fajla
```

u komandnom prozoru.

Prekidanje programa vrši se aktiviranjem:

`<Ctrl> + <C>` ili `<Ctrl> + <Break>`.

Pozivanje i editovanje M-fajla ili funkcije iz prozora *Command window* se može izvršiti i na sledeći način. Selektovati ime fajla ili funkcije koja je prethodno pozvana iz *Command window* prozora i desnim tasterom miša kliknuti na selektovani tekst. Na ekranu se pojavljuje meni iz koga treba izabrati opciju *Open selection*. M-fajl će biti otvoren u *Editor-Debuggeru*.

Primer 1

Kreirati M-file pomoću editora:

```
x=-5*pi:0.1:5*pi;  
y=sin(x)./x;  
plot(x,y)
```

Zapamtiti M-fajl pod imenom *program.m* i preći u komandni prozor.

Uneti ime M-fajla:

```
>> program
```

i na ekranu će se pojaviti grafik funkcije $\sin(x)/x$.

Primer 2

Kreirati M-file pomoću editora:

```
function c=mojafunkcija(a,b)  
c=sqrt((a.^2)+(b.^2)); %(.^) se koristi jer a i b mogu biti vektori
```

Zapamtiti M-fajl pod imenom *mojafunkcija.m* i preći u komandni prozor.

Uneti ime funkcije i željene vrednosti argumenata, npr. $a=7.5$ i $b=3.342$ na sledeći način:

```
>> mojafunkcija(7.5,3.342)
```

i na ekranu će se pojaviti rezultat:

```
ans =  
8.2109
```

Script i funkcija

U principu, postoje dve vrste Matlab programa (M-fajlova):

- script, i
- funkcija

Script izvršava niz Matlab komandi pri čemu se ne vrši zadavanje ulaznih argumenata niti se generiše izlazna promenljiva. Primer 1 predstavlja Matlab skript.

Funkcija zahteva argumente i generiše izlaze. U primeru 2 definisana je funkcija koja ima dva argumenta (a i b) i koja generiše izlaz (c).

M-file	
script	function
Ne prima argumente i ne generiše izlaze	Prima argumente i generiše izlaze
Varijable koje kreira unutar M-fajla smeštaju se u <i>Workspace</i> i mogu se pozivati i koristiti za druga izračunavanja	Varijable koje kreira unutar M-fajla su lokalnog karaktera, ne smeštaju se u <i>Workspace</i> i ne mogu se direktno koristiti za druga izračunavanja
Koristi se za automatizaciju niza koraka koji se često izvršavaju ili serije komandi	Koristi se za proširenje Matlab jezika na oblast koja nas zanima

Script

Script M-fajl će biti ilustrovan na primeru u kome se izračunava vrednost ρ (ro) nekoliko trigonometrijskih funkcija ugla θ (teta). Na osnovu izračunatih vrednosti generišu se odgovarajući polarni dijagrami.

Primer 3

Kreirati M-fajl:

```
% M-fajl koji generise laticu cveta, cvet.m
teta = -pi:0.01:pi;    %definisanje teta i koraka izracunavanja

ro=2*sin(5*teta).^2;
polar (teta, ro) %generisanje crteza
pause

ro=cos(10*teta).^3;
polar (teta, ro) %generisanje crteza
pause

ro=sin(teta).^2
polar (teta, ro) %generisanje crteza
pause
```

```
ro=5*cos(3.5*teta).^3;
polar(teta, ro) %generisanje crteza
```

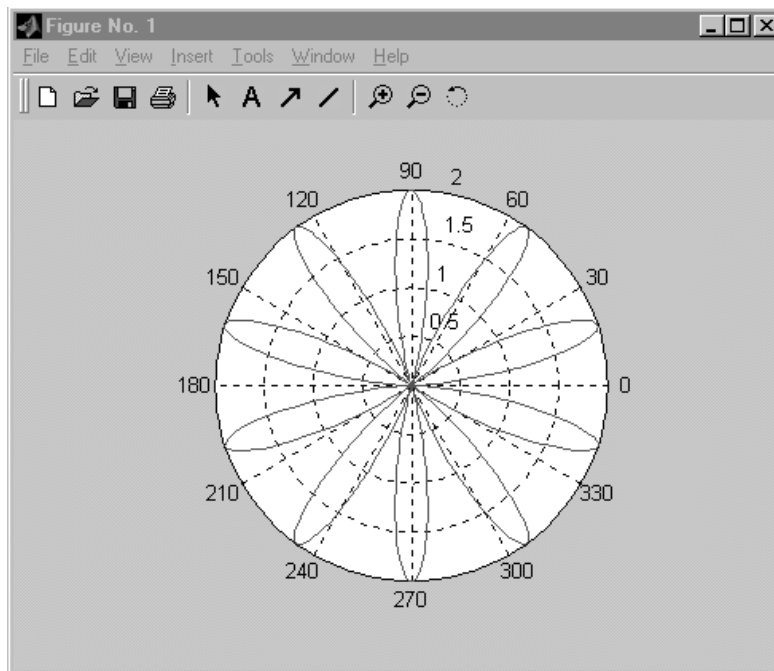
Sačuvati M-fajl pod imenom *cvet.m*. Izvršavanje fajla vrši se iz *Command window* upisom imena fajla i pritiskom na taster Enter:

```
>>cvet
```

Rezultat izvršavanja programa *cvet.m* prikazan je na slici 8. Kreirani script se može izvršiti i iz iz prozora *Editor/Debugger* komandom iz menija *Debug* → *Run*.

Ukupno se generiše četiri crteža. Nakon prvog crteža, usled komande *pause* program se zaustavlja. Prelazak na novi crtež vrši se aktiviranjem ma kog tastera.

Script fajl kreira sve potrebne varijable. Po izvršenju programa promenljive (*teta* i *ro*) ostaju u radnom prostoru. Ove vrednosti mogu se pregledati upisom naziva promenljive u *Command window* i pritiskom na Enter.



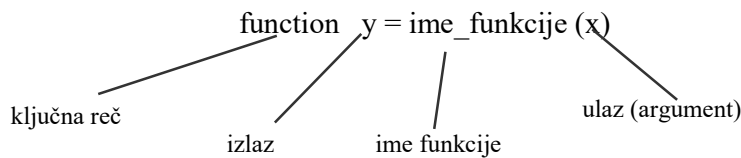
Slika 8. Prvi crtež dobijen izvršavanjem programa *cvet.m*

Funkcija

Kreiranje Matlab funkcije prikazano je na primeru M-fajla *mojafunkcija.m*.

```
function c=mojafunkcija(a,b)%Linija u kojoj se definiše funkcija
%Linije u kojima se upisuje komentar. Ovaj komentar se može dobiti
%pozivom help mojafunkcija iz komandnog prozora
c=sqrt((a.^2)+(b.^2)); %Telo funkcije, sadrži kod koji vrši
                       %izračunavanje i pridružuje
                       %vrednost izlaznom argumentu c
```

U prvoj liniji programa definiše se funkcija ključnom rečju *function*. Na taj način Matlab se informiše da M-fajl sadrži funkciju. Ako se zadaje funkcija jednog argumenta, onda ona ima oblik:



Ako funkcija sadrži više ulaznih i izlaznih promenljivih, onda se ona piše u sledećoj formi:

```
function [x,y,z] = ime_funkcije (argument1, argument2, argument3)
```

Telo funkcije sadrži Matlab kod koji vrši izračunavanje i dodeljuje vrednost izlazu. Telo može da sadrži pozive funkcija, izračunavanja, komentare.

Funkcija se može pozvati iz Matlab *Command window* prozora ili iz drugog M-fajla.

Zadatak 3

Slučajni proces $X(t)$ podleže Gausovoj raspodeli čija je standardna devijacija jednaka 1 i srednja vrednost jednaka 0. Napisati Matlab funkciju: $Y = \text{transformacija}(X, st_dev, sr_vred)$ kojom se ulazni vektor X transformise u izlazni vektor Y koji ima Gausovu raspodelu sa standardnom devijacijom st_dev i srednjom vrednošću sr_vred .

Rešenje:

M-fajl *transformacija.m* koji realizuje traženu funkciju:

```
function Y=transformacija(X,st_dev,sr_vred)
%TRANSFORMACIJA(X,ST_DEV,SR_VRED)
%
% Ova funkcija transformise ulazni vektor X sa standardnom
% devijacijom 1 i srednjm vrednoscu 0 u vektor Y sa standardnom
% devijacijom ST_DEV i srednjm vrednoscu SR_VRED.
Y=st_dev*X+sr_vred;
```

NAPOMENA1: Ime M-fajla treba da bude isto kao i ime funkcije.

NAPOMENA2: Pozvanjem *help transformacija* iz komandnog prozora dobija se pomoć koja predstavlja tekst komentara između linije sa ključnom rečju *function* i tela funkcije.

Zadatak 4

Napisati Matlab script kojim se generiše vektor X dužine 50000. Elementi vektora X pokoravaju se Gausovoj raspodeli čija je standardna devijacija jednaka 1 i srednja vrednost jednaka 0. Izvršiti transformaciju vektora X u vektor Y koji ima Gausovu raspodelu sa standardnom devijacijom jednakom 20 i srednjom vrednošću jednakom 60. Na istoj slici, jedan ispod drugog, prikazati histograme vektora X i Y .

Rešenje:

M-fajl *script.m* koji rešava postavljeni zadatak:

```
X=randn(1,50000);
```

```

Y=transformacija(X,20,60);
subplot(211), hist(X,50)
subplot(212), hist(Y,50)

```

Osnovne informacije o funkcijama *randn* i *subplot* mogu se dobiti sa *help randn*, odnosno *help subplot*.

Programske strukture

Upravljanje tokom izvršavanja programa vrši se korišćenjem programskih struktura. Kao i većina drugih viših programskih jezika i Matlab sadrži programske strukture *if*, *else*, *elseif*, *switch*, *while* i *for*.

if

Komanda omogućuje izvršavanje grupe instrukcija pod određenim logičkim uslovom. Kombinuje se sa komandama *else* i *elseif*.

Način primene komande *if* je:

```

if logički_izraz
    %instrukcije;
    %...
    %...
end

```

| *

Ako je tačan logički izraz Matlab izvršava sve instrukcije između komandi *if* i *end*. U protivnom deo (*) se preskače.

else i *elseif*

Uz komandu *else* ne zadaje se logički uslov. Ova komanda se izvršava ako logički uslov uz prethodnu *if* komandu nije ispunjen.

Uz komandu *elseif* se zadaje logički uslov. Ova komanda se izvršava ako logički uslov uz prethodnu *if* komandu nije ispunjen, a logički uslov koji je zadat uz komandu *elseif* ispunjen. U okviru jednog *if* bloka može se nalaziti više *elseif* komandi.

Primer 4

```

%program deljenje.m
if n<0 %ako je n negativan broj treba prikazati poruku o gresci
    error('Ulazna velicina mora biti pozitivna.');
```

%ako je n pozitivno i parno podeliti ga sa dva

```

elseif rem(n,2)==0
    A=n/2;
else
    A=(n+1)/2; %ako je n pozitivno i neparno dodaj 1 i podeli sa 2
end

```

Dati ime gornjem programu, npr. *deljenje.m* i izvršiti ga.

Funkcionisanje programa možemo proveriti unošenjem neke vrednosti za *n* u *Command window*, na primer:

```
>> n=-5
```

izvršavanje programa *deljenje.m*:

```
>> deljenje
```

na ekranu se dobija poruka

```
??? Ulazna velicina mora biti pozitivna.
```

unošenje nove vrednosti za n

```
>> n=7
```

izvršavanje programa *deljenje.m*

```
>> deljenje
```

pozivanje rezultata A

```
>> A  
A=4
```

switch

Switch je programska struktura koja omogućava višestruko grananje u zavisnosti od vrednosti izraza koji stoji ovu komandu (*switch_izraz*). Način upotrebe komande *switch* je:

```
switch switch_izraz  
case case_izraz  
    %naredbe  
case {case_izraz1, case_izraz2, case_izraz3}  
    %naredbe  
    %...  
otherwise  
    %naredbe  
end
```

Izvršava se grupa naredbi koja sledi iza one ključne reči *case* čija se vrednost izraza (*case_izraz*) poklapa sa vrednošću izraza uz ključnu reč *switch* (*switch_izraz*). Uz ključnu reč *case* može se zadati više izraza uokvirenih vitičastim zagradama {*case_izraz1*, *case_izraz2*, *case_izraz3*}. U tom slučaju izvršava se blok komandi koji sledi nakon *case* ako je *switch_izraz* jednak ma kom od zadatih vrednosti *case_izraz1*, *case_izraz2* ili *case_izraz3*. Ukoliko se vrednost izraza *switch_izraz* ne poklapa ni sa jednom vrednošću izraza *case_izraz*, onda se izvršava grupa naredbi koja sledi iza komande *otherwise*.

while

While petlja se izvršava sve dok je logički izraz uz ovu komandu istinit. Ova struktura se koristi na sledeći način:

```
while logički_izraz  
    %naredbe  
end
```

Primer 5

Segment programa za izračunavanje najmanjeg celog broja n čiji faktorijel je 100 cifreni broj:

```
%program faktorijel.m  
n=1;  
while prod(1:n)<1e100 %proizvod elemenata od 1 do n
```

```

                                %1e100 je 100-cifreni broj
n=n+1;
end

```

Nakon izvršavanja programa *faktorijel.m* u komandnom prozoru se dobija rezultat:

```

>> n
n=70

```

for

For petlja izvršava naredbu ili grupu naredbi predodređeni broj puta. Ova struktura se koristi na sledeći način:

```

for index=start:inkrement:kraj
    %naredbe
end

```

Ukoliko se izostavi, predodređena vrednost inkrementa je 1.

Primer 6

Sledeća petlja se izvršava 5 puta:

```

for i=2:6
    x(i)=2*x(i-1);
end

```

Primer 7

Višestruka petlja:

```

for i=1:m
    for j=1:n
        A(i,j)=1/(i+j-1);
    end
end

```

Indeks *for* petlje može biti matrica, npr. *A* dimenzija $m \times n$. Tada se struktura koristi na sledeći način:

```

for i=A
    %naredbe
end

```

Spoljna petlja izvršava se od 1 do *m* a unutrašnja od 1 do *n*.

break, continue i return

Uz programske strukture *for* i *while* vrlo često se koriste i komande *break* i *continue*.

Komanda *break* prekida izvršavanje *for* ili *while* petlje. Na primer:

```

for i=1:length(x)
    if x(i)==10
        break
    end
end

```

Komanda *continue* nastavlja sa sledećom iteracijom *for* ili *while* petlje.

Komandom *return* prekida se sa izvršavanje i ostvaruje povratak iz tekuće funkcije.

Pomoć za sve pomenute komande može se dobiti pozivom:

```
>> help komanda
```

Unošenje podataka

Tokom izvršavanja M-fajla podaci se mogu uneti na tri načina.

1. Komandom *input*, nakon čega se podatak može uneti preko tastature.
2. Korišćenjem komande *keyboard*.
3. Generisanjem pogodnog grafičkog korisničkog interfejsa (GUI) preko koga se vrši unos podataka.

input

Nakon komande *input* pojavljuje se prompt, nakon čega se čeka na odgovor korisnika. Komanda *input* se koristi na sledeći način:

```
n=input('prompt string')
```

Vrednost koju korisnik unese sa tastature smešta se u promenljivu *n*. Tip promenljive *n* određuje se implicitno kao kada se promenljive unose u komandnom prozoru.

Ako je potrebno da se niz karaktera koji korisnik unosi sa tastature interpretira kao Matlab promenljiva tipa string, koristi se komanda *input* u obliku:

```
str=input('unesi vrednost promenljive str', 's');
```

keyboard

Kada se ova komanda koristi u M-fajlu prekida se izvršavanje fajla i predaje kontrola korisniku. Korisnik može izvršavati proizvoljne komande u komandnom prozoru ili pogledati trenutne vrednosti promenljivih. Komandom *return* nastavlja se sa izvršavanjem M-fajla.

Komanda *keyboard* može biti od koristi prilikom debugovanja programa.

pause

Komanda *pause*, kada se koristi bez argumenta, prekida izvođenje programa sve dok korisnik ne pritisne taster.

Komanda *pause (n)* prekida izvršavanje programa za *n* sekundi.

Zadatak 5

Napisati Matlab funkciju $s = \text{suma_kolona}(A)$ koja vrši sumiranje elemenata po kolonama matrice *A*.

Rešenje:

M-fajl *suma_kolona.m*:

```
function s=sumakolona(A)
```



```

%Izracunava sumu elemenata po kolonama date matrice.
[m n]=size(A);
s=zeros(1,n);
for i=1:m
    s=s+A(i,:);
end

```

Provera rada kreirane funkcije može se izvršiti na sledeći način:

```

>> A=rand(3,5)
A =
    0.9501    0.4860    0.4565    0.4447    0.9218
    0.2311    0.8913    0.0185    0.6154    0.7382
    0.6068    0.7621    0.8214    0.7919    0.1763

>> suma_kolona(A)
ans =
    1.7881    2.1394    1.2964    1.8521    1.8363

>> B=randn(2,3)+j*randn(2,3)
B =
   -0.4326 + 1.1892i    0.1253 + 0.3273i   -1.1465 - 0.1867i
   -1.6656 - 0.0376i    0.2877 + 0.1746i    1.1909 + 0.7258i

>> suma_kolona(B)
ans =
   -2.0981 + 1.1515i    0.4130 + 0.5019i    0.0444 + 0.5391i

```

Zadatak 6

Napisati Matlab funkciju $Y=gaus(N, st_dev, sr_vred)$ koja generiše slučajni vektor sa N elemenata čija je standardna devijacija st_dev i srednja vrednost sr_vred . Omogućiti da se funkcija može pozvati sa tri, dva, jednim ili bez argumenata. Podrazumevane vrednosti argumenata su: $N=1$, $st_dev=1$ i $sr_vred=0$.

Rešenje:

M-fajl *gaus.m*:

```

function Y=gaus(N,st_dev,sr_vr)
%Generise vektor od N elemenata cija je standardna devijacija ST_DEV
% i srednja vrednost SR_VRED
switch (nargin) % nargin je promenljiva koja se automatski kreira.
    % Jednaka je broju argumanata koji su korisцени pri
    % pozivu funkcije gaus
case 0 % bez argumenata
    Y=randn;
case 1 % jedan argumenat - N
    Y=randn(1,N);
case 2 % dva argumenta - N i st_dev
    Y=transformacija(randn(1,N),st_dev,0);
case 3 % sva tri argumenta - N, st_dev i sr_vr
    Y=transformacija(randn(1,N),st_dev,sr_vr);
end

```

Provera rada funkcije:

```

>> gaus
ans =
   -0.5883

>> gaus(5)

```

```

ans =
    2.1832   -0.1364    0.1139    1.0668    0.0593

>> gaus(5,100)
ans =
   -9.5648  -83.2349   29.4411 -133.6182   71.4325

>> gaus(5,100,200)
ans =
   362.3562  130.8224  285.7997  325.4001   40.6270

```

Zadatak 7

Napisati deo programa u Matlabu koji od korisnika zahteva potvrdu za nastavak rada. Ako korisnik odgovori potvrdno program nastavlja sa radom. Ako je odgovor korisnika negativan program završava sa radom. U slučaju neregularnog odgovora potrebno je korisnika upozoriti na grešku i ponovo postaviti pitanje.

Rešenje:

Deo Matlab programa u fajlu *pitanje.m*:

```

while 1
    u=input('Zelite li da se izvršavanje nastavi? ','s');
    switch upper(u)
        case 'DA'
            break
        case 'NE'
            return
        otherwise
            disp('Odgovorite sa DA ili NE.')
    end
end

```

Grafička prezentacija podataka

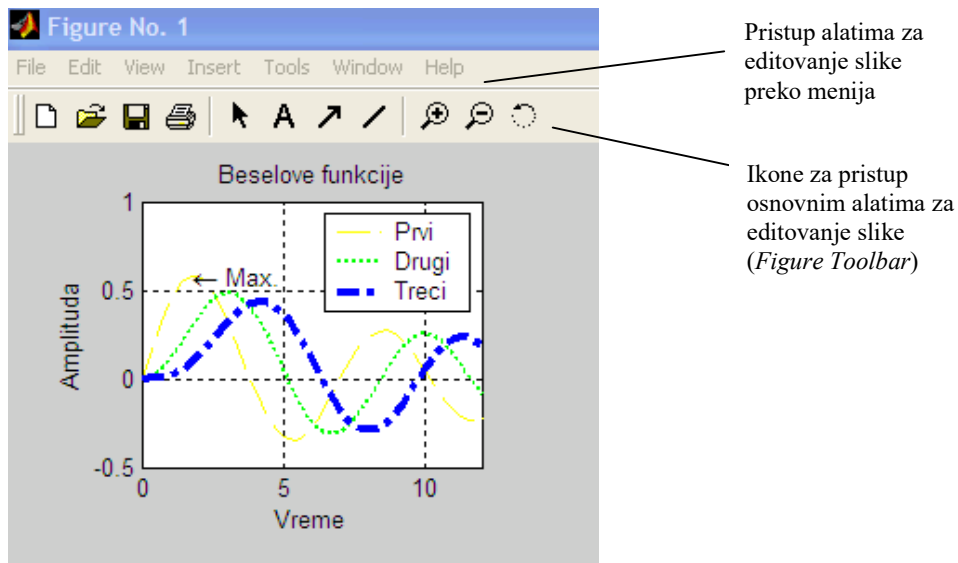
Grafičko predstavljanje podataka i rezultata odvija se kroz sledeće korake.

Tipičan kod

1. Priprema podataka	<pre>x=0:0.2:12; y1=bessel(1,x); y2=bessel(2,x); y3=bessel(3,x);</pre>
2. Izbor prozora i pozicije oblasti u kojoj se crta u okviru prozora	<pre>figure(1) subplot(2,2,1)</pre>
3. Poziv funkcije za crtanje	<pre>h=plot(x,y1,x,y2,x,y3)</pre>
4. Izbor linije i karakteristika markera	<pre>set(h,{'Color'},{'r'; 'g'; 'b'}) set(h(1),'Color','y') set(h,{'LineWidth'}, {1; 2; 3}) set(h,{'LineStyle'},{'--'; ':'; '-.'})</pre>
5. Podešavanje granica osa podele, mreže	<pre>axis([0 12 -0.5 1]) grid on</pre>
6. Opremanje grafa nazivima uz ose, legendom, tekstom	<pre>xlabel('Vreme') ylabel('Amplituda') legend(h, 'Prvi', 'Drugi', 'Treci') title('Beselove funkcije') [y ix]=max(y1); text(x(ix),y,'\leftarrow Max.')</pre>
7. Eksportovanje grafa	<pre>print -depsc -tiff -r200 myplot</pre>

Za najjednostavniju primenu dovoljno je koristiti samo prva tri koraka.

Podešavanje pojedinih karakteristika linija, osa, ispisivanje teksta, eksportovanje grafika u različitim formatima kao i čitav niz drugih funkcija za editovanje slike mogu se izvršavati interaktivno koristeći alate koji su dostupni u prozoru slike (slika 9).



Slika 9. Alati za interaktivno podešavanje slike

Osnovne funkcije za generisanja 2D i 3D grafika

Neke od najčešće korišćenih funkcija za grafičku prezentaciju rezultata prikazane su u tabeli 1. Osim funkcija nabrojanih u tabeli 1, postoji čitav niz drugih funkcija za generisanje 2D i 3D grafika. Detaljna uputstva za korišćenje grafičkih mogućnosti Matlaba mogu se naći u dokumentaciji a neki karakteristični primeri dati su u demo programa.

Tabela 1. Osnovne funkcije za generisanje 2D i 3D grafika

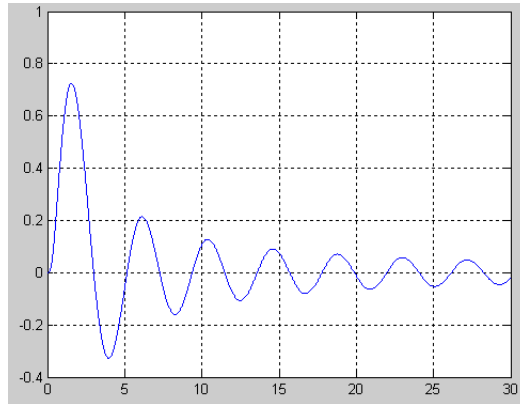
Funkcija	Opis
plot	2D grafika sa linearnom podelom na obe ose
plot3	3D grafika sa linearnom podelom po osama
mesh	3D mrežasta grafika
loglog	2D graf sa logaritamskom podelom na obe ose
semilogx	2D graf sa logaritamskom podelom na x i linearnom podelom na y osi
semilogy	2D graf sa logaritamskom podelom na y i linearnom podelom na x osi
fplot	Grafik funkcije koja je prethodno definisana u M-fajlu

Primer 8

Grafičko predstavljanje funkcije

$$y(t) = \frac{2}{\sqrt{1+(1.5t)^2}} \sin(1.5t + \text{arctg}(-1.5t))$$

```
%program graf.m
t=0:0.01:30;
y=2./sqrt(1+(t*1.5).^2).*sin(1.5*t+atan(-1.5*t));
plot(t,y) %komanda za 2D graf
grid on   %postavljanje mreže
```

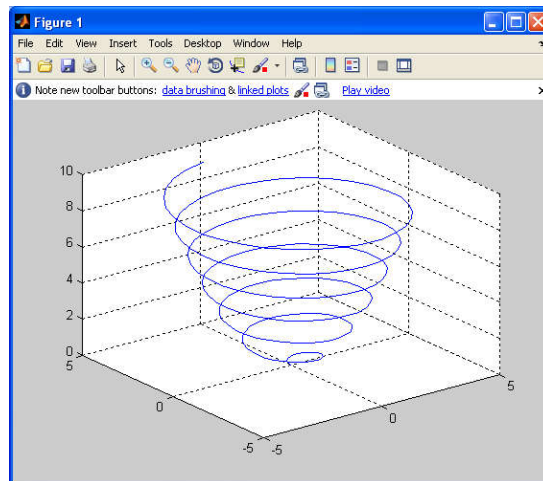


Slika 10. Rezultat dobijen izvršavanjem programa graf.m

Primer 9

Crtanje trodimenzionalnog 3D linijskog grafika: koordinate x , y i z su funkcije zajedničke premenljive t .

```
t=0:0.1:6*pi;
x=sqrt(t).*sin(2*t);
y=sqrt(t).*cos(2*t);
z=0.5*t;           %definisanje funkcije z
plot3(x,y,z)      %crtanje linijske grafike u 3D prostoru
grid on
```

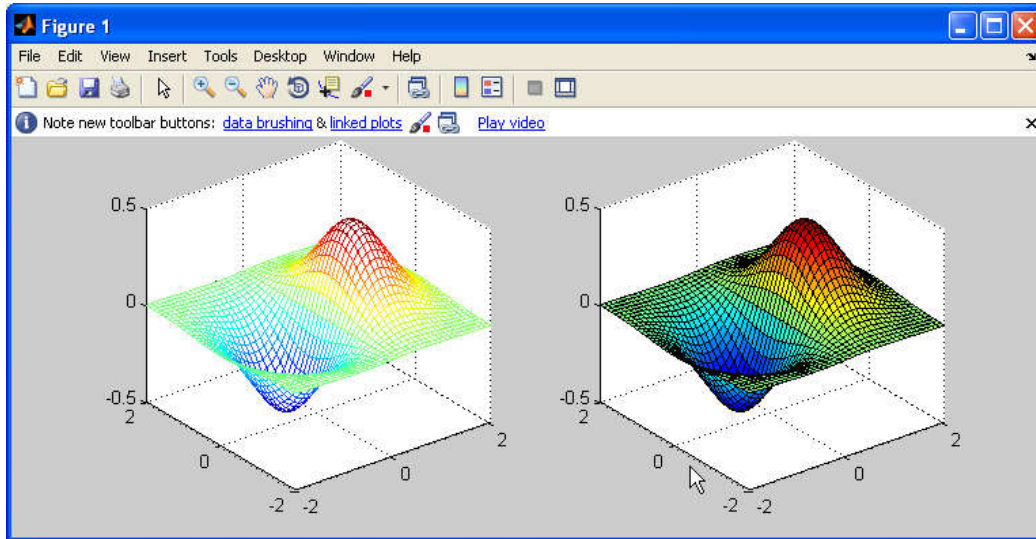


Slika 11. Rezultat dobijen izvršavanjem programa iz primera 9

Primer 10

Crtanje mrežastog i površinskog 3D grafika funkcije z , koja predstavlja funkciju dve varijable, x i y .

```
%mreža tačaka u x,y ravni u kojima se racuna i crta funkcija z
[x,y]=meshgrid([-2:0.1:2], [-2:0.1:2]);
z=x.*exp(-x.^2-y.^2);           %definisanje funkcije z
subplot(121), mesh(x,y,z)       %crtanje u 3D prostoru - mrezasto
subplot(122), surf(x,y,z)       %crtanje u 3D prostoru - površinsko
```



Slika 12. Rezultat dobijen izvršavanjem programa iz primera 10

Simboličko računanje - Symbolic Math Toolbox

Svi prethodni primeri generisanja funkcija i vršenja različitih operacija nad definisanim funkcijama podrazumevaju da su promenljive i njihove funkcije zadate u tabelarnoj formi sa dovoljnom tačnošću. Međutim, Matlab omogućava i simboličke operacije nad funkcijama korišćenjem dodatnog alata - *Symbolic Math Toolbox*.

Ideja simboličkog računanja u Matlabu biće veoma ukratko prezentovana kroz nekoliko jednostavnih primera:

```
>> clear %briše radni prostor
>> syms x %definiše simboličku promenljivu x
>> syms a
>> syms b
>> int(x^2,a,b) %integral f(x)=x^2 u granicama od a do b
ans =
b^3/3 - a^3/3

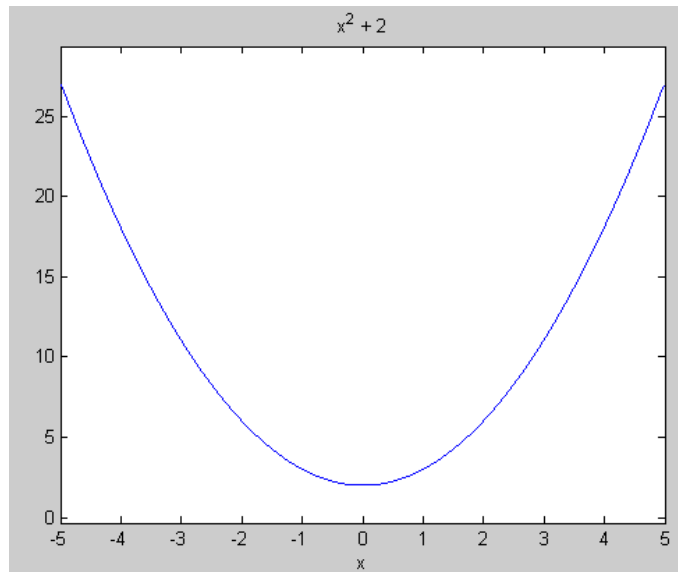
>> int(1/(1+x^2),a,b)
ans =
atan(b) - atan(a)

>> diff(log(x)) %prvi izvod funkcije log(x) tj. ln(x)
ans =
1/x

>> y=x^2+2 %y je takodje simbolička promenljiva
y =
x^2 + 2

>> diff(y)
ans =
2*x

>> ezplot(y,[-5 5]) %grafik funkcije y u granicama od -5 do 5
```



Slika 13. Rezultat dobijen izvršavanjem prethodne komande

```
>> solve(x^2-4) %traženje nula funkcije f(x)=x^2-4
ans =
    -2
     2
```

```
>> solve('x^3=27') % rešavanje jednačine -> isto kao: solve(x^3-27)
ans =
          3
- (3*3^(1/2)*i)/2 - 3/2
 (3*3^(1/2)*i)/2 - 3/2
```

Prilozi

1. Često korišćeni operatori

Znak	Opis
+	Sabiranje
-	Oduzimanje
*	Množenje skalarnih vrednosti, vektora i matrica
.*	Množenje vektora i matrica element po element
/	Deljenje (za matrice - sa desne strane)
./	Deljenje vektora i matrica element po element
^	Stepenovanje
.^	Stepenovanje vektora i matrica element po element
:	Generisanje vektora sa konstantnim korakom između elemenata
[]	Generisanje vektora i matrica
,	Razdvaja kolone vektora ili matrica; razdvaja komande; razdvaja argumente pri pozivu funkcije
%	Koristi se na početku komentara u M-fajlu
()	Prioritet operacija, argumenti funkcija, indeksi vektora i matrica
=	Operator dodele vrednosti
<	Manje
>	Veće
<=	Manje ili jednako
>=	Veće ili jednako
==	Jednako
~=	Različito
&	Logičko I (AND)
	Logičko ILI (OR)
~	Logičko NE (NOT)

2. Korisne komade

Komanda	Opis
clc	Briše sadržaj komandnog prozora
clear	Uklanja sve promenljive iz radnog prostora
clear x	Uklanja promenljivu x iz radnog prostora
help	Pomoć za Matlab funkcije
lookfor	Traži zadatu reč u Matlab sistemu za pomoć
whos	Prikazuje spisak promenljivih i njihovih vrednosti iz radnog prostora
format bank	Zaokruživanje rezultata na dve decimale
format long	Zaokruživanje na 14 decimala
format long e	Naučna notacija sa 15 decimala
format short	Zaokruživanje na 4 decimale
format short e	Naučna notacija sa 4 decimale

3. Unapred definisane promenljive i konstante

<i>Komanda</i>	<i>Opis</i>
ans	Vrednost poslednjeg izraza
eps	Najmanja razlika između dva broja u Matlabu
i	Imaginarna jedinica
j	Imaginarna jedinica
pi	Broj π
inf	Beskonačnost
NaN	Nije broj (<i>not a number</i>)