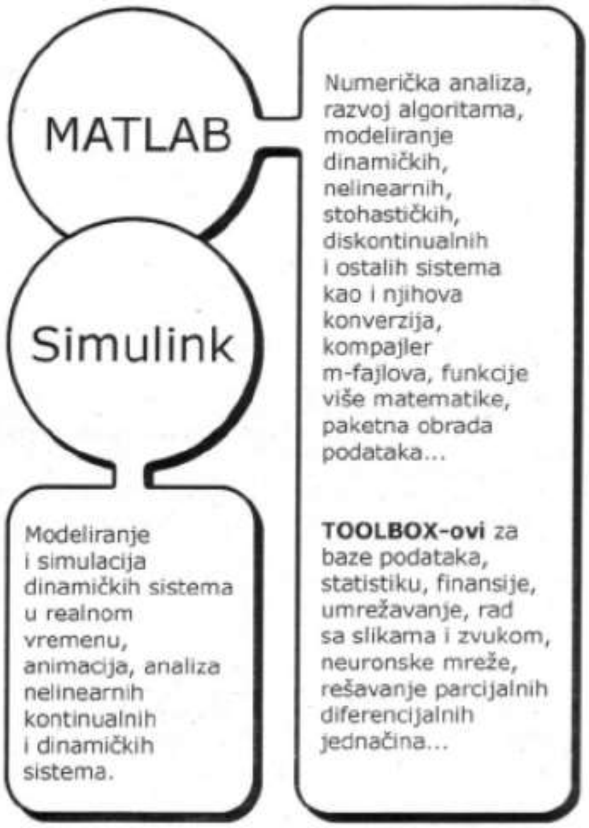
Contents

1. UVOD

MATLAB je moćan programski jezik za tehničke proračune. Nije klasičan programski jezik, već predstavlja nešto između softverskog paketa i programskog jezika. Ime je dobio od reči MATrična LABoratorija (engl. *matrix laboratory*), pošto mu je osnovni element podataka matrica (niz). MATLAB se može koristiti za matematička izračunavanja, modelovanje i simulacije, analizu i obradu podataka, grafičko prikazivanje rezultata i razvoj algoritama.



MATLAB je široko rasprostranjen na univerzitetima i višim školama, na uvodnim i naprednim kursevima iz matematike, prirodnih nauka, a naročito inženjerskih oblasti. U industriji se ovaj program upotrebljava za istraživanje, razvoj i projektovanje. Prva objavljena verzija je MATLAB 4.0., poslednja verzija MATLAB R2012a. Praktično, svake godine se pojavljuje nova verzija.

1.1 POKRETANJE MATLAB-A I NJEGOVI PROZORI

Komandni prozor (Command Window) je glavni prozor, koji se automatski otvara kada se MATLAB pokrene. Služi za unošenje i izvršavanje komandi i programa.

Grafički prozor (Figure) se automatski otvara kada se izvršavaju grafičke komande; sadrži grafikone koje su te komande nacrtale. Prozor se otvara iz menija **File** u komandnom prozoru.

Prozor za pisanje programa (Editor). U njemu se pišu i uređuju programi u obliku skript ili funkcijskih datoteka.

Prozor sistema za pomoć (**Help**): Sadrži ugrađenu pomoć, može se otvoriti iz menija **Help**. Interaktivan je i služi za dobijanje pomoćnih informacija o bilo kojoj komponenti ili svojstvu MATLAB-a.

Ostali prozori: Prozor sa prethodnim komandama (**Command History**), Prozor radnog prostora (**Workspace Window**), Prozor tekućeg direktorijuma (**Current Directory**), Prozor za pokretanje (**Launch Pad Window**).

Podešavanje vidljivosti pojedinih prozora na ekranu se vrši u meniju **Desktop**, izborom odgovarajuće opcije u polju **Desktop Layont.**

1.2 Rad u komandnom prozoru

Komandni prozor je glavni MATLAB-ov prozor i služi za izvršavanje komandi, otvaranje prozora, pokretanje programa koje je napisao korisnik i upravljanje MATLAB-om.

- Komanda se upisuje iza komandnog odzivnika >>
- Upisana komanda biće izvršena nakon pritiska na taster **Enter.** Izvršava se samo poslednje upisana komanda.
- U isti red može biti upisano više komandi, ako se razdvoje zarezom. Nakon pritiska **Enter**, upisane komande se izvršavaju redom kako su upisane, sleva na desno.
- Pomoći strelice ↑ i ↓ prikazuju se prethodno upisane komande
- Komanda se prenosi u sledeći red unošenjem ... na kraju reda
- Upisivanjem znaka **;** na kraju komande njen rezultat se neće prikazati na ekranu nakon izvršavanja.
- Znak % se koristi na početku komentara
- Komanda clc se koristi za brisanje sadržaja komandnog prozora

Osnovne matematičke operacije – aritmetičke operacije sa skalarima:
sabiranje (+), oduzimanje (-), množenje (*), deljenje zdesna (/), deljenje sleva (\), stepenovanje (^).

Prioritet izvršavanja. MATLAB izvršava operacije prema sledećem redosledu prioriteta, koji je isti kao na većini kalkulatora.

Prioritet	**Matematička operacija**
Najviši	Zagrade. Kad su zagrade ugnežđene, prvo se izračunva unutrašnja zagrada.
Drugi po redu	Stepenovanje
Treći po redu	Množenje, deljenje (jednak prioritet)
Četvrti po redu	Sabiranje i oduzimanje (jednak prioritet)

<u>Korišćenje MATLAB-a kao kalkulatora.</u> MATLAB se najjednostavnije koristi kao kalkulator, kada se u komandni prozor upisuju matematički izrazi i izvršavaju pritiskom tastera Enter.

<u>Primer 1.1:</u> Izračunati vrednosti izraza

a)	$5+3$	`>> 5+3`
b)	$\frac{5+3}{5\cdot 6}$	`>> (5+3)/(5*6)`
c)	$9\cdot\frac{6}{12}+7\cdot 5^{3/2}$	`>> 9*6/12+7*5^(3/2)`
d)	$\frac{35.7\cdot 64-7^3}{45+5^2}$	`>> (35.7*64-7^3)/(45+5^2)`

e)	$(2+7)^3 - \dfrac{273^{2/3}}{2} + \dfrac{55^2}{3}$	`>> (2+7)^3-273^(2/3)/2+55^2/3`

Formati prikaza rezultata. Korisnik može da izabere format u kojem MATLAB prikazuje rezultat na ekranu. Podrazumevani format je fiksni zarez i 4 decimale (format **short**). Promena formata prikaza rezultata se zadaje komandom **format** posle koje se upisuje naziv željenog formata prikaza. Format prikaza na ekranu ne utiče na preciznost kojom MATLAB izračunava i pamti brojeve.

Komanda	**Opis**	**Primer**

`format short`	Fiksni zarez sa četiri decimale za decimalne brojeve u opsegu: 0.001≤broj≤1000. Izvan ovog opsega primenjuje se format `short e`	`>> 290/7`
		`ans =`
		`41.4286`

`format long`	Fiksni zarez sa 14 decimala za decimalne brojeve u opsegu 0.001≤broj≤100. Izvan tog opsega primenjuje se format `long e`	`>> 290/7`
		`ans =`
		`41.428571428571431`

`format short e`	Naučna notacija sa 4 decimale	`>> 290/7`
		`ans =`
		`4.1429e+001`

`format long e`	Naučna notacija sa 15 decimala	`>> 290/7`
		`ans =`
		`4.142857142857143e+001`

`format short g`	Pet cifara s fiksnim ili pokretnim zarezom	`>> 290/7`
		`ans =`
		`41.429`

`format long g`	Petnaest cifara s fiksnim ili pokretnim zarezom	`>> 290/7`
		`ans =`
		`41.4285714285714`

`format bank`	Dve decimale	`>> 290/7`
		`ans =`
		`41.43`

<u>Primer 1.2</u> Izračunati vrednost izraza $\dfrac{470}{7}$ i prikazati rezultat u osnovnim formatima

```
>> format short
>> 470/7

>> format long
>> 470/7

>> format short e
>> 470/7

>> format long e
>> 470/7

>> format bank
>> 470/7
```

Ugrađene elementarne matematičke funkcije. Sem osnovnih aritmetičkih operacija, izrazi u MATLAB-u mogu sadržati i funkcije. MATLAB ima veoma veliku biblioteku ugrađenih funkcija, što ga čini jako popularnim za inženjersku primenu. Naravno, korisnik može da definiše svoje funkcije. Funkcija se poziva imenom i argumentom u zagradama. Na primer, funkcija `sqrt(x)` izračunava kvadratni koren broja. Ime funkcije je `sqrt`, a argument joj je `x`. Argument funkcije može biti broj, promenljiva kojoj je prethodno dodeljena numerička vrednost, ili izraz koji sadrži brojeve i/ili promenljive. I argumenti i izrazi mogu sadržati druge funkcije.

Funkcija	Opis		Funkcija	Opis
`sqrt(x)`	Kvadratni koren		`factorial(x)`	Faktorijel od x (x!)
`exp(x)`	Eksponencijalna funkcija (e^x)		`sin(x)`	Sinus ugla x (u radijanima)
`abs(x)`	Apsolutna vrednost		`cos(x)`	Kosinus ugla x (u radijanima)
`log(x)`	Prirodni logaritam, tj. logaritam sa osnovom e (ln)		`tan(x)`	Tangens ugla x (u radijanima)
`log10(x)`	Logaritam sa osnovom 10		`cot(x)`	Kotangens ugla x (u radijanima)

Primer 1.3: Izračunati

a) $$\cos^2\left(\frac{5\pi}{6}\right)\cdot\sin\left(\frac{7\pi}{8}\right)^2+\frac{\tan\left(\frac{\pi\cdot\ln 8}{6}\right)}{\sqrt{7}}$$

```
>> (cos(5*pi/6))^2*sin(7*pi/8)^2+tan(pi*log(8)/6)/sqrt(7)
```

b) $$\log\left|5!-2^5\right|-e^2$$

```
>> log10(abs(factorial(5)-2^5))-exp(2)
```

Definisanje promenljivih. Promenljiva je ime od jednog slova ili proizvoljne kombinacije slova i cifara (s početnim slovom) kojem je pridružena numerička vrednost. Promenljiva kojoj je pridružena numerička vrednost, može se upotrebljavati u matematičkim izrazima, funkcijama i svim MATLAB iskazima i komandama. Promenljiva je zapravo ime određene lokacije u memoriji. Kada se definiše nova promenljiva, MATLAB joj dodeljuje odgovarajuću lokaciju u memoriji gde čuva njoj pridruženu vrednost. Ako se promenljivoj promeni vrednsot, menja se sadržaj odgovarajuće lokacije u memoriji.

Pravila o imenima promeljivih: Mogu sadržati slova, cifre i podvlake; Moraju počinjati slovom; MATLAB pravi razliku između velikih i malih slova; Treba izbegavati korišćenje imena ugrađenih funkcija za promenljive (sin, cos,...).

Operator dodele. U MATLAB-u se znak = naziva operatorom dodele. Ovaj operator dodeljuje vrednost promenljivoj.

```
Ime_promenljive = numerička vrednost ili izraz
```

Levo od operatora dodele može biti samo jedna promenljiva. Desno može biti broj ili izraz koji sadrži brojeve i/ili promenljive kojima su prethodno dodeljene numeričke vrednosti.

Unapred definisane promenljive. Neke često korišćene promenljive automatski se definišu čim se MATLAB pokrene. To su:

`ans`	Ako promenljivoj iz poslednjeg izraza nije dodeljeno ime, MATLAB joj automatski dodeljuje ime snima u memoriju kao `ans`
`pi`	Broj π
`eps`	Najmanja razlika između dva broja koju MATLAB još može da registruje. Jednaka je 2^{-52}
`inf`	Beskonačno velika vrednost
`i`	Imaginarna jedinica
`j`	Isto kao i
`NaN`	Skraćeno od Not-a-Number (nije broj). Upotrebljava se kada MATLAB ne može da izračuna numeričku vrednsot. Npr. 0/0

Primer 1.4: Definisati promenljivu x i dodeliti joj vrednost 13.5 a zatim izračunati izraze

a) $x^3+5x^2-26.7x-52$ b) $\frac{\sqrt{14x^3}}{e^{3x}}$ c) $\log\left|x^2-x^3\right|$

```
>> x=13.5

>> x^3+5*x^2-26.7*x-52

>> sqrt(14*x^3)/exp(3*x)

>> log10(abs(x^2-x^3))
```

Primer 1.5: Definisati promenljive a,b,c i d kao $a=15.62, b=\text{-}7.08, c=62.5$ i $d=0.5(ab\text{-}c)$, a zatim izračunati vrednost izraza: $A=a+\frac{ab}{c}\frac{(a+d)^2}{\sqrt{|ab|}}$. Nakon toga dodeliti nove vrednosti promenljivim zaokruživanjem na najbliži ceo broj i izračunati novu vrednost izraza A.

```
>> a=15.62;b=-7.08;c=62.5;d=0.5*(a*b-c);

>>A=a+a*b/c*(a+d)^2/sqrt(abs(a*b))

>> a=round(a)

>> b=round(b)

>> c=round(c)

>> d=round(0.5*(a*b-c))

>> A=a+a*b/c*(a+d)^2/sqrt(abs(a*b))
```

Primer 1.6: Trigonometrijska formula je data jednačinom:

$$\cos^2\frac{x}{2}=\frac{\tan x+\sin x}{2\tan x}$$

Proveriti da li je formula ispravna tako što ćete izračunati vrednost obe strane jednačine, dodeljivanjem promenljivoj $x = \frac{\pi}{5}$.

```
>> x=pi/5;
>> (cos(x/2))^2

ans =

    0.9045

>> (tan(x)+sin(x))/(2*tan(x))

ans =

    0.9045
```

1.3 Zadaci za vežbanje

Zadatak 1.1: Izračunajte:

a) $\frac{3^7 \log(76)}{7^3 + 546} + \sqrt[3]{910\sin(\pi/6)}$, b) $43\frac{\left(\sqrt[4]{250}+23\right)^2}{e^{\left(45-\sqrt{3}\right)}}$

Zadatak 1.2: Definišite promenljive x i z kao $x = 9.6$ i z=8.1 i izračunajte:

a) $xz^2 - \left(\frac{2z}{3x}\right)^{\frac{x}{z}}$, b) $\frac{443z}{2x^3} + \frac{e^{-xz}}{(x+z)}$

Zadatak 1.3: Definisati kompleksne promenljive: $a = 2 + j3$ i $b = 5e^{j\frac{2\pi}{3}}$ a zatim izračunati vrednost izraza:

$$X = a - b + ab - a/b$$

Zadatak 1.4: Napravite MATLAB kod za sledeći izraz:

$$5^2 \cdot 3 - \frac{\sin(2+3^4)}{\sqrt[3]{22}}$$

Izaberite ime promenljive tri ponuđena: `5x`, `tan`, `izraz` i dodelite joj vrednsot prethodnog izraza. Prikažite vrednosti te promenljive u različitim formatima (short e, long i bank)

Zadatak 1.5: Izračunajte (jednom komandom) poluprečnik r lopte čija je zapremina V=350 cm^3. Pomoću tog r izračunajte površinu lopte P.

Zadatak 1.6: Rastojanje d tačke (x_0, y_0) od prave $Ax + By + C = 0$ dato je formulom:

$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

Izračunajte rastojanje tačke (2, -3) od prave 3*x* + 5*y* – 6 = 0. Prvo definišite promenljive *A*, *B*, *C*, x_0 i y_0, a potom izračunajte *d*.

Zadatak 1.7: Date su dve trigonometrijske formule:

a) $\tan 2x = \dfrac{2\tan x}{1-\tan^2 x}$, b) $\tan\dfrac{x}{2} = \sqrt{\dfrac{1-\cos x}{1+\cos x}}$

Proverite ispravnost formula tako što ćete izračunati obe strane svake jednačine za $x = \dfrac{3}{17}\pi$.

Zadatak 1.8: Dat je trougao sa stranicama: *a* = 18 cm, *b* = 35 cm i *c* = 50 cm. Primenom kosinusne teoreme odredite ugao γ naspram stranice *c* u trouglu.
(kosinusna teorema: $c^2 = a^2 + b^2 - 2ab\cos\gamma$)

Zadatak 1.9: Definišite sledeće promenljive:

cena_stola=9999.55 RSD
cena_stolice=2999.66 RSD

a) Promenite format prikaza na `bank`
b) Izračunajte cenu dva stola i osam stolica
c) Isto kao pod b) ali dodajte 20% PDV
d) Isto kao pod c) ali zaokružite ukupnu cenu do najbližeg celog broja.

Zadatak 1.10: Jačina zemljotresa *M* izračunava se po Rihterovoj skali kao $M = \dfrac{2}{3}\log\dfrac{E}{E_0}$, gde je *E* energija koju je zemljotres oslobodio, a $E_0 = 10^{4.4}$ (J) konstanta (energija malog referentnog zemljotresa). Izračunajte koliko puta više energije oslobodi zemljotres jačine 7.2 po Rihteru od zemljotresa jačine 5.3 po Rihteru.

Zadatak 1.11: Moment trofazne asinhrone mašine se izračunava pomoću sledećeg izraza:

$$M = \frac{3}{314}\frac{R_r}{s}\frac{U_f^2}{\left(R_s + \dfrac{R_r}{s}\right)^2 + \left(X_s + X_r\right)^2}$$

Izračunati vrednost momenta asinhrone mašine *M* za sledeće vrednosti parametara:
otpor statora $R_s = 0.05\Omega;$ reaktansa statora $X_s = 0.25\Omega;$ otpor rotora $R_r = 0.05\Omega;$ reaktansa rotora $X_r = 0.21\Omega;$ fazni napon $U_f = 220V$, i klizanje *s* = 0.04.

Zadatak 1.12: Efektivna vrednost struje u rednom RLC kolu se izračunava prema izrazu:

$$I = \frac{U}{\sqrt{R^2 + \left(\omega L - \dfrac{1}{\omega C}\right)^2}}$$

Izračunati struju *I* ako je efektivna vrednost naizmeničnog napona *U* =25 V, otpornost u kolu *R*=95 Ω, induktivnost *L*= 0.5 *H*, kapacitivnost $C = 20\cdot 10^{-6} F$ i kružna frekvencija $\omega = 3400\,\mathrm{s}^{-1}$

Zadatak 1.13: Trgovac je platio za robu 24000 dinara. Polovinu te robe prodao je uz zaradu od 15%, trećinu uz zaradu od 8%, a ostatak uz gubitak od 6%. Koliko dinara je zaradio?

Rešenje:

1.1.

Matlab koristi radijane umesto stepene. Npr. sin(π/6)=1/2, a ako se u matlabu ukuca sin(30) dobija se resenje ans=-0.988, sto nema veze sa ½. Zato se stepeni prvo moraju pretvoriti u radijane prema formuli:

$x[\mathrm{rad}] = \frac{\mathrm{ugao}[^{o}]}{180} \cdot 3.14$, odnosno:

$$\frac{\pi}{6}[\mathrm{rad}] = \frac{30}{180} \cdot 3.14 = 0.5236$$

Sa druge strane, može se koristiti direktno ugao zadat preko broja π, s tim da se umesto π ne koristi 180^{o}, vec 3.14 rad.

a) 3^7*log10(76)/(7^3+546)+(910*sin(pi/6))^(1/3) ans=12.3183;
b) 43*(250^(1/4)+23)^2/(exp(45-sqrt(3))) ans=5.0629e-15

1.2.

a)
```
>>x=9.6;
>> z=8.1;
>> x*z^2-(2*z/(3*x))^(x/z)

ans =

     629.3504
```

b)
```
>> 443*z/(2*x^3)+exp(-x*z)/(x+z)

ans =

    2.0279
```

1.3.

```
>> a=2+i*3;
>> b=5*exp(i*2*3.14/3);
>> X=a-b+a*b-a/b

X =

 -13.8140 + 0.4926i
```

1.4.

```
>> izraz=5^2*3-sin(2+3^4)/(22^(1/3))

izraz =

   74.6544
```

```
>> format short e
>> format long
>> format bank
```

Vratiti podrazumevani format: >> format short

1.5.

Zapremina lopte je:

$$V = \frac{4 \cdot r^3 \cdot \pi}{3} \Rightarrow r = \sqrt[3]{\frac{3 \cdot V}{4 \cdot \pi}}$$

Površina lopte je:

$$V = 4 \cdot r^2 \cdot \pi$$

```
>> r=(3*350/(4*pi))^(1/3)

r =

    4.3718

>> P=4*r^2*pi

P =

  240.1759
```

1.6.

```
>> A=3;
>> B=5;
>> C=-6;
>> x0=2;
>> y0=-3;
>> d=abs(A*x0+B*y0+C)/sqrt(A^2+B^2)

d =

    2.5725
```

1.7.

a)

```
>> x=3*pi/17;
>> tan(2*x)

ans =

    2.0083

>> 2*tan(x)/(1-(tan(x))^2)
```

b)

```
>> tan(x/2)

ans =

    0.2845

>> sqrt((1-cos(x))/(1+cos(x)))
```

ans =

2.0083

ans =

0.2845

1.8.

$$\gamma = a\cos\frac{a^2 + b^2 - c^2}{2ab}$$

```
>> a=18;
>> b=35;
>> c=50;
>> gama=acos((a^2+b^2-c^2)/(2*a*b))

gama =

   2.4261
```

2.4261 (u rad) ili $\frac{2.4261}{3.14}\cdot 180 = 139.17^{o}$

1.9.

a)

```
>> cena_stola=9999.55;
>> cena_stolice=2999.66;
>> format bank
>> cena=2*cena_stola+8*cena_stolice

cena =

      43996.38
```

b)

```
>> cena_sa_PDV=cena+0.2*cena

cena_sa_PDV =

      52795.66
```

c)

```
>> cena_zaokruzena=round(cena_sa_PDV)

cena_zaokruzena =

       52796.00
```

1.10.

$$M = \frac{2}{3}\log\frac{E}{E_0} \Rightarrow \log\frac{E}{E_0} = \frac{3M}{2}$$ (pošto nema osnovu dopisuje se 10)

$$\log_{10}\frac{E}{E_0} = \frac{3M}{2} \Rightarrow \frac{E}{E_0} = 10^{\frac{3M}{2}} \Rightarrow E = E_0 \cdot 10^{\frac{3M}{2}}$$

```
>> E0=10^4.4;
>> M1=7.2;
>> M2=5.3;
>> (E0*10^(3*M1/2))/(E0*10^(3*M2/2))

ans =

  707.9458
```

1.11.

```
>> Rs=0.05;
>> Xs=0.25;
>> Rr=0.05;
>> Xr=0.21;
>> Uf=220;
>> s=0.04;
>> M=3*Rr*Uf^2/(314*s*(Rs+Rr/s)^2+(Xs+Xr)^2)

M =

  338.6510
```

1.12

```
>> U=25;
>> R=95;
>> L=0.5;
>> C=20*10^-6;
>> omega=3400;
>> I=U/sqrt(R^2+(omega*L-1/(omega*C))^2)

I =

   0.0148
```

1.13

```
>> zarada=(1.15*24000/2+1.08*24000/3+24000/6-0.06*24000/6)-24000

zarada =

       2200
```

2. NIZOVI (VEKTORI I MATRICE)

Niz je osnovni oblik u kojem MATLAB čuva podatke i radi sa njima. Niz je skup brojeva poređanih u vrste (redove) i/ili kolone. Najjednostavniji niz je jednodimenzionalni niz – vrsta ili kolona brojeva. Dvodimenzionalni niz je skup brojeva poređanih u vrste i kolone. U prirodnim i tehničkim naukama, jednodimenzionalni nizovi predstavljaju vektore, a dvodimenzionalni matrice.

2.1 Generisanje vektora

Vektor je jednodimenzionalni niz skupa brojeva poređanih u vrstu ili kolonu.

Vektor se generiše upisivanjem elemenata unutar uglastih zagrada [].

`Ime_promenljive = [elementi vektora]`

Vektor vrsta se dobija kada se elementi vektora razdvoje razmakom ili zarezom.

Primer 2.1: Napraviti vektor vrstu V od elemenata: 32, 4, 81, $e^{2.5}$, 63, $\cos(\pi/3)$ i 14.2.

```
>> V=[32,4,81,exp(2.5),63,cos(pi/3),14.2]
```

Vektor kolona se dobija kda se elementi vektora razdvoje znakom `;` ili ako se prilikom upisivanja posle svakog elementa pritisne taster Enter.

Primer 2.2: Napraviti vektor kolonu K od elemenata: 55, 14, ln(51), 987, 0, $\sqrt{11}$ i $5\sin(1.3\pi)$.

```
>> K=[55;14;log(51);987;0;sqrt(11);5*sin(1.3*pi)]
```

Pored ručnog unošenja svakog pojedinačnog elementa, vektori se mogu generisati pomoću ugrađenih funkcija za generisanje vektora sa određenim osobinama.

Generisanje vektora sa konstantnim korakom između elemenata zadavanjem prvog elementa, koraka i poslednjeg elementa:

`Ime_promenljive = [m:q:n]` ili `Ime_promeljive = m:q:n`

Gde je `m` prvi element vektora, `q` korak između susednih elemenata, i `n` poslednji element vektora. Ako se izostavi korak, podrazumeva se korak 1.

Primer 2.3: Generisati vektor X u kojem je prvi element 1, poslednji 15, a korak između elemenata 2.

```
>> X=[1:2:15]
```

Primer 2.4: Generisati vektor Y u kojem je prvi element 1, poslednji 15, a korak između elemenata 1.

```
>> X=[1:15]
```

Primer 2.5: Generisati vektor Z čiji je prvi element 20 a poslednji -5, a korak između elemenata -1.

```
>> VK=20:-1:-5
```

Za generisanje vektora sa konstantnim korakom između elemenata, zadavanjem prvog i poslednjeg elementa i ukupnog broja elemenata vektora, koristi se funkcija `linspace`

```
Ime_promenljive = linspace(xi,xf,n)
```

Gde je `xi` prvi element vektora, `xf` poslednji element vektora, a `n` broj elemenata vektora.

Primer 2.6: Generisati vektor V sa 10 jednako razmaknutih elemenata, u kome je prvi element -3 a poslednji 25.

```
>> V=linspace(-3,25,10)
```

2.2 Generisanje matrica

Matrica je dvodimenzionalni niz čiji su elementi poređani u vrste i kolone. Elementi matrice mogu biti brojevi ili matematički izrazi koji sadrže brojeve, unapred definisane promenljive i funkcije.

Generisanje matrica: Elementi matrice se upisuju vrstu po vrstu unutar uglastih zagrada. Sve vrste moraju imati jednak broj elemenata. Matrica se definiše dodeljivanjem elemenata matrice promenljivoj:

```
Ime_promenljive = [elementi prve vrste;elementi druge vrste;
     elementi treće vrste;…;elementi poslednje vrste]
```

Primer 2.7: Generisati matricu A

$$A = \begin{bmatrix} 3 & 10 & 9 & 6 & 4 \\ 11 & 7 & 3 & 21 & 6 \\ 8 & 8 & 1 & 3 & 3 \end{bmatrix}$$

```
>> A=[3,10,9,6,4;11,7,3,21,6;8,8,1,3,3]
```

MATLAB sadrži funkcije za **generisanje specijalnih matrica**. Neke od njih su date u sledećoj tabeli:

Funkcija	**Opis**
`eye`	Generiše jediničnu matricu (matrica u kojoj su elementi na glavnoj dijagonali jedinice a vandijagonalni nule).
`ones`	Generiše matricu u kojoj su svi elementi jedinice.
`zeros`	Generiše nula matricu (matricu sa u kojoj su svi elementi nule).
`rand`	Generiše uniformno raspodeljenih slučajnih brojeva u intervalu 0 do 1 (kao skalara ili matrice)
`magic`	Generiše magičnu matricu tzv. “čarobni kvadrat” (za n>=3, dobija se matrica čiji su elementi prirodni brojevi od 1 do n^2 i kod koje svaka vrsta i kolona imaju isti zbir elemenata.

Primer 2.8: Upotrebom ugrađenih funkcija generisati nula matricu dimenzija 5x3, matricu dimenzija 3x4 čiji su svi elementi jedinice, jedniničnu matricu četvrtog reda, matricu slučajnih brojeva dimenzija 4x2 i magičnu matricu trećeg reda.

```
>> nula_matrica=zeros(5,3)
nula_matrica =
     0     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0

>> matrica_jedinica=ones(3,4)
matrica_jedinica =
     1     1     1     1
     1     1     1     1
     1     1     1     1

>> jed_matrica=eye(4)
jed_matrica =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1

>> matrica_sluc_br=rand(4,2)
matrica_sluc_br =
    0.1270    0.2785
    0.9134    0.5469
    0.6324    0.9575
    0.0975    0.9649

>> magicna_matrica=magic(4)
magicna_matrica =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

Vrste matrica se mogu generisati kao vektori, koristeći funkcije za generisanje vektora (vektori sa jednako razmaknutim elementima ili vektori generisani pomoću komande `linspace`).

Primer 2.9: Generisati maticu M dimenzija 3x10, čiju prvu vrstu čine elementi od 1 do 10, drugu vrstu čini vektor čiji je prvi element 5 a poslednji 25 i treću vrstu čine elementi:
$-1, \sqrt{3}, 5, \ln(20), 18, \sin(\pi/8), -2, e^{5/4}$, 0 i 100.

```
>> M=[1:10;linspace(5,25,10);-1,sqrt(3),5,log(20),18,sin(pi/8),-2,exp(5/4),0,100]
```

Operator transponovanja pretvara vektor vrstu u vektor kolonu i obratno. Kada se primeni na matricu pretvara njene vrste u kolone i obratno. Operator transponovanja se primenjuje upisivanjem polunavodnika ' iza vektora odnosno matrice koje treba trasponovati.

2.3 Adresiranje vektora i matrica

Adresiranje vektora se odnosi na definisanje položaja njegovih elemenata. Ako postoji vektor `ve`, tada `ve(k)` označava element tog vektora na mestu `k`. Svaki element vektora se može upotrebljavati kao posebna promenljiva, kojoj se može dodeljivati nova vrednost i na taj način menjati sam vektor. Svaki element vektora se može upotrebljavati i kao posebna promenljiva u drugim matematičkim izrazima.

Primer 2.10: Napravite vektor vrstu `VR` od elemenata: 12, 15, 17, -1, 0, 3, 5, 7 i 9. Zatim prikažite peti element vektora `VR` i dodelite mu novu vrednost 1. Prikažite izmenjeni vektor `VR`. Izračunajte zbir kvadrata drugog elementa i dvostruke vrednosti devetog elementa vektora `VR`.

```
>> VR=[12,15,17,-1,0,3,5,7,9]
VR =
    12    15    17    -1     0     3     5     7     9

>> VR(5)
ans =
     0

>> VR(5)=1;
>> VR
VR =
    12    15    17    -1     1     3     5     7     9

>> VR(2)^2+2*VR(9)
ans =
   243
```

Adresiranje matrica. Adresa elementa matrice je definisana brojem vrste i kolone u kojoj se nalazi. Ako je definisana matrica `A`, tada `A(m,n)` označava se element matrice `A` u preseku vrste `m` i kolone `n`. Kao i kod vektora, može se menjati vrednost pojedinačnih elemenata matrice dodeljivanjem nove vrednosti tom elementu. Pojedinačni elementi matrice se mogu upotrebljavati kao nezavisne promenljive u matlab izrazima i funkcijama.

Primer 2.11: Napravite matricu $M = \begin{bmatrix} 2 & 5 & 4 \\ 3 & 6 & 1 \\ 8 & 9 & 7 \end{bmatrix}$

Zatim prikažite element matrice `M` u preseku prve vrste i treće kolone i dodelite mu novu vrednost 100. Prikažite izmenjenu matricu `M`. Izračunajte razliku sinusa elementa u preseku prve vrste i prve kolone i elementa u preseku treće vrste i druge kolone.

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

```
>> M(1,3)
ans =
     4

>> M(1,3)=100;
>> M
M =
     2     5   100
     3     6     1
     8     9     7

>> sin(M(1,1))-sin(M(3,2))
ans =
    0.4972
```

2.3.1 Upotreba dvotačke u adresiranju vektora i matrica

Za vektor, recimo `V`**,** `V(:)` označava sve elemente vektora `V`, a `V(m:n)` označava elemente vektora `V` od indeksa `m` do ineksa `n`.

Primer 2.12: Dat je vektor vrsta VR sa elementima: 1, 2, 3, 4, 5, 6, 7, 8, 9,10. Napravite vektor kolonu VK od trećeg do osmog elementa vektora VR.

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

>> VK=VR(3:8)'
VK =
     3
     4
     5
     6
     7
     8
```

Za matricu, recimo A, upotreba dvotačke u adresiranju ima sledeće varijante:
`A(:,n)` – označava (izdvaja) kolonu `n` matrice `A`.
`A(n,:)` – označava (izdvaja) vrstu `n` matrice `A`.
`A(:,m:n)` – označava (izdvaja) elemente u svim vrstama između kolona `m` i `n` matrice `A`.
`A(m:n,:)` – označava (izdvaja) elemente u svim kolonama između vrsta `m` i `n` matrice `A`.
`A(m:n,p:q)` – označava (izdvaja) elemente u vrstama od `m` do `n` i kolonama od `p` do `q` matrice `A`.
Iz vektora ili matrice se može ukolniti element, vrsta ili kolona njihovim adresiranjem i dodeljivanjem vrednosti [].

Primer 2.13: Dat je vektor vrsta VR sa elementima: 1, 2, 3, 4, 5, 6, 7, 8, 9,10. Koristeći VR napravite vektor kolonu VK koja izuzimajući peti element u vektoru VR.

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

>> VR(5)=[]
VR =
     1     2     3     4     6     7     8     9    10
```

```
>> VK=VR'
VK =
     1
     2
     3
     4
     6
     7
     8
     9
    10
```

Dodavanje novih elemenata postojećem vektoru se vrši tako što se dodeli vrednost novom elementu. Na primer, ako vector ima *n* elemenata, a nova vrednost se dodeli element čija je adresa (indeks) *n*+2 ili veća MATLAB dodeljuje nule elementima između poslednjeg originalnog elementa i novog elementa.

Primer 2.14:

```
>> X=[2,5,9,4]
X =
     2     5     9     4

>> X(5)=100
X =
     2     5     9     4   100

>> X(10)=50
X =
     2     5     9     4   100     0     0     0     0    50
```

Dodavanje novih elemenata matrici se vrši tako što se dodele vrednosti novim vrstama/kolonama odnosno kolonama. Ovde treba voditi računa o dimenzijama matrice. Takođe, mogu se postojećim elementima matrice dodeliti nove vrednosti.

Primer 2.15:

```
>> A=magic(3)
A =
     8     1     6
     3     5     7
     4     9     2

>> A(:,4)=[1,10,100]
A =
     8     1     6     1
     3     5     7    10
     4     9     2   100

>> A(4,:)=5
A =
     8     1     6     1
     3     5     7    10
     4     9     2   100
     5     5     5     5
```

```
>> A(1,:)=2:2:8
A =
     2     4     6     8
     3     5     7    10
     4     9     2   100
     5     5     5     5
```

2.4 Matematičke operacije sa vektorima i matricama

Sabiranje i odzimanje može biti sprovedeno nad vektorima i matricama jednakih dimenzija.

Primer 2.16

```
>> V1=[5,6,9];
>> V2=[-5,6,100];
>> V3=V1+V2
V3 =
     0    12   109
>> V3=V1+V2'
Error using  +           greška jer su vektori nejednakih dimenzija
Matrix dimensions must agree.

>> M1=[1,1,1;2,2,2]
M1 =
     1     1     1
     2     2     2

>> M2=[-1,5,9;5,5,5]
M2 =
    -1     5     9
     5     5     5

>> M3=M1+M2
M3 =
     0     6    10
     7     7     7

>> M2=M2'
M2 =
    -1     5
     5     5
```

```
     9     5


>> M3=M1+M2
Error using  +           %greška jer su matrice nejednakih dimenz.
Matrix dimensions must agree.
```

Množenje vektora i matrica u Matlabu izvodi se u skladu sa pravilima linearne algebre. To znači da ako su A i B dve matrice, operacija A*B može biti izvedena samo ako je broj kolona matrice A jednak broju vrsta matrice B.

Primer 2.17.

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

>> B=[5,1;6,9;10,20]
B =
     5     1
     6     9
    10    20

>> C=A*B
C =
   100   131
   100   167
   173   259
    63    90

>> D=B*A
Error using  *          %greška jer se ne slaže br. kol. B i vrsta A
Inner matrix dimensions must agree.
```

Operacije nad pojedinačnim elementima vektora i matrica se obavljaju se nad svakim elementom vektora ili matrice. Operacije na pojedinačnim elementima mogu se izvoditi samo nad nozovima istih dimenzija. Operacije sabiranja i oduzimanja su po prirodi operacije nad pojedinačnim elementima. Kod množenja i deljenja se ispred znaka matematičke operacije stavlja se tačka i to na sledeći način: Ako su dati vektori $a=[a_1\ a_2\ a_3\ a_4]$ i $b=[b_1\ b_2\ b_3\ b_4]$, množenje, deljenje i stepenovanje nad pojedinačnim elementima ova dva vektora vrši se na sledeći način:

$$a.*b=[a_1b_1\ a_2b_2\ a_3b_3\ a_4b_4],$$
$$a./b=[a_1/b_1\ a_2/b_2\ a_3/b_3\ a_4/b_4],$$
$$a.\wedge b=\left[(a_1)^{b_1}\ (a_2)^{b_2}\ (a_3)^{b_3}\ (a_4)^{b_4}\right]$$

Operacije nad pojedinačnim elementima su veoma pogodne za izračunavanje vrednosti neke funkcije ili izraza za mnogo različitih vrednosti argumenta ili argumenata.

Primer 2.18

```
>> X=[2,4,8;1,3,5]
X =
```

```
      2      4      8
      1      3      5

>> Y=[4,5,6;7,8,9]
Y =
      4      5      6
      7      8      9

>> Z=X.*Y
Z =
      8     20     48
      7     24     45

>> W=Y./X
W =
    2.0000    1.2500    0.7500
    7.0000    2.6667    1.8000

>> Q=W.^X
Q =
    4.0000    2.4414    0.1001
    7.0000   18.9630   18.8957
```

2.5 *Ugrađene funkcije u Matlabu za analizu vektora i matrica*

U MATLAB-u postoji veliki broj funkcija za analizu vektora i matrica. U sledećoj tabeli su date najčešće korišćene funkcije.

`length(A)`	Određuje broj elemenata vektora A	`>> A=[1,5,9,3,7,9,5,6,4,15,0,95];`
		`>> length(A)`
		`ans =`
		`    12`
		
		`>> x=length(A)`
		`x =`
		`  12`

`size(A)`	Daje broj vrsta i kolona matrice A	`>> A=[5,6,8;5,5,5;6,6,6;9,9,9]`
		`A =`
		`     5     6     8`
		`     5     5     5`
		`     6     6     6`
		`     9     9     9`
		
		`>> size(A)`
		`ans =`
		`     4     3`
		
		`>> dimA=size(A)`
		`dimA =`
		`     4     3`

`reshape(A,m,n)`	Preuređuje matricu A sa r vrsta i s kolona tako da ima m vrsta i n kolona, tako da je r*s=m*n	`>> A=[5,6,8;5,5,5;6,6,6;9,9,9]`
		`A =`
		`     5     6     8`
		`     5     5     5`

<table>
<tr><td></td><td></td><td><pre>
     6   6   6
     9   9   9

>> B=reshape(A,2,6)
B =
     5   6   6   6   8   6
     5   9   5   9   5   9
</pre></td></tr>
<tr><td>diag(V)</td><td>Kada je V vektor, generiše kvadratnu matricu sa elementima V na dijagonali.
Kada je V matrica, generiše vektor od elemenata na glavnoj dijagonali matrice V.</td><td><pre>
>> V=[1,2,3,4]

V =
     1   2   3   4

>> X=diag(V)
X =
     1   0   0   0
     0   2   0   0
     0   0   3   0
     0   0   0   4

>> V=magic(3)
V =
     8   1   6
     3   5   7
     4   9   2

>> X=diag(V)
X =
     8
     5
     2
</pre></td></tr>
<tr><td>mean(A)</td><td>Ako je A vektor, dobija se srednja vrednost elemenata tog vektora.
Ako je A matrica, dobija se vektor čiji su elementi srednje vrednosti kolona matrice A</td><td><pre>
>> A=[1,5,9,3,7,9,5,6,4,15,0,95];
>> mean(A)
ans =
   13.2500

>> A=[1,5,9;3,18,9;5,6,4]
A =
     1   5   9
     3   18   9
     5   6   4
>> mean(A)
ans =
    3.0000   9.6667   7.3333
</pre></td></tr>
<tr><td>C=max(A)

[d,n]=max(A)</td><td>Ako je A vektor, C je najveći element vektora A. Ako je A matrica, C je vektor vrsta koji sadrži najveće elemente svake kolone matrice A

Ako je A vektor, d je najveći element vektora A a n položaj tog elementa u vektoru A</td><td><pre>
>> A=[1,5,9,3,18,9,5,6,4,15];
>> C=max(A)
C =
    18
>> [d,n]=max(A)
d =
    18
n =
     5

>> A=[1,5,9;3,18,9;5,6,4]
A =
     1   5   9
     3   18   9
     5   6   4
>> C=max(A)
C =
     5   18   9
</pre></td></tr>
</table>

`min(A)`
`[d,n]=min(A)`
	Isto kao `max(A)`, ali za najmanji element.
	Isto kao `[d,n]=max(A)`, ali za najmanji element
		`>> A=[1,5,9,3,18,9,5,6,4,15];`
		`>> min(A)`
		`ans =`
		`     1`
		
		`>> [d,n]=min(A)`
		`d =`
		`     1`
		`n =`
		`     1`

`sum(A)`	Ako je `A` vektor, dobija se zbir elemenata vektora. Ako je `A` matrica, dobija se vektor čiji su elementi jednaki zbiru kolona matrice `A`.	`>> A=[1,5,9,3,18,9,5,6,4,15];`
		`>> sum(A)`
		`ans =`
		`    75`
		
		`>> A=[1,5,9;3,18,9;5,6,4];`
		`>> sum(A)`
		`ans =`
		`     9    29    22`

`sort(A)`	Ako je `A` vektor, ređa elemente vektora po rastućem redosledu. Ako je `A` matrica, sortira elemente kolona matrice `A`.	`>> A=[1,5,9,3,18,9,5,6,4,15];`
		`>> sort(A)`
		`ans =`
		`   1   3   4   5   5   6   9   9   15   18`
		
		`>> A=[1,5,9;3,18,9;5,6,4]`
		`A =`
		`     1     5     9`
		`     3    18     9`
		`     5     6     4`
		`>> sort(A)`
		`ans =`
		`     1     5     4`
		`     3     6     9`
		`     5    18     9`

`det(A)`	Izračunava determinantu kvadratne matrice `A`.	`>> A=[1,5,9;3,18,9;5,6,4]`
		`A =`
		`     1     5     9`
		`     3    18     9`
		`     5     6     4`
		
		`>> det(A)`
		`ans =`
		` -465.0000`

`inv(A)`	Izračunava inverznu matricu kvadratne matrice `A`.	`>> A=[1,5,9;3,18,9;5,6,4]`
		`A =`
		`     1     5     9`
		`     3    18     9`
		`     5     6     4`
		
		`>> inv(A)`
		`ans =`
		`   -0.0387   -0.0731    0.2516`
		`   -0.0710    0.0882   -0.0387`
		`    0.1548   -0.0409   -0.0065`

`dot(a,b)`	Izračunava skalarni proizvod vektora `a` i `b`.	`>> a=[2,3,5];`
		`>> b=[6,1,8];`
		`>> c=dot(a,b)`
		`c =`
		`    55`

`cross(a,b)`	Izračunava vektorski proizvod vektora `a` i `b`. Vektori moraju	`>> a=[2,3,5];` `>> b=[6,1,8];`

	imati po tri elementa.	`>> c=cross(a,b)`
		 `c =`
		 `19 14 -16`

2.6 Znakovni nizovi

Znakovni niz se dobija upisivanjem znakova unutar polunavodnika. Mogu da sadrže slova, brojeve, ostale simbole i razmake. Koriste se u kreiranju rezultata programa, formatiranju grafikona, itd.

Primer 2.19

```
>> Nastavnik='Jordan Radosavljevic'

Nastavnik =

Jordan Radosavljevic
```

MATLAB ima ugrađenu funkciju `char` koja od znakovnih nizova nejednakog broja elemenata generiše matricu sa vrstama jednake dužine. Ulazni argument funkcije char jesu znakovni nizovi razdvojeni zarezima.

Primer 2.20

```
>> Nastavnici=char('Jordan','Dardan','Miroljub','Nebojsa','Sasa','Zarko','Aleksandar','Slobodan','Bojan')

Nastavnici =

Jordan
Dardan
Miroljub
Nebojsa
Sasa
Zarko
Aleksandar
Slobodan
Bojan
```

Primer 2.21 Primer kreiranja izlaznih rezultata jednog programa iz elektroenergetike:

```
disp(' ')
   disp('REZULTAT:')
   disp(' ')
   disp('            pgub         Up   Sp=Pp+jQp ')
   disp(' o----.-----Zv-----.-----o---->   ')
   disp('      |             |      |        ')
   disp('    Yvo/2         Yvo/2    = Qk     ')
   disp('      |             |      |        ')
   disp('     ---           ---    ---       ')
   disp(' ')
   fprintf('Potrebna snaga kompenzacije je: Qk=%f MVAr.\n',Qk)
```

2.7 Zadaci za vežbanje

Zadatak 2.1 Napravite vektor kolonu kod koga je prvi element 1, poslednji -29 a elementi se smanjuju za po -3.

Zadatak 2.2 Napravite vektor vrstu kod koga je je prvi element 0, poslednji 100 a elementi se povećavaju za po 5.

Zadatak 2.3 Napravite vektor vrstu sa 12 jednako razmaknutih elemenata od kojih je prvi 10 a poslednji 100.

Zadatak 2.4 Napravite prikazanu matricu B koristeći ugrađene funkcije za generisanje vektora, i to za kreiranje prve i druge vrste dvotaču i za kreiranje treće vrste funkciju `linspace`.

$$B = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 & 25 \\ 72 & 66 & 70 & 54 & 48 & 42 & 36 & 30 & 24 \\ 0 & 0.125 & 0.250 & 0.375 & 0.500 & 0.625 & 0.750 & 0.875 & 1.000 \end{bmatrix}$$

Zadatak 2.5. Definisati matricu A, $A = \begin{bmatrix} 3 & 10 & 9 & 6 & 4 \\ 11 & 7 & 3 & 21 & 6 \\ 8 & 8 & 1 & 3 & 3 \end{bmatrix}$, a potom:

a) Napraviti vektor B od elemenata prve vrste matrice A.
b) Napraviti vektor C od elemenata druge kolone matrice A
c) Napravite matricu D od prve, druge i teće kolone matrice A
d) Napravite vektor E od elemenata A(1,3), A(2,4), A(3,5).
e) Odredite dimenziju matrice A i vektora B.
f) Smanjiti dimenziju vektora B ukanjanjem drugog elementa

Zadatak 2.6 Generisati:
a) Matricu A dimenzija 3×4 čiji su svi elementi nule.
b) Matricu B dimenzija 5×3 čiji su svi elementi jedninice.
c) Jediničnu matricu C dimenzija 4×4.

Zadatak 2.7 Dat je vektor A čiji su elementi: 4/3, 5, sin(π/3), 0, $\sqrt{22}$, ln(10). Potrebno je:
a) Odrediti maksimalnu vrednost i poziciju (indeks) tog elementa u vektoru
b) Odrediti minimalnu vrednost poziciju (indeks) tog elementa u vektoru
c) Odrediti srednju vrednost vektora A
d) Odrediti indekse elemenata vektora A koji su različiti od nule
e) Sortiraj elemente vektora A po rastućem nizu.

Zadatak 2.8 Data je matrica: $A = \begin{bmatrix} 1 & 5 & 3 \\ 2 & 4 & 6 \\ 3 & 8 & 4 \end{bmatrix}$

a) Odrediti determinantu matrice A
b) Odrediti transponovanu matricu matrice A
c) Odrediti inverznu matricu matrice A
d) Odrediti adjugovanu matricu matrice A
e) Odrediti maximalne vrednosti matrice A
f) Odrediti srednje vrednosti matrice A
g) Odrediti $A - A^T, A - 3, 3 \cdot A$ i A^3
h) Definiši kompleksnu matricu K čiji je realni deo prva vrsta matrice A a imaginarni deo treća vrsta transponovane matrice A.

Zadatak 2.9 Rešite sistem jednačina od četiri linearne jednačine koristeći ugrađenu funkciju `inv`

$5x + 4y - 2z + 6w = 4$
$3x + 6y + 6z + 4.5w = 13.5$
$6x + 12y - 2z + 16w = 20$
$4x - 2y + 2z - 4w = 6$

Zadatak 2.10 Data su tri cilindra čiji su poluprečnici osnova *r*: 2, 4, 6 a visine *h*: 3, 6, 9. Primenom operacija nad pojedinačnim elementima vektora odrediti zaperminine sva tri cilindra ($V = \pi r^2 h$)

Zadatak 2.11 Pomoću MATLABA pokažite da zbir beskonačnog reda $\sum_{n=0}^{\infty} \frac{1}{(2n+1)(2n+3)}$ konvergira broju 0.5. Uradite to izračunavanjem sume za:

a) *n*=50,
b) *n*=500,
c) *n*=5000.

Rešenje:

2.1.

```
>> [1:-3:-29]'

ans =

     1
    -2
    -5
    -8
   -11
   -14
   -17
   -20
   -23
   -26
   -29
```

2.2.

```
>> [0:5:100]

ans =

  Columns 1 through 15

     0     5    10    15    20    25    30    35    40    45    50    55    60    65    70

  Columns 16 through 21

    75    80    85    90    95   100
```

2.3

```
>> linspace(10,100,12)

ans =

  Columns 1 through 9

   10.0000   18.1818   26.3636   34.5455   42.7273   50.9091   59.0909   67.2727   75.4545
```

```
  Columns 10 through 12

   83.6364   91.8182  100.0000
```

2.4.

```
>> [1:3:25;72,66,70,54:-6:24;linspace(0,1,9)]

ans =

    1.0000    4.0000    7.0000   10.0000   13.0000   16.0000   19.0000   22.0000   25.0000
   72.0000   66.0000   70.0000   54.0000   48.0000   42.0000   36.0000   30.0000   24.0000
         0    0.1250    0.2500    0.3750    0.5000    0.6250    0.7500    0.8750    1.0000
```

2.5

```
>> A=[3,10,9,6,4;11,7,3,21,6;8,8,1,3,3]

A =

     3    10     9     6     4
    11     7     3    21     6
     8     8     1     3     3
```

a)

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

B =

     3    10     9     6     4
```

b)

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

C =

    10
     7
     8
```

c)

```
>> D=A(:,1:3)

D =
```

```
     3    10     9
    11     7     3
     8     8     1
```

d)

```
>> E=[A(1,3),A(2,4),A(3,5)]

E =

     9    21     3
```

e)

```
>> size(A)

ans =

     3     5

>> size(B)

ans =

     1     5
```

f)

```
>> B(2)=[]

B =

     3     9     6     4
```

2.6.

```
>> A=zeros(3,4)

A =

     0     0     0     0
     0     0     0     0
     0     0     0     0

>> B=ones(5,3)

B =

     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1

>> C=eye(4)
```

```
C =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

2.7.

```
>> A=[4/3,5,sin(pi/3),0,sqrt(22),log(10)]

A =

    1.3333    5.0000    0.8660         0    4.6904    2.3026
```

a)

```
>> [max_vred,poz]=max(A)

max_vred =

     5


poz =

     2
```

b)

```
>> [min_vred,poz]=min(A)

min_vred =

     0


poz =

     4
```

c)

```
>> sred_vred=mean(A)

sred_vred =

    2.3654
```

d)

```
>> poz_elem_raz_od_nule=find(A)
```

poz_elem_raz_od_nule =

```
    1     2     3     5     6
```

e)

```
>> niz=sort(A)

niz =

         0    0.8660    1.3333    2.3026    4.6904    5.0000
```

2.8.

Inverzna matrica matrice $A = [a_{ij}]_{n\times n}$ je matrica A^{-1} takva da važi

$$A \cdot A^{-1} = A^{-1} \cdot A = I.$$

Ako je $\det A \neq 0$, tada je

$$A^{-1} = \frac{1}{\det A}\,\mathrm{adj}\,A.$$

Odavde je: adjA=A^{-1}*det A

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

A =

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

a)

```
>> determinanta=det(A)

determinanta =

    30
```

b)

```
>> A_transponovano=A'

A_transponovano =

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

c)

```
>> A_inverzno=inv(A)

A_inverzno =

   -1.0667    0.1333    0.6000
    0.3333   -0.1667    0.0000
    0.1333    0.2333   -0.2000
```

d)

```
>> A_adjungovano=A_inverzno*determinanta

A_adjungovano =

  -32.0000    4.0000   18.0000
   10.0000   -5.0000    0.0000
    4.0000    7.0000   -6.0000
```

e)

```
>> max_vred=max(A)

max_vred =

     3     8     6
```

f)

```
>> sred_vred=mean(A)

sred_vred =

    2.0000    5.6667    4.3333
```

g)

```
>> A-A'

ans =

     0    -3     0
     3     0     2
     0    -2     0

>> A-3

ans =

    -2    -1     0
     2     1     5
     0     3     1
```

```
>> 3*A

ans =

     3     6     9
    15    12    24
     9    18    12




>> A^3

ans =

         253         338         408
         656         868        1055
         534         744         859
```

h)

```
>> K=[A(1,:)+i.*A_transponovano(3,:)]

K =

   1.0000 + 3.0000i   2.0000 + 6.0000i   3.0000 + 4.0000i
```

2.9.

```
>> A=[5,4,-2,6;3,6,6,4.5;6,12,-2,16;4,-2,2,-4]

A =

    5.0000    4.0000   -2.0000    6.0000
    3.0000    6.0000    6.0000    4.5000
    6.0000   12.0000   -2.0000   16.0000
    4.0000   -2.0000    2.0000   -4.0000

>> B=[4;13.5;20;6]

B =

    4.0000
   13.5000
   20.0000
    6.0000

>> X=inv(A)*B

X =

   -0.6667
   31.6667
```

```
  -11.3333
  -23.6667
```

2.10.

```
>> r=[2,4,6]

r =

     2     4     6

>> h=[3,6,9]

h =

     3     6     9

>> V=pi*r.^2.*h

V =

   1.0e+03 *

    0.0377    0.3016    1.0179
```

2.11.

```
>> n=0:1:50;
>> K=1./((2.*n+1).*(2.*n+3));
>> suma1=sum(K)

suma1 =

    0.4951

>> n=0:1:500;
>> K=1./((2.*n+1).*(2.*n+3));
>> suma2=sum(K)

suma2 =

    0.4995

>> n=0:1:5000;
>> K=1./((2.*n+1).*(2.*n+3));
>> suma3=sum(K)

suma3 =

    0.5000
```

3. GRAFIČKO PRIKAZIVANJE PODATAKA

MATLAB ima veoma razvijene različite tehnike za grafičko prikazivanje podataka. Grafički sistem MATLAB-a čine dva nivoa funkcija. Prvom nivou pripadaju visoko razvijene funkcije za prikazivanje 2D i 3D podataka. Drugi, niži nivo grafičkog sistema MATLAB-a čine funkcije kojima se definiše izgled grafikona (boja, orjentacija koordinatnih osa, više grafika u jednom prozoru, itd.). Korišćenjem funkcija prvog i drugog nivoa postiže se jasno i efektno grafičko predstavljanje podataka.

3.1 Dvodimenzionalni (2D) grafikoni

MATLAB ima više ugrađenih funkcija za pravljenje različitih vrsta 2D grafikona. Među njima su standardni grafici sa linearnom podelom na osama, grafici sa logaritamskim podelama na osama, trakasti, stepenasti, polarni, i još mnogo drugih. Izgled grafikona može se podesiti formatiranjem. Može se izabrati boja, debljina i vrsta linija, dodavati markeri i linije mreže, ispisivati nazivi grafikona i komentari na grafikonu. Više grafika može biti nacrtano na istom grafikonu, kao i više grafikona na istom grafičkom prozoru.

Komanda `plot` služi za crtanje dvodimenzionalnih grafikona. Najjednostavniji oblik te komande je:

`plot(x)` – crta elemente vektora x u funkciji indeksa
`plot(x,y)` – crta elemente vektora `y` u funkciji elemenata vektora `x`

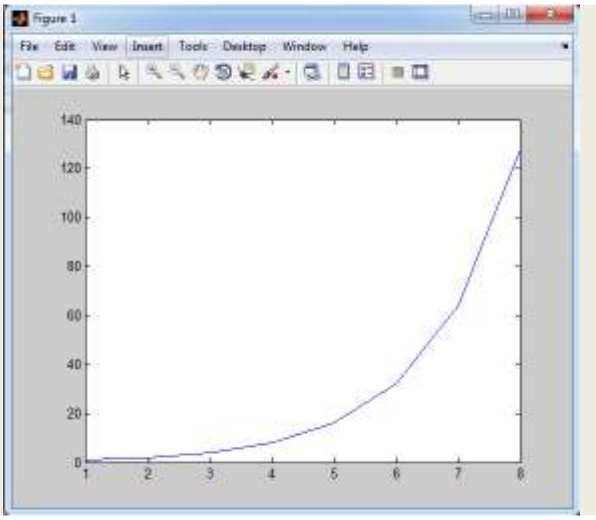
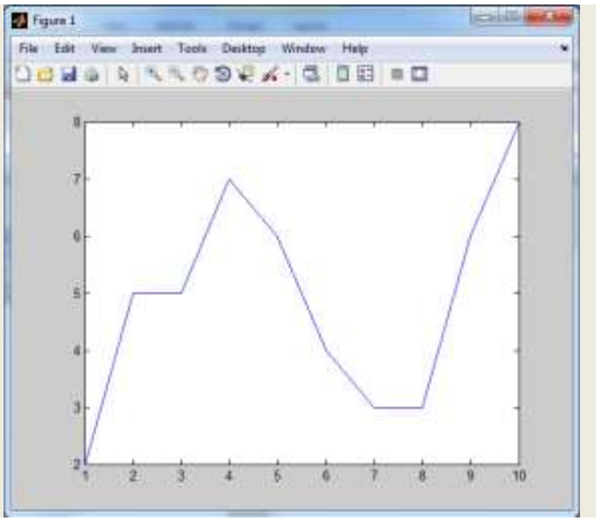
Argumenti `x` i `y` su vektori, tj. jednodimenzionalni nizovi. Oba vektora moraju imati isti broj elemenata. Izvršavanjem komande `plot` otvara se grafički prozor (Figure) u kome se prikazuje grafik. Grafik se sastoji od pravolinijskih segmenata između tačaka čije su koordinate definisane elementima vektora `x` i `y`.

Primer 3.1: Ako je x vektor sa elementima x: 1, 2, 4, 8, 16, 32, 64 i 128 primenom komande plot(x) dobija se:

```
>> x=[1,2,4,8,16,32,64,128];
>> plot(x)
```

Primer 3.2: Ako su elementi vektora X: 1, 2, 3, 4, 5, 6, 7, 8, 9 i 10, a vektora Y: 4, 5, 1, 8, 7, 2, 9, 3, 6 i 2, grafik promene Y u zavisnosti od X je:

```
>> X=1:10; Y=[2,5,5,7,6,4,3,3,6,8];
>> plot(X,Y)
```

Komanda `plot` ima opcione argumente za zadavanje boje i stila linije, boje i vrste markera kojim se crta grafik:

```
plot(x,y,'oznaka linije','ime svojstva',vrednost svojstva)
```

Oznake linije su opcione

Oznake stila linije

Vrsta linije	Oznaka
puna (podrazumevana)	-
isprekidana	--
tačkasta	:
crta-tačka	-.

Oznake boje linije

Boja	Oznaka
crvena	r
zelena	g
plava	b
žuta	y

Oznaka vrste markera

Marker	Oznaka
plus	+
kružić	o
zvezdica	*
tačka	.

Primeri:

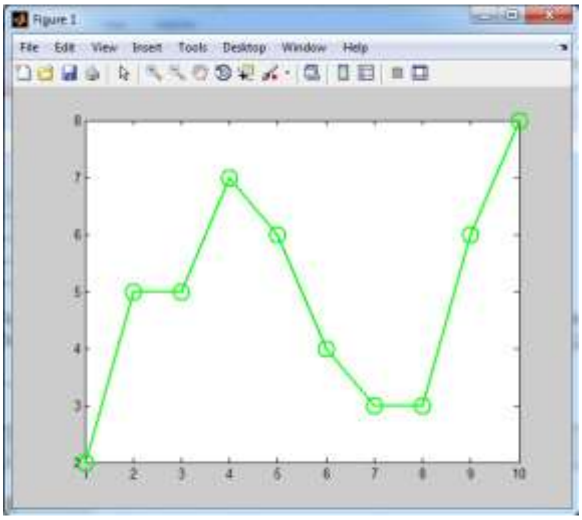
`plot(x,y)`	Puna plava linija (podrazumevano)
`plot(x,y,'r')`	Puna crvena linija
`plot(x,y,'--y')`	Žuta ispregikdana linija
`plot(x,y,'*')`	Tačke označene zvezdicama (nisu povezane linijama)

Ime svojstva i vrednost svojstva su opcioni.

Ime svojstva	Opis	Moguće vrednsoti
`linewidth`	Zadaje debljinu linije	Broj (podrazumevano 0.5)
`markersize`	Zadaje veličinu markera	Broj

Primer 3.3: Nacrtati grafik vektora X i Y iz Primera 3.2 u kome su tačke povezane punom zelenom linijom i označene markerima u obliku kružića. Debljina linije je 2 a veličina markera 12.

```
>> X=1:10; Y=[2,5,5,7,6,4,3,3,6,8];
>> plot(X,Y,'-go','linewidth',2,'markersize',12)
```



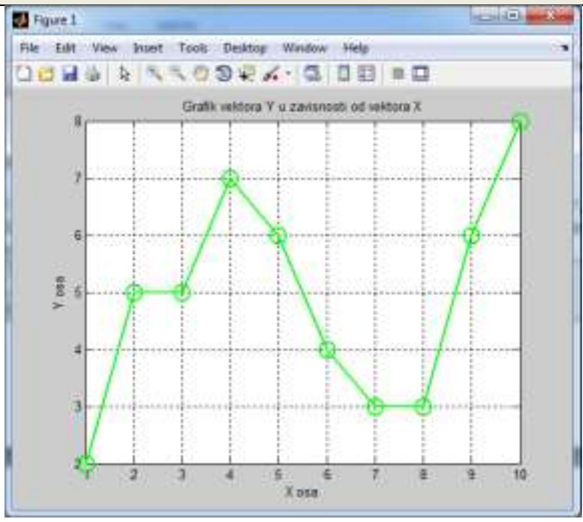
Formatiranje grafikona

Komande `xlabel`, `ylabel` i `title` služe za označavanje x ose, y ose i imena grafikona, respektivno.

```
xlabel('oznaka x ose')
ylabel('oznaka y ose')
title('oznaka grafikona')
```
`grid on` – postavljanje mreže

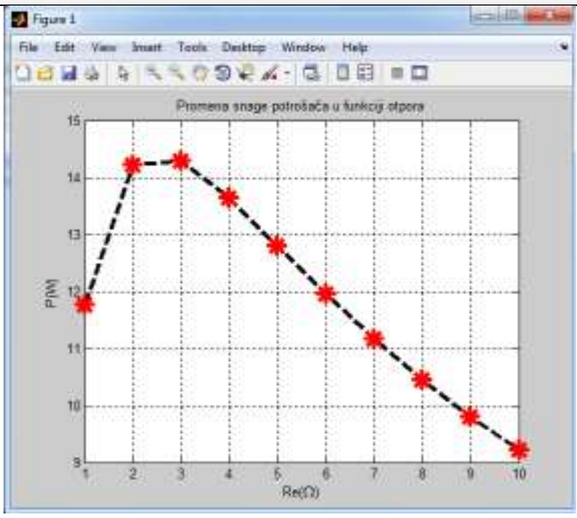
Primer 3.4: U grafiku iz Primera 3.3, označiti osu x sa “X osa“, osu y sa “Y osa“, i imenovati grafikon sa: “Grafik vektora Y u zavisnosti od vektora X“. Postaviti mrežu radi lakšeg očitavanja podataka sa grafika.

```
>> X=1:10; Y=[2,5,5,7,6,4,3,3,6,8];
>> plot(X,Y,'-go','linewidth',2,'markersize',12);
>> xlabel('X osa'); ylabel('Y osa');
>> title('Grafik vektora Y u zavisnosti od vektora X');
>> grid on
```



Primer 3.5 Električno kolo sadrži naponski izvor E, unutrašnjeg otpora R_E i otpornik potrošača R_P. Snaga koja se na potrošaču pretvara u toplotu je: $P = R_P \frac{E^2}{(R_P + R_E)^2}$. Nacrtati snagu P u funkciji od otpora potrošača R_P za $1\Omega \le R_P \le 10\Omega$, ako je $E = 12V$ i $R_E = 2.5\Omega$. Izgled grafika podesiti tako da bude crna isprekidana linija, debljine 3, tačke markirati zvezdicama crvene boje veličine 15. Označiti odgovarajuće ose sa P (W) i R_P (Ω), grafik nasloviti sa: “Promena snage potrošača u funkciji otpora”.

```
>> E=12;Re=2.5;Rp=[1:10];
>> P=Rp.*E^2./(Rp+Re).^2;
>> plot(Rp,P,'--k*','linewidth',3,'markersize',15, 'markeredgecolor','r');
>> xlabel('Re(\Omega)')
>> ylabel('P(W)');
>> title('Promena snage potrošača u funkciji otpora');
>> grid on
```



Napomena: Grčko slovo se uključuje u tekst upsivanjem: `\englesko ime slova` (sve to kao znakovni niz). Npr.: `'\alpha'` – α, `'\gamma'` – γ, `'\theta'` – θ, `'\pi'` – π, `'\Delta'` – Δ, itd.

Specijalni grafikoni.

`bar(x,y)` vertikalni trakasti grafikon; `barh(x,y)` horizontalni trakasti grafikon; `stairs(x,y)` stepenasti grafikon; `stem(x,y)` grafikon diskretnih podataka.

Primer 3.6: Prikazati vektor $x = \left[4/3 \quad \sin\frac{\pi}{3} \quad \sqrt{22} \quad \ln 10 \right]$ u obliku vertikalnog trakastog, horizontalnog trakastog, stepenastog i grafikona diskretnih podataka.

```
>> x=[4/3,sin(pi/3),sqrt(22),log(10)];
>> bar(x)
>> barh(x)
>> stairs(x)
>> stem(x)
```

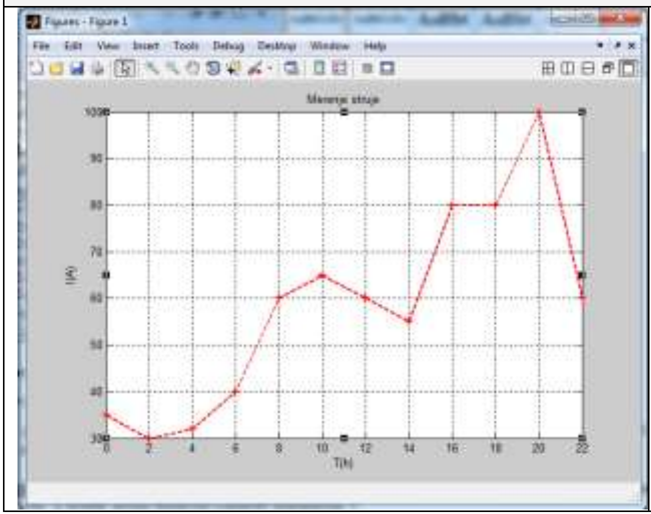
Primer 3.7 U jednom eksperimentu je u toku 24 časa, na svaka 2 časa merena struja jednog potrošača. Zabeležene su sledeće vrednosti:

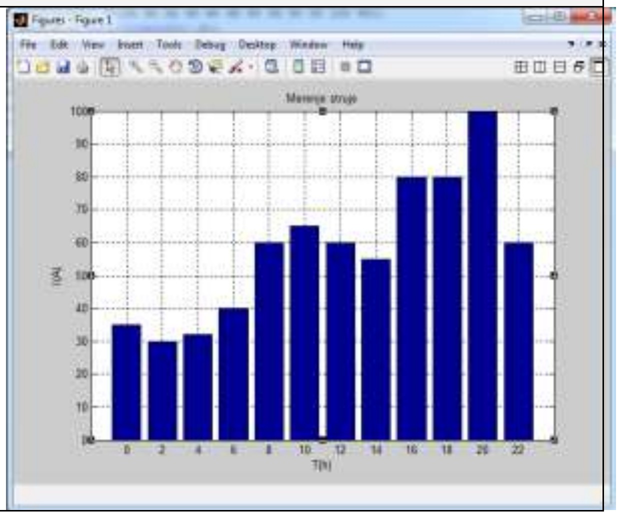
T [h]	0	2	4	6	8	10	12	14	16	18	20	22
I [A]	35	30	32	40	60	65	60	55	80	80	100	60

a) Odrediti maksimalnu i minimalnu vrednost izmerene struje kao i vreme (čas) u kome su izemerene te vrednosti

b) Odrediti srednju vrednost struje u toku celog perioda
c) Grafički prikazati izmerene stuje u funkciji vremena na kontinualnom (plot) dijagramu sa isprekidanim linijama crvene boje, debljine linije 2, sa markerima u obliku znaka +. Dijagram nasloviti sa "Merenje struje" . Osu x označiti sa T (h) a y osu sa I (A).
d) Grafički prikazati izmerene stuje u funkciji vremena na trakastom (bar) dijagramu plave boje. Dijagram nasloviti sa "`Merenje struje`" . Osu x oznaciti sa T (h) a y osu sa I (A).

```
>> T=[0:2:22]; I=[35 30 32 40 60 65 60 55 80 80 100 60];
>> [Imax Tmax]=max(I);
>> [Imin Tmin]=min(I);
>> Isrednje=mean(I);
>> plot(T,I,'-r+','linewidth',2);
>> xlabel('T(h)'); ylabel('I(A)'); title('Merenje struje'); grid on;
>> bar(T,I);
>> xlabel('T(h)'); ylabel('I(A)'); title('Merenje struje'); grid on;
```





Crtanje grafikona sa više grafika

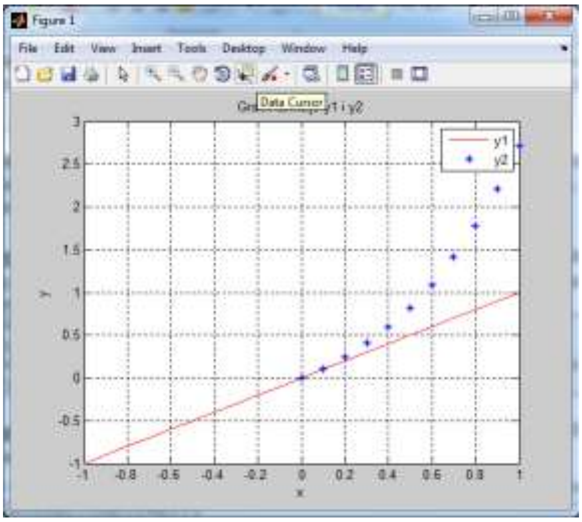
Često je potrebno nacrtati više grafika na istom grafičkom prozoru. Najjednostavniji način za to je upotreba komande plot na sledeći način:

```
plot(x,y,'b',u,v,'g',t,h,'r')
```

Na ovaj način se dobijaju tri grafika na istom grafičkom prozoru: `y` u zavisnosti od `x` (plavom bojom), `v` u yavisnosti od `u` (zelenom bojom), `h` u zavisnosti od `t` (crvenom bojom). Vektori svakog para moraju biti iste dužine.

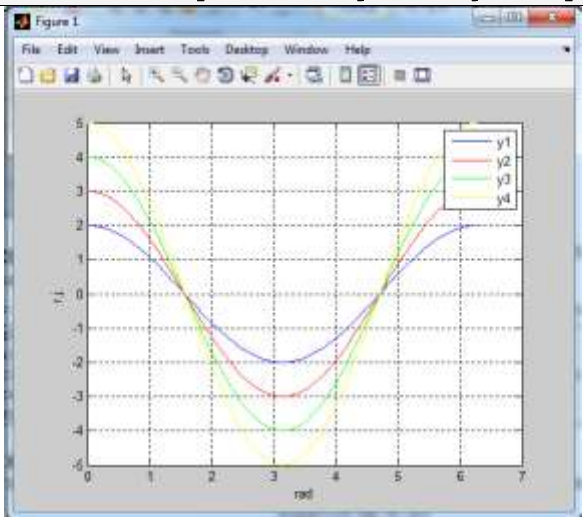
Primer 3.8: Nacrtati grafike funkcija $y = x$ za $-1 \le x \le 1$; i $y = x \cdot e^x$ za $0 \le x \le 1$ u okviru jednog grafikona. Grafik prve funkcije neka bude crvene boje, a grafik druge funkcije označiti markerom * .

```
>> x1=[-1:0.1:1]; y1=x1; x2=[0:0.1:1]; y2=x2.*exp(x2);
>> plot(x1,y1,'r',x2,y2,'*');
>> xlabel('x'); ylabel('y'); title('Grafik funkcije y1 i y2');
>> legend('y1','y2'); grid on
```



Primer 3.9: Generisi vektor x sa vrednostima od 0 do 2π sa inkrementom $\pi/100$, a zatim na istom dijagramu nacrtaj dijagrame sledećih funkcija: $y_1 = 2\cos(x)$; $y_2 = 3\cos(x)$; $y_3 = 4\cos(x)$; $y_4 = 5\cos(x)$. Radi bolje čitljivosti formirati mrežasti dijagram.

```
>> x=[0:pi/100:2*pi];
>> y1=2.*cos(x);y2=3.*cos(x);y3=4.*cos(x);y4=5.*cos(x);
>> plot(x,y1,'b',x,y2,'r',x,y3,'g',x,y4,'y');
>> xlabel('rad');ylabel('r.j.');legend('y1','y2','y3','y4');grid on
```



Crtanje više grafikona u istom grafičkom prozoru

Na istoj stranici (istom grafičkom prozoru) može se nacrtati više grafikona pomoću komande:

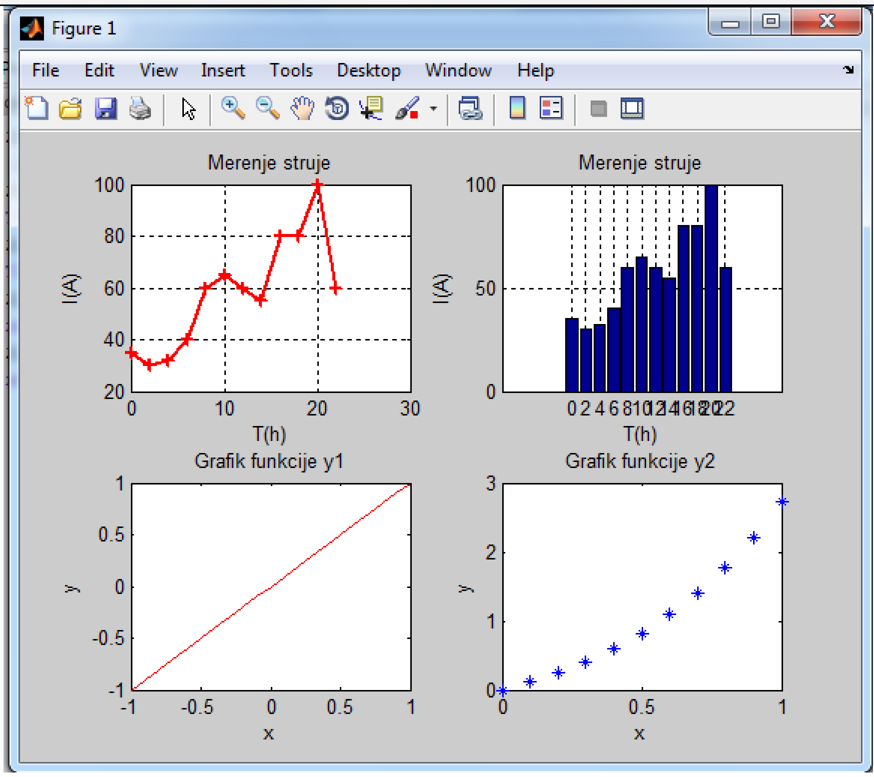
`subplot(m,n,p)`

Ova komanda deli grafički prozor na `m*n` pravouganih podgrafikona. Broj `p` označava tekući podgrafikon u koji se smešta grafikon koji se crta sledećom komandom za crtanje (`plot, bar, ...`).

Primer 3.10: Grafikone iz Primera 3.7 i 3.8 nacrtati u istom grafičkom prozoru.

```
>> T=[0:2:22]; I=[35 30 32 40 60 65 60 55 80 80 100 60];
>> x1=[-1:0.1:1]; y1=x1; x2=[0:0.1:1]; y2=x2.*exp(x2);
>> subplot(2,2,1),plot(T,I,'-r+','linewidth',2),...
xlabel('T(h)'),ylabel('I(A)'),title('Merenje struje'),grid on;
>> subplot(2,2,2),bar(T,I),...
xlabel('T(h)'),ylabel('I(A)'),title('Merenje struje'),grid on;
```

```
>> subplot(2,2,3),plot(x1,y1,'r'), ...
xlabel('x'),ylabel('y'),title('Grafik funkcije y1');
>> subplot(2,2,4), plot(x2,y2,'*'), ...
xlabel('x'),ylabel('y'),title('Grafik funkcije y2');
```



Crtanje grafika funkcije komandom `fplot`

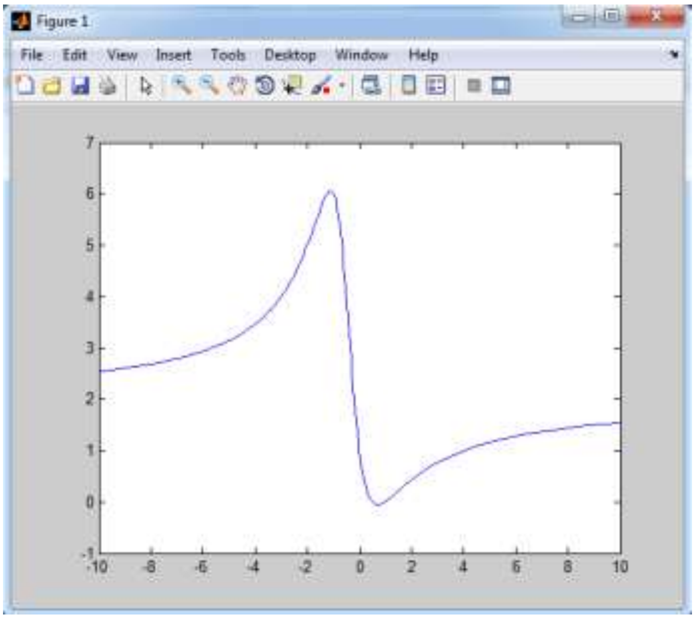
Pomoću komande `fplot` crta se funkcija oblika *y* = *f*(*x*) u datom opsegu promenljive *x*. Komanda ima opšti oblik:

```
fplot('funkcija',granice,oznaka linije)
```

`'funkcija'` – predstavlja MATLAB kod funkcije u obliku znakovnog niza. Može sadržati sve ugrađene MATLAB-ove funkcije kao i one koje je prethodno definisao sam korisnik.
`granice` – Granice su vektor u obliku `[xmin,xmax]` koji zadaje domen promenljive *x*, ili vector sa četiri elementa u obliku `[xmin,xmax,ymin,ymax]` koji definiše i domen x I domen y (granice na x i y osi).
`oznaka linije` – kao kod komande `plot`

<u>Primer 3.11:</u> Nacrtati grafik funkcije $f(x)=\dfrac{2x^2-3x+1}{x^2+x+1}$ za $-10 \le x \le 10$.

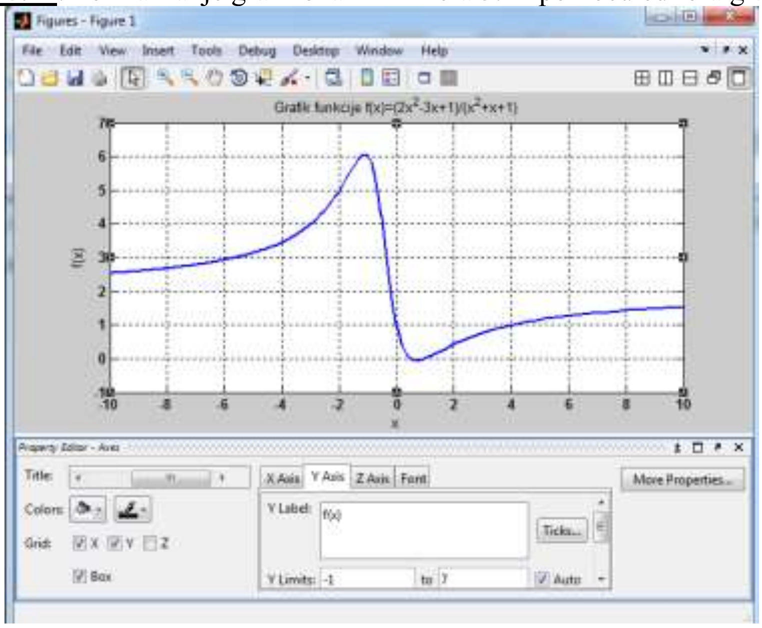
```
>> fplot('(2*x^2-3*x+1)/(x^2+x+1)',[-10,10]);
```



Formatiranje grafikaona u editoru grafikona

Grafikon se najlakše može formatirati pomoću interaktivnog editora u okviru samog grafičkog prozora. Ovaj editor se aktivira izborom strelice sa menija grafikona i dvostrukim klikom na grafikon. Na ovaj način moguće je potpuno formatirati grafikon od početka, ili izmeniti postojeća podešavanja koja su definisana pomoću komandi u programu ili komandnom prozoru.

Primer 3.12: Formatiranje grafikona iz Primera 3.11 pomoću editora grafikona u grafičkom prozoru



3.2 *Trodimenzionalni (3D) grafikoni*

Trodimenzionalni (3D) grafikoni se koriste za prikazivanje podataka u trodimenzionalnom prostoru. MATLAB ima više funkcija za prikazivanje različitih tipova 3D grafikona, i to linijskih, površinskih, mrežastih, i još mnogo drugih. Ovi grafikoni se, slično kao i 2D grafikoni, mogu lako formatirati tako da imaju specifičan izgled, a mogu im se dodavati i specijalni efekti. U nastavku je dat pregled najčešće korišćenih tipova 3D grafikona u MATLABU i funkcije za njihovu realizaciju.

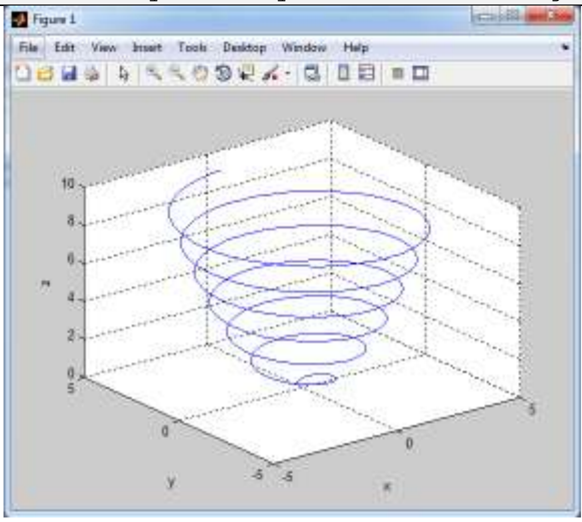
Trodimenzinalni linijski grafikon je linija dobijena povezivanjem tačaka u trodimenzionalnom prostoru. Crtanje linijskog 3D grafikona ostvaruje se pomoću komande:

```
plot3(x,y,z)
```

`x,y i z` su vektori čiji su elementi koordinate tačaka u trodimenzionalnom prostoru.

Primer 3.13: Ako su date koordinate *x*, *y* i *z* u funkciji paramatra *t* sledećim funkcijama: $x=\sqrt{t}\sin(2t)$; $y=\sqrt{t}\cos(2t)$; $z=0{,}5t$. Nacrtati linijski 3D grafikon tačaka u opsegu $0\le t\le 6\pi$.

```
>> t=0:pi/100:6*pi;
>> x=sqrt(t).*sin(2*t);
>> y=sqrt(t).*cos(2*t);
>> z=0.5*t;
>> plot3(x,y,z);
>> xlabel('x');ylabel('y');zlabel('z'); grid on
```



Mrežasti i površinski grafikoni 3D grafikoni

Ovi grafikoni omogućavaju predstavljanje funkcije tipa *z=f(x,y)*, gde su x i y nezavisne promenljive, a z je zavisna promenljiva. To znači da se u datom domenu vrednost promenljive z može izračunati za svaku kombinaciju x i y. Mrežasti i površinski grafkoni se crtaju u tri koraka:

1. Najpre se za dati opseg nezavisno promenljivih *x* i *y* formira mreža (rešetka) tačaka, koje predstavljaju oblast definisanosti funkcije *z*.

   `[X,Y]=meshgrid(x,y)`

2. Zatim se izračunaju vrednosti funkcije *z* u svim tačkama rešetke.

   `Z=F(X,Y)`

3. Kada su određene sve tačke u prostoru, crtaju se grafikoni.

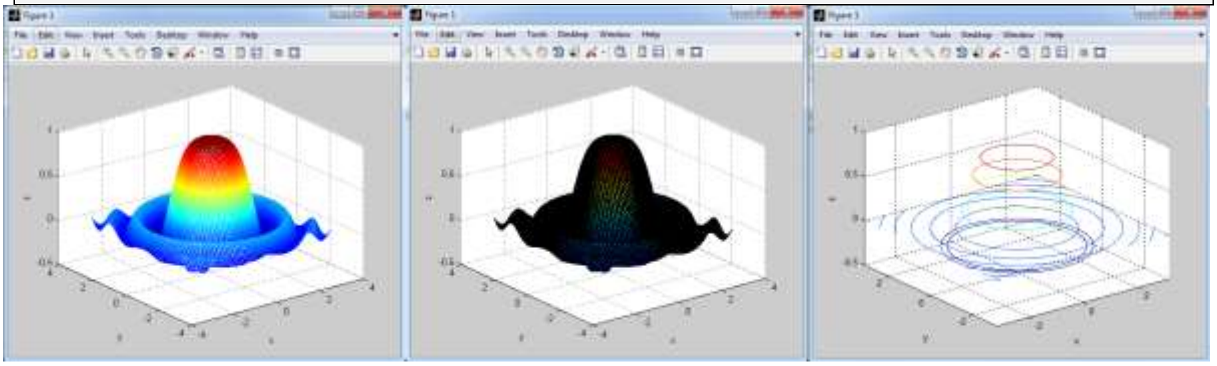
   Mrežasti: `mesh(X,Y,Z)`
   Površinski: `surf(X,Y,Z)`

Primer 3.14: Nacrtati 3D mrežasti, površinski i konturni grafik funkcije dve promenljive:

$$z=\frac{\sin\left(x^2+y^2\right)}{x^2+y^2} \quad \text{za} \quad -\pi\le x,y\le\pi$$

```
>> x=-pi:pi/100:pi; y=-pi:pi/100:pi;
>> [X,Y]=meshgrid(x,y);
>> Z=sin(X.^2+Y.^2+eps)./(X.^2+Y.^2+eps);
>> mesh(X,Y,Z); xlabel('x');ylabel('y');zlabel('z');
```

```
>> surf(X,Y,Z); xlabel('x');ylabel('y');zlabel('z');
>> contour3(X,Y,Z); xlabel('x');ylabel('y');zlabel('z');
```



Primer 3.15: U istom grafičkom prozoru nacrtati sledeće funkcije:

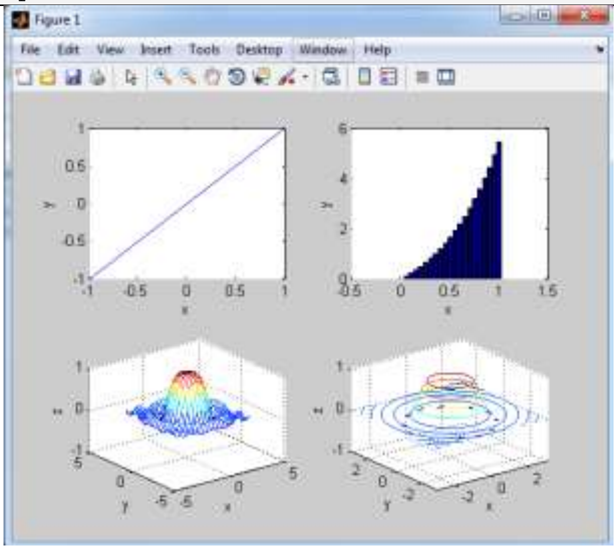
$y = x$ za $-1 \le x \le 1$ u vidu kontinualnog linijskog grafikona

$y = x \cdot e^x$ za $0 \le x \le 1$ u vidu vertikalnog trakastog grafikona

$z = \frac{\sin(x^2 + y^2)}{x^2 + y^2}$ za $-\pi \le x, y \le \pi$ u vidu trodimenzionalnog mrežastog grafikona

$z = \frac{\sin(x^2 + y^2)}{x^2 + y^2}$ za $-\pi \le x, y \le \pi$ u vidu konturnog grafika po nivoima na z-osi

```
>> x1=-1:0.1:1;y1=x1;
>> subplot(2,2,1),plot(x1,y1);xlabel('x');ylabel('y');
>> x2=0:0.05:1;y2=2.*x2.*exp(x2);
>> subplot(2,2,2),bar(x2,y2);xlabel('x');ylabel('y');
>> [X,Y]=meshgrid(-pi:pi/10:pi,-pi:pi/10:pi);
>> Z=sin(X.^2+Y.^2+eps)./(X.^2+Y.^2+eps);
>> subplot(2,2,3),mesh(X,Y,Z);xlabel('x');ylabel('y');zlabel('z');
>> subplot(2,2,4),contour3(X,Y,Z);xlabel('x');ylabel('y');zlabel('z');
```



3.3 *Zadaci za vežbanje*

Zadatak 3.1: Podaci o satnim dotocima u akumulaciji jedne hidroelektrane su dati u tabeli.

T (h)	1	2	3	4	5	6	7	8
Q (m^3/s)	750	800	750	700	600	550	600	650

Nacrtati promenu Q sa vremenom u formi kontilnualnog, trakastog, stepenastog i diskretnog grafikona. Označiti ose i nasloviti grafikone na odgovarajući način.

Zadatak 3.2: U toku dana su na jednoj lokaciji na svaki sat merene vrednosti brzine vetra, temperature i solarne iradijacije. Izmerene vrednosti su date u sledećoj tabeli.

t (h)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
v(m/s)	10.8	10.7	11.3	10.9	11.5	11.6	10.7	10.9	10.1	10.4	10.7	11.6	11.6	11.9	11.5	11.3	11.5	11.7	12.2	11.7	11.5	11.3	11.3	11.3
Ta(°C)	15	15	14	13	14.5	15	15.2	17	17	17.5	18	19	20	21	21	21	21	21	20	19	18	17	16	15.5
Is (W/m^2)	0	0	0	0	90	220	250	450	650	680	820	850	830	850	680	600	250	200	150	70	50	0	0	0

a) Odrediti maksimalne i minimalne vrednosti izmerenih veličina kao i vreme (čas) u kome su izemerene te vrednosti.
b) Odrediti srednje vrednosti ovih veličina u posmatranom periodu.
c) Grafički prikazati izmerene veličine na tri grafikona u okviru jednog grafičkog prozora (podeliti grafički prozor na tri dela). Brzinu vetra prikazati kao kontinualni grafik, temperaturu ambijenta kao trakasti vertikalni grafik i solarnu iradijaciju kao grafik diskretnih vrednosti. Označiti ose odgovarajućim veličinama.

Zadatak 3.3: Nacrtati grafike funkcija x, x^3, e^x i e^{x^2} u intervalu $0 \le x \le 4$ na istom grafikonu, i to:

a) u Dekartovom pravouglom koordinatnom sistemu
b) u logaritamskoj razmeri po y osi (koristiti komandu `semilogy(x,y)`)
c) u logaritamskoj razmeri po x i y osi, (koristiti komandu `loglog(x,y)`)

Zadatak 3.4: Komandom `fplot` nacrtati funkciju:

$$f(x) = -0.01x^5 + 0.03x^4 - 0.4x^3 + 2x^2 - 6x + 5$$

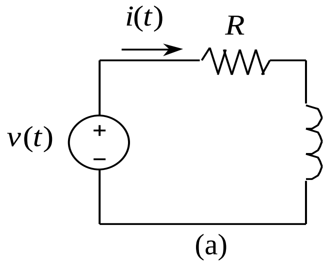
u domenu $-4 \le x \le 6$. Označiti koordinatne ose na odgovarajući način.

Zadatak 3.5: Nacrtati funkciju $y = 4\cos(x) + 2x^2$, njen prvi i drugi izvod na istom grafikonu, za $-2\pi \le x \le 2\pi$. Funkciju nacrtati punom linijom crne boje, prvi izvod isprekidanom linijom crvene boje, a drugi izvod linijom crta tačka plave boje. Označiti koordinatne ose, nasloviti grafikon i dodati legende.

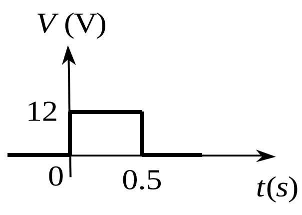
Zadatak 3.6: Otpornik R = 4 (Ω) i kalem L = 1.3 (H) redno su povezani sa naponskim izvorom, kao na slici (a) (RL kolo). Kada se na krajevima naponskog izvora pojavi pravougaoni naponski impuls visine V = 12 (V) i trajanja 0.5(s), prikazan na slici (b), struja kroz kolo kao funkcija vremena je:

$$i(t) = \frac{V}{R}\left(1 - e^{(-Rt)/L}\right) \quad \text{za } 0 \le t \le 0.5\text{s}$$

$$i(t) = e^{-(Rt)/L}\,\frac{V}{R}\left(e^{(0.5R)/L} - 1\right) \quad \text{za } 0.5s \le t$$



(a)



(b)

Nacrtati grafik struje u funkciji vremena, za $0 \le t \le 2$ s.

Zadatak 3.7: Nacrtati funkciju $z = 1.8^{-1.5\sqrt{x^2+y^2}}\sin(x)\cos(0.5y)$ u opsegu $-3 \le x \le 3$ i $-3 \le y \le 3$ u obliku mrežastog, površinskog i 3D konturnog grafikona.

Zadatak 3.8: Nacrtati 3D mrežaste grafikone za sledeće dve funkcije:

(a) $z = \sin(x)\sin(y)$ u opsegu $-3\pi \le x \le 3\pi$ i $-3\pi \le y \le 3\pi$

(b) $z = (x^2 + y^2)\cos(x^2 + y^2)$ u opsegu $-1 \le x \le 1$ i $-1 \le y \le 1$

Zadatak 3.9: Dva tačkasta naelektrisanja $q_1=4 \cdot 10^{-10}$ (C) i $q_2=6 \cdot 10^{-10}$ (C), nalaze se u ravni *x*-*y* u tačkama (0.3, 0, 0) i (-0.3, 0, 0). Izračunati električni potencijal u tačkama ravni *x*-*y* za $-0.2 \le x \le 0.2$ i $-0.2 \le y \le 0.2$ Podesiti grafikon tako da osa *z* prikazuje (predstavlja) vrednost električnog potencijala.
(Podsetnik: Izraz za elekrični potencijal tačkastog naelektrisnja *q* u tački na rastojanju *r* je $V = \frac{q}{4\pi\varepsilon_0 r}$, gde je $\varepsilon_0 = 8.854 \cdot 10^{-12} \frac{C}{Nm^2}$ dielektrička konstanta vakuuma. Ukupni potencijal u nekoj tački se dobija kao superpozicija (suma) potencijala pojedinačnih naelektrisanja).

Zadatak 3.10: Položaj čestice u kretanju je dat kao funkcija vremena preko sledećih jednačina:

$$x = (2 + 4\cos(t))\cos(t)$$
$$y = (2 + 4\cos(t))\sin(t)$$
$$z = t^2$$

Nacrtati grafikon položaja čestice za period $0 \le t \le 20$.

Zadatak 3.11: Posmatra se redno RLC kolo sa izvorom naizmeničnog napona. Napon izvora dat je formulom: $v_s = V_m \sin(\omega t)$, gde je $\omega = 2\pi f$ a f frekvencija. Amplituda struje u ovom kolu se izračunava prema izrazu:

$$I_m = \frac{V_m}{\sqrt{R^2 + \left(\omega L - \frac{1}{\omega C}\right)^2}}$$

gde je *R* otpornost otpornika, *C* kapacitivnost kondenzatora i *L* induktivnost kalema. U ovom konkretnom slučaju je: $C = 15 \cdot 10^{-6} F$, $L = 240 \cdot 10^{-3} H$ i V_m=24 *V*.

a) Nacrtati 3D grafikon struje I_m (osa *z*) kao funkciju od ω (osa *x*) za $60Hz \le f_d \le 110Hz$ i *R* (osa *y*) za $10\Omega \le R \le 40\Omega$.
b) Predstaviti projekciju grafikona na ravan *x*-*z* upotrebom komande `view(az,el)`. Proceniti na osnovu tog grafikona prirodnu (sopstvenu) frekvenciju kola (frekvenciju na kojoj je I_m ima maksimum). Uporedite procenu sa izračunatom vrednošću izraza $\frac{1}{2\pi\sqrt{LC}}$.

Rešenje:

3.1.

```
>> T=1:8;
>> Q=[750 800 750 700 600 550 600 650];
>> plot(T,Q);
>> xlabel('T(h)'); ylabel('Q(m^3/s)'); title('Merenje dotoka - kont. grafik');

>> bar(T,Q);
```

```
>> xlabel('T(h)'); ylabel('Q(m^3/s)'); title('Merenje dotoka - trakasti grafik');

>> stairs(T,Q);
>> xlabel('T(h)'); ylabel('Q(m^3/s)'); title('Merenje dotoka - step. grafik');

>> stem(T,Q);
>> xlabel('T(h)'); ylabel('Q(m^3/s)'); title('Merenje dotoka - diskr. grafik');
```

3.2.

```
>> t=1:24;
>> v=[10.8   10.7   11.3   10.9   11.5   11.6   10.7   10.9   10.1   10.4   10.7   11.6
      11.6   11.9   11.5   11.3   11.5   11.7   12.2   11.7   11.5   11.3   11.3
      11.3];
>> Ta=[15    15     14     13     14.5   15     15.2   17     17     17.5   18     19
      20     21     21     21     21     21     20     19     18     17     16
      15.5];
>> Is=[0     0      0      0      90     220    250    450    650    680    820    850
      830    850    680    600    250    200    150    70     50     0      0      0];

>> [vmax Tmax]=max(v)
vmax =
  12.2000
Tmax =
   19
>> [vmin Tmin]=min(v)
vmin =
  10.1000
Tmin =
    9
>> vsrednje=mean(v)
vsrednje =
  11.2500
>> [Tamax Tmax]=max(Ta)
Tamax =
   21
Tmax =
   14
>> [Tamin Tmin]=min(Ta)
Tamin =
   13
Tmin =
    4
>> Tasrednje=mean(Ta)
Tasrednje =
  17.5292
>> [Ismax Tmax]=max(Is)
Ismax =
  850
Tmax =
   12
>> [Ismin Tmin]=min(Is)
Ismin =
    0
```

```
Tmin =
    1
>> Issrednje=mean(Is)
Issrednje =
  320.4167

>> subplot(3,1,1),plot(t,v),xlabel('t(h)'),ylabel('v(m/s)'),title('Brzina vetra u funkciji vremena')
>> subplot(3,1,2),bar(t,Ta),xlabel('t(h)'),ylabel('Ta(^oC)'),title('Temperatura ambijenta u funkciji vremena')
>> subplot(3,1,3),stem(t,Is),xlabel('t(h)'),ylabel('Is(W/m^2)'),title('Solarna iradijacija u funkciji vremena')
```

3.3.

```
>> x=0:0.1:4;
>> y1=x;
>> y2=x.^3;
>> y3=exp(x);
>> y4=exp(x.^2);
>> plot(x,y1,'r',x,y2,'b',x,y3,'y',x,y4,'g'),grid on
>> semilogy(x,y1,'r',x,y2,'b',x,y3,'y',x,y4,'g'),grid on
>> loglog(x,y1,'r',x,y2,'b',x,y3,'y',x,y4,'g'),grid on
```

3.4.

```
>> fplot('-0.01*x^5+0.03*x^4-0.4*x^3+2*x^2-6*x+5',[-4,6]), xlabel('x'),ylabel('f(x)')
```

3.5.

```
>> x=-2*pi:0.01:2*pi;
>> y=4.*cos(x)+2.*x.^2;
>> y1=-4*sin(x)+4.*x;
>> y2=-4*cos(x)+4;
>>plot(x,y,'-k',x,y1,'--r',x,y2,'-.b'), xlabel('x'),ylabel('y,dy/dx,d^2y/dx^2'),title('Funkcija y i njeni prvi i drugi izvod'),legend('y=4*cos(x)+2*x^2','dy/dx','d^2y/dx^2');grid on;
```

3.6

```
>> R=4;
>> L=1.3;
>> t1=0:0.01:0.5;
>> V1=12;
>> i1=V1/R*(1-exp(-R.*t1./L));
>> t2=0.5:0.01:2;
>> V2=0;
>> i2=V2/R*(1-exp(-R.*t2./L))
>> plot(t1,i1,'r',t2,i2,'b');
>> xlabel('t(s)'), ylabel('i(A)'), title('Zavisnost struje u rednom RL kolu u funkciji vremena');
>> legend('i za 0<=t<=0.5 s','i za t>0.5 s');
```

3.7.

```
>> x=-3:0.1:3;
>> y=-3:0.1:3;
>> [X,Y]=meshgrid(x,y);
>> Z=1.8.^(-1.5*sqrt(X.^2+Y.^2)).*sin(X).*cos(0.5*Y);
>> mesh(X,Y,Z); xlabel('x');ylabel('y');zlabel('z');
>> surf(X,Y,Z); xlabel('x');ylabel('y');zlabel('z');
>> contour3(X,Y,Z); xlabel('x');ylabel('y');zlabel('z');
```

3.8.

a)

```
>> x=-3*pi:pi/20:3*pi;
>> y=-3*pi:pi/20:3*pi;
>> [X,Y]=meshgrid(x,y);
>> Z=sin(X).*sin(Y);
>> mesh(X,Y,Z); xlabel('x');ylabel('y');zlabel('z');
```

b)

```
>> x=-1:0.01:1;
>> y=-1:0.01:1;
>> [X,Y]=meshgrid(x,y);
>> Z=(X.^2+Y.^2).*cos(X.^2+Y.^2);
>> mesh(X,Y,Z); xlabel('x');ylabel('y');zlabel('z');
```

3.9.

- Zadatak se rešava po sledećim koracima
- U ravni x-y formirana je rešetka koja obuhvata domen $-0{,}2 \le x \le 0{,}2$ i $-0{,}2 \le y \le 0{,}2$.
- Izračunava se rastojanje od svake tačke rešetke do čestica
- Izračunava se električni potencijal u svakoj tački
- Iscrtava se grafikon električnog potencijala

```
>> eps0=8.85e-12;q1=4e-10;q2=6e-10;
>> x=-0.2:0.01:0.2;
>> y=-0.2:0.01:0.2;
>> [X,Y]=meshgrid(x,y);
>> r1=sqrt((X+0.3).^2+Y.^2);
>> r2=sqrt((X-0.3).^2+Y.^2);
>> V=(1/4*pi*eps0)*(q1./r1+q2./r2);
>> mesh(X,Y,V)
>> xlabel('x(m)');ylabel('y(m)');zlabel('V(V)')
```

3.10.

```
>> t=[0:0.1:20];
>> x=(2+4*cos(t)).*cos(t);
>> y=(2+4*cos(t)).*sin(t);
>> z=t.^2;
>> plot3(x,y,z), xlabel('x'), ylabel('y'), zlabel('z'), grid on;
```

3.11.

a)

```
>> C=15*10^(-6); L=240*10^(-3); Vm=24;
>> f=linspace(60,110,1000);
>> R=linspace(10,40,1000);
>> Omega=2*pi*f;
>> [Omega,R]=meshgrid(Omega,R);
>> Im=Vm./sqrt(R.^2+(Omega*L-1./(Omega*C)).^2);
>> mesh(Omega,R,Im), xlabel('\omega (1/s)'), ylabel('R (\Omega)'),
zlabel('Im (A)');
```

b)

```
view(0,0)
```

4. SKRIPT I FUNKCIJSKE DATOTEKE, UPRAVLJANJE TOKOM PROGRAMA

Dosada su sve komande upisivane i izvršavane u komandnom prozoru MATLAB-a. Rešavani su jednostavni zadaci upisivanjem nekoliko redova komandi. Međutim, problem kod ovakvog načina rada je što se komande u komandnom prozoru ne mogu snimiti i ponovo izvršiti. To onemogućava rešavanje složenijih zadataka koji zahtevaju veliki broj komandi (recimo 100 i više redova). Zato se programi pišu i snimaju u prozoru za pisanje programa (editoru) kao skript (komandne) ili funkcijske datoteke.

4.1 Skript datoteke

Komandna ili skript datoteka je niz MATLAB komandi i izraza snimljenih kao poseban program. Ovi programi se mogu menjati i preuređivati neograničen broj puta. Izvršavaju se upisivanjem imena programa u komandni prozor MATLAB-a i pritiskom na taster Enter. Komande se izvršavaju redosledom kako su napisane u datoteci, osim kada je taj redosled promenjen pomoću funkcija za upravljanje tokom programa (relacionih i logičkih operatora, uslovnih iskaza i petlji). Sve promenljive definisane u komandnom prozoru MATLAB-a prosleđuju se u komandnu datoteku i obrnuto. To su tzv. globalne promenljive. Za skript datoteke se još koristi i naziv m datoteke, zato što pri snimanju dobijaju ekstenziju .m. Mogu se pisati i menjati u bilo kom editoru teksta, a potom ih gotove preneti (kopirati) u MATLAB-ov Editor (prozor za pisanje programa).

Pravljenje, snimanje i izvršavanje skript datoteka

Skript datoteke se prave i uređuju u prozoru za pisanje programa (prozor Editor). U meniju **File** odabere se **New** i zatim **M-file**. Nakon toga će se otvoriti prozor za pisanje programa. Tada se može pristupiti kreiranju novog programa, tj. skript datoteke. Odgovarajuće komande se upisuju red po red. Prelazak u novi red se ostvaruje pritiskom na taster enter, pri čemu se svakom redu automatski dodeljuje broj. Obično se na početku skript datoteka pišu komentari (komentar započinje znakom %) koji opisuju program koji sledi.

Da bi se mogla pokrenuti i izvršiti, skript datoteka mora biti snimljena. To se jednostavno ostvaruje naredbom **Save as** iz menija **File.** Tom prilikom se izabere ime skript datoteke i lokacija (folder) gde će biti snimljena. Pravila za imena skript datoteka su ista kao i za imena promenljvih (moraju početi slovom, mogu sadržati kombinaciju slova, brojeva i podvlaka, razlikuju velika i mala slova, ne smeju im se davati imena ugrađenih MATLAB-ovih komandi funkcija i promenljivih).

Skript datoteka se pokreće (izvršava) upisivanjem njenog imena u komandni prozor MATLAB-a i pritiskom na taster **Enter**. Drugi način je da se u prozoru same datoteke (Editoru) pritisne ikonica **Run**. Da bi MATLAB mogao da pokrene komandnu datoteku, ona mora biti u tekućem direktorijumu ili na putanji za pretraživanje. Putanja tekućeg direktorijuma prikazuje se u padajućoj listi **Current Directory** na paleti alatki u komandnom prozoru. Tekući direktorijum se menja u prozoru Current Directory tako što se izabere jedinica diska i direktorijum (folder) u kojem je datoteka snimljena.

Definisanje ulaznih promenljivih u skript datotekama

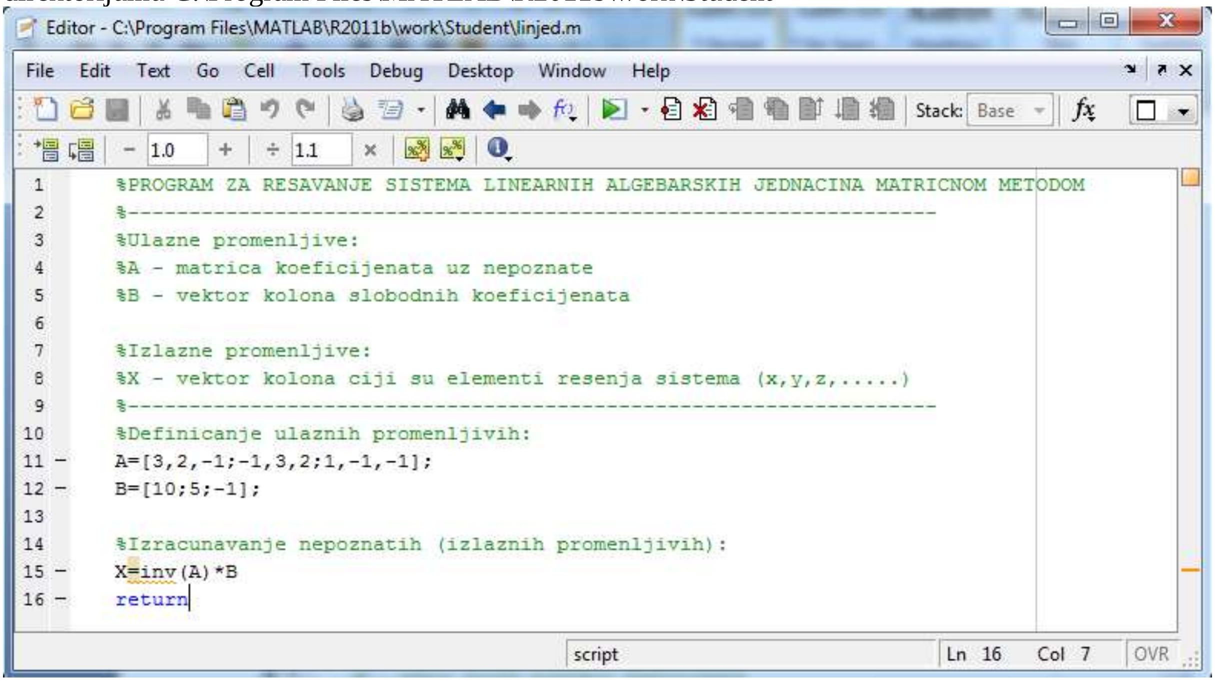
Kada se pokrene komandna datoteka, svim ulaznim promenljivama koje se koriste u njoj moraju biti dodeljene vrednosti. Postoji nekoliko načina za to.

- <u>promenljiva je definisana i dodeljena joj je vrednost u okviru same skript datoteke.</u>

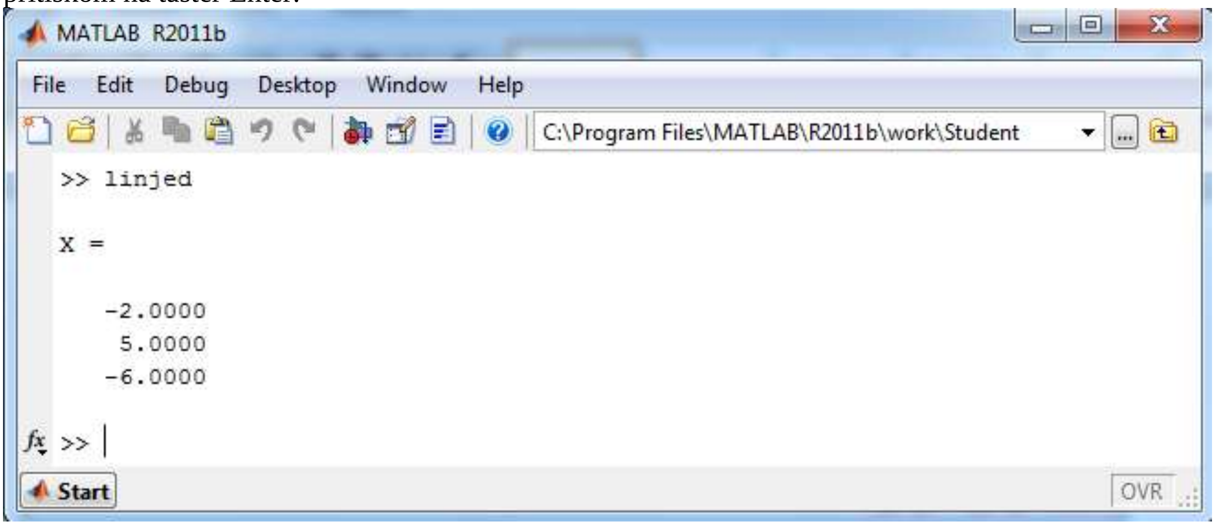
<u>**Primer 4.1:**</u> Napraviti program u obliku skript datoteke za rešavanje sistema linearnih algebarskih jednačina u matričnom obliku: **[A][X]=[B].** Neka ime programa bude `linjed`. Primeniti program za rešavanje sistema jednačina:

$$3x + 2y - y = 10$$
$$-x + 3y + 2z = 5$$
$$x - y - z = -1$$

Program upisan u prozor Editor i snimljen pod imenom `linjed.m u` folderu Student u direktorijumu C:\Program Files\MATLAB\R2011b\work\Student



Izvršavanje programa se ostvaruje upisivanjem imena programa u komandni prozor MATLAB-a i pritiskom na taster Enter:



- <u>Promenljive se definišu i dodeljuju im se vrednosti u okviru komandnog prozora.</u>

Promenljivama se mogu dodeljivati vrednosti u komandnom prozoru, pre izvršavanja programa. Svakoj promenljivoj se mogu dodeliti druge vrenosti pre izvršenja programa. Treba voditi računa da svaku promenljivu (pod istim imenom) prepoznaju sve skript datoteke.

Primer 4.2: Napisati program u formi skript datoteke za izračunavanje prosečne ocene studenata. Program snimiti pod imenom POS.

Program upisan u prozor Editor i snimljen pod imenom `POS.m` u folderu Student u direktorijumu C:\Program Files\MATLAB\R2011b\work\Student

```
%PROGRAM ZA IZRACUNAVANJE PROSECNE OCENE STUDENATA
%---------------------------------------------------
%Ulazne promenljive: OS - vektor ciji su elementi ocene studenata
%Izlazna promenljiva: PO - prosecna ocena studenata
%---------------------------------------------------
PO=sum(OS)/length(OS)
return
```

Pre izvršavanja programa je neophodno definisati vektor OS čiji su elementi ocene studenata za koje treba izračunati prosečnu ocenu. Nakon toga se upisuje ime programa POS i prritisne Enter:

```
>> OS=[8,9,6,7,6,10];
>> POS

PO =

       7.6667

>>
```

- <u>Promenljive se definišu u skript datoteci, ali im se vrednosti dodeljuju u komandnom prozoru, nakon pokretanja skript datoteke (program u toku izvršavanja zahteva od korisnika upisivanje vrednosti promenljivih). U skript datoteci se promenljive definišu na sledeći način</u>.

```
ime_promenljive=input('tekst poruke koja će biti ispisana u komandnom prozoru')
```

Primer 4.3: Napisati program u formi skript datoteke za izračunavanje prosečne ocene studenata. Ulaznu promenljvu (vektor sa ocenama studenata) definisati u skript datoteci a vrednsoti mu dodeljivati u komandnom prozoru u toku izvršavanja programa. (Iskoristiti prethodni primer).

Program upisan u prozor Editor i snimljen pod imenom `POS.m` u folderu Student u direktorijumu C:\Program Files\MATLAB\R2011b\work\Student (izvršena modifikacija porgrama POS iz Primera 4.2)

```
%PROGRAM ZA IZRACUNAVANJE PROSECNE OCENE STUDENATA
%---------------------------------------------------
%Ulazne promenljive: OS - vektor ciji su elementi ocene studenata
%Izlazna promenljiva: PO - prosecna ocena studenata
%---------------------------------------------------
%Definisanje ulayne promenljive:
OS=input('Unesite vektor ciji su elementi ocene studenata:');
%Izracunavanje prosecne ocene studenata:
PO=sum(OS)/length(OS)
return
```

Nakon pokretanja programa POS, zahteva se upisivanje vektora koji sadrži ocene studenata, nakon čega se izračunava prosečna ocena:

```
>> POS
Unesite vektor ciji su elementi ocene studenata:[8,9,6,7,6,10]

PO =
```

```
       7.6667

>>
```

Komande za kreiranje prikaza rezultata (izlaznih promenljivih)

Nakon izvršavanja komandi MATLAB automatski generiše prikaz njihovih rezultata. Standardni prikaz rezultata u MATLAB-u je je dat u svim prethodnim primerima i podrazumeva ispisivanje imena promenljive i njene vrednosti. Rezultat se ne prikazuje ukoliko na kraju reda koji se izvršava stoji znak tačka zarez. Međutim, u cilju bolje preglednosti i jasnije interpretacije rezultata u MATLAB-u se za generisanje prikaza rezultata često koriste komande `disp` i `fprintf`.

Komanda `disp` se koristi za prikazivanje elemenata promenljive bez prikazivanja njenog imena, kao i za prikazivanje tekstualnih poruka na ekranu. Opšti oblik komande `disp` je:

`disp(ime promenljive)` ili `disp('tekst kao znakovni niz')`

Primer 4.4: Korišćenje komande `disp` u programu POS, iz Primera 4.3, za prikazivanje prosečne ocene.

```
%PROGRAM ZA IZRACUNAVANJE PROSECNE OCENE STUDENATA
%-------------------------------------------------
%Ulazne promenljive: OS - vektor ciji su elementi ocene studenata
%Izlazna promenljiva: PO - prosecna ocena studenata
%-------------------------------------------------
%Definisanje ulazne promenljive:
OS=input('Unesite vektor ciji su elementi ocene studenata:');
%Izracunavanje prosecne ocene studenata:
PO=sum(OS)/length(OS);
disp('Prosecna ocena studenata je:')
disp(PO)
return
```

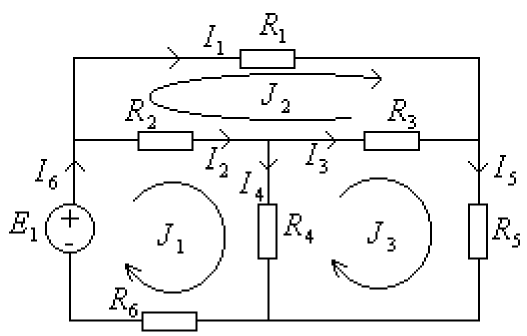
Kada se program izvrši u komandnom prozoru, dobija se:

```
>> POS
Unesite vektor ciji su elementi ocene studenata:[8,9,6,7,6,10]
Prosecna ocena studenata je:
       7.6667

>>
```

Često je potrebno dobijene rezultate prikazati u tabelarnom obliku. U tom slučaju se najpre definiše promenljiva u obliku matrice, čije kolone čine izračunate promenljive koje su vektori. Nakon toga se njeno prikazivanje na ekranu podesi preko komande `disp`.

Primer 4.5: Napraviti program u obliku skript datoteke za rešavanje prostog električnog kola na prikazanog na slici.

Ulazne promenjive: R_1, R_2, R_3, R_4, R_5 i V definisati u skript datoteci a vrednosti im dodeljivati u komandnom prozoru. Rezultat programa treba da bude u obliku tabele



(matrice) sa otporima u prvoj i strujama u drugoj koloni. Program snimiti pod imenom `elkolo`.
Uputstvo: Napisati jednačine po II K.Z. za sve tri konture. Dobijeni sistem od tri jednačine sa tri nepoznate struje rešiti kao matričnu jednačinu.

$$(R_2+R_4+R_6)J_1-R_2J_2-R_4J_3=E_1$$
$$-R_2J_1+(R_1+R_2+R_3)J_2-R_3J_3=0$$
$$-R_4J_1-R_3J_2+(R_3+R_4+R_5)J_3=0$$

Ova jednacina se rešava kao matrična: **[J]=[R]$^{-1}$[E],** gde su:

$$[\mathrm{R}]=\begin{bmatrix} R_2+R_4+R_6 & -R_2 & -R_4 \\ -R_2 & R_1+R_2+R_3 & -R_3 \\ -R_4 & -R_3 & R_3+R_4+R_5 \end{bmatrix},\quad [\mathrm{J}]=\begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix},\quad [\mathrm{E}]=\begin{bmatrix} E_1 \\ 0 \\ 0 \end{bmatrix}$$

Nepoznate stuje po granama kola se dobijaju primenom I KZ:

$$I_1=J_2;\; I_2=J_1-J_2;\; I_3=J_3-J_2;\; I_4=J_1-J_3;\; I_5=J_3;\; I_6=J_1;$$

Primeniti program za sledeće vrednosti ulaznih promenljivih:
$R_1=20\Omega,\, R_2=14\Omega,\, R_3=12\Omega,\; R_4=18\Omega,\; R_5=8\Omega,\, R_6=15\Omega,\, E=100V.$

```
% PROGRAM elkolo ZA RESAVANJE PROSTOG EL. KOLA METODOM KONTURNIH STRUJA
%-----------------------------------------------------------------------
% Definisanje ulaznih promenljivih:
R1=input('Unesi vrednost otpora R1=');
R2=input('Unesi vrednost otpora R2=');
R3=input('Unesi vrednost otpora R3=');
R4=input('Unesi vrednost otpora R4=');
R5=input('Unesi vrednost otpora R5=');
R6=input('Unesi vrednost otpora R6=');
E1=input('Unesi vrednost ems E1=');
E2=input('Unesi vrednost ems E2=');
E3=input('Unesi vrednost ems E3=');

% Definisanje matice koeficijenata [R] i vektora ems kontura [E]
R=[R2+R4+R6,-R3,-R4;-R2,R1+R2+R3,-R3;-R4,-R3,R3+R4+R5];
E=[E1;E2;E3];

%Izracunavanje konturnih struja:
J=inv(R)*E;

%Izracunavanje struja po granama kola:
I=[J(1);J(1)-J(2);J(3)-J(2);J(1)-J(3);J(3);J(1)];

%Definisanje tabele za prikaz rezultata:
Tabela=[[R1;R2;R3;R4;R5;R6],I];

%Koriscenje komande disp za kreiranje prikaza rezultata u trazenoj formi:
disp(' ')
disp('REZULTAT:')
disp(' ')
disp('    Otpor     Struja')
disp('    (Omi)    (Amperi)')
disp(' ')
disp(Tabela)
return
```

Kada se program `elkolo` izvrši, u komandnom prozoru se dobija:

```
>> elkolo
Unesi vrednost otpora R1=20
Unesi vrednost otpora R2=14
Unesi vrednost otpora R3=12
Unesi vrednost otpora R4=18
Unesi vrednost otpora R5=8
Unesi vrednost otpora R6=15
Unesi vrednost ems E1=100
Unesi vrednost ems E2=0
Unesi vrednost ems E3=0

REZULTAT:

    Otpor      Struja
    (Omi)     (Amperi)
   20.0000    3.3083
   14.0000    1.7655
   12.0000    0.5115
   18.0000    1.2540
    8.0000    2.0543
   15.0000    3.3083
```

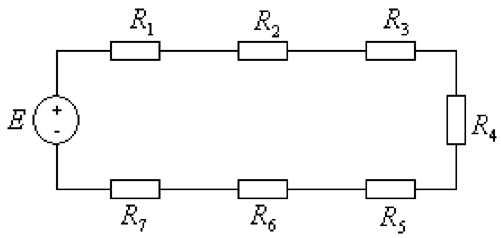
Komanda `fprintf` se koristi za prikaz rezultata na ekranu ili upisivanje u određenu datoteku. Komandom `fprintf` se rezultat može formatirati, odnosno tekst i numeričke vrednosti mogu biti izmešani i prikazani u istom redu. Pored toga, može se zadati i format prikaza brojeva. Komanda `fprintf` omogućava prelazak u novi red na proizvoljnom mestu znakovnog niza, tako što se pre znaka kojim treba da počne prikaz u novom redu umetne sekvenca `\n`.
Komanda `fprintf` ima mnogo opcija i može biti dosta složena. Jedan od oblika ove komande, koji se koristi za prikazivanje izmešanog teksta i numeričkih podataka u istom redu, je:

```
fprintf('Tekst koji objasnjava rezultat %f.',promenljiva)
```

Znak `%` u komandi `fprintf` označava mesto gde broj treba uneti u tekst, `f` je format prikaza rezultata, a `promenljiva` je promenljiva čija se vrednost ispisuje.
Umesto `f` se za format prikaza može koristiti i oznaka `g` koja daje kraći zapis (eliminišu se nule iza decimalnog zareza), koji je često pregledniji i samim tim praktičniji.

Primer 4.6: Napisati program u obliku skript datoteke za izračunavnje napona i snage na otpornicima u redno vezanom kolu, sa proizvoljnim brojem otpornika i jednim naponskim izvorom. Predvideti definisanje ulaznih promenljivih u skript datoteci i dodeljivanje vrednosti tako definisanih ulaznih promenljivih u komandnom prozoru. Rezultat programa treba da bude u obliku tabele (matrice) sa otporima u prvoj koloni, naponima u drugoj i snagama u trećoj koloni (koristiti komandu `disp`). Nakon tabele program treba da prikaže vrednost struje u kolu i ukupnu snagu (koristiti komandu `fprintf`). Program snimiti pod imenom `rednokolo`. (Podsetnik: $U_n = R_n \dfrac{E}{R_{ekv}}$,

$P_n = R_n \dfrac{E^2}{R_{ekv}^2}$ napon i snaga na otpornicima u rednom kolu)



Primeniti program za kolo na slici, koje ima sledeće vrednosti otpora i napona izvora:
$R_1 = 20\Omega, R_2 = 14\Omega, R_3 = 12\Omega, R_4 = 18\Omega, R_5 = 8\Omega, R_6 = 15\Omega, R_7 = 10\Omega, E = 24V$.

```
%PROGRAM rednokolo ZA IZRACUN. NAPONA I SNAGE NA OTPORNICIMA U REDNOM KOLU
%-----------------------------------------------------------------------------
%Definisanje ulaznih promenljivih:
E=input('Unesite vrednost napona izvora E=');
Rn=input('Unesite vrednosti otpornika kao elemente u vektoru vrsti\nRn=');
%-----------------------------------------------------------------------------
Rekv=sum(Rn); %izracunavanje ekvivalentnog otpora kola
Un=Rn*E/Rekv; %izracunavanje napona na otpornicima
Pn=Rn*E^2/Rekv^2;  %izracunavanje snage na otpornicima
I=E/Rekv; %izracunavanje struje u kolu
Ptotal=E*I; %izracunavanje ukupne snage kola
Tabela=[Rn' Un' Pn']; %generisanje tabele (matrice) sa kolonama Rn,Un i Pn
disp(' ') %ostavlja se jedno prazno mesto radi preglednosti
disp('REZULTATI:')
disp(' ')
disp('   Otpor      Napon      Snaga') %zaglavlje za prikazivanje rezult.
disp('   (omi)       (V)        (W)')
disp('')
disp(Tabela)
disp('')
fprintf('Struja kola je %f A.\n',I) %prikaz rezultata preko komande fprintf
fprintf('Ukupna snaga u kolu je %f W.\n',Ptotal)
return
```

Kada se program `rednokolo` izvrši, u komandnom prozoru se dobija:

```
>> rednokolo
Unesite vrednost napona izvora E=24
Unesite vrednosti otpornika kao elemente u vektoru vrsti
Rn=[20,14,12,18,8,15,10]

REZULTATI:

   Otpor      Napon      Snaga
   (omi)       (V)        (W)
   20.0000    4.9485    1.2244
   14.0000    3.4639    0.8571
   12.0000    2.9691    0.7346
   18.0000    4.4536    1.1019
    8.0000    1.9794    0.4897
   15.0000    3.7113    0.9183
   10.0000    2.4742    0.6122

Struja kola je 0.247423 A.
Ukupna snaga u kolu je 5.938144 W.
```

Često je potrebno rezultat ili poruku u programu kreirati kao tekst sa više umetnutih brojeva (više promenljivih). U tom slučaju se koristi sledeći oblik komande `fprintf`

```
fprintf('...tekst...%f...%f... 'promenljiv1,promenljiva2)
```

Primer 4.7: Napisati program u obliku skript datoteke za izračunavanje aktivne snage, reaktivne snage i faktora snage potrošača komleksne impedanse Z priključenog na napon U. Predvideti definisanje i dodeljivanje vrednosti ulaznih promenljivih u komandnom prozoru. Primeniti program za Z =(20+j10) Ω i U=24V

```
%PROGRAM impedansa ZA ODREDJIVANJE SNAGE I FAKTORA SNAGE NA KOMPLEKSNOJ
%IMPEDANSI
%--------------------------------------------------------------------------
S=abs(U)^2/conj(Z); %Komplkensa prividna snaga potrosaca
P=real(S); %Aktivna snaga potrosaca
Q=imag(S); %Reaktivna snaga potrosaca
cosfi=P/abs(S); %Faktor snage potrosaca
%Kreiranje izgleda rezultata pomoću komande fprintf:
fprintf('Aktivna snaga potrosaca je %f W, reaktivna snaga %f VAr, i faktor
snage %f\n',P,Q,cosfi)
```

Najpre se u komandnom prozoru definišu ulazne promenljive Z i U i dodele i se vrednosti, a tek onda aktivira program `impedansa`:

```
>> Z=20+1i*10;
>> U=24;
>> impedansa
Aktivna snaga potrosaca je 23.040000 W, reaktivna snaga 11.520000 VAr, i
faktor snage 0.894427
```

Primer 4.8: Relativni odnosi struje jednofaznog i dvofaznog zemljospoja prema struji trofaznog kratkog spoja na istoj lokaciji u EES, u funkciji odnosa ekvivalentnih reaktansi nultog i direktnog redosleda $\left(k = \dfrac{X_0}{X_d}\right)$ su dati preko relacija:

$$\frac{I_{k1E}}{I_{k3}} = \frac{3}{2+k}; \qquad \frac{I_{k2E}}{I_{k3}} = \frac{\sqrt{3}}{1+2k}\sqrt{k^2+k+1}$$

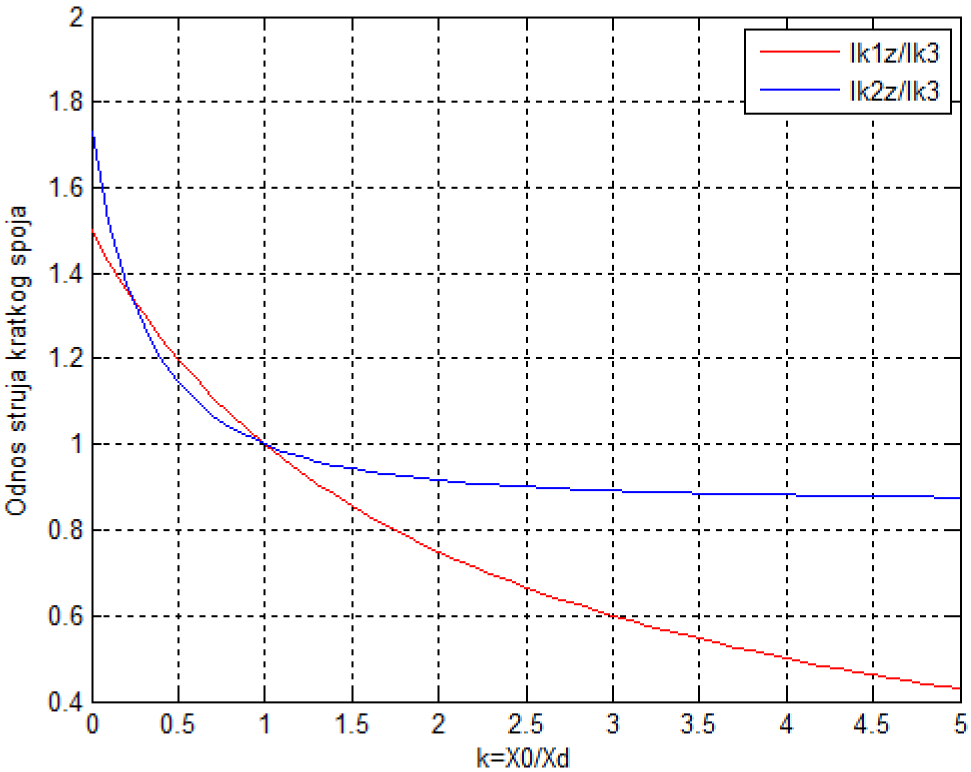
Napraviti program u obliku skript datoteke koji izračunava vrednosti ovih odnosa i crta njihove grafike u na istom grafikonu u funkciji parametra *k*. Program snimiti pod imenom `odnosstruja`
Primeniti program za vrednost parametra *k* od 0 do 5 sa korakom 0,1.

```
%PROGRAM odnosstruja ZA ODREDJIVANJE I GRAFICKO PRIKAZIVANJE ODNOSA STRUJA
%KRATKIH SPOJEVA U FUNKCIJI PARAMETRA k=X0/Xd U EES-u
%--------------------------------------------------------------------------
%Definisanje ulaznog parametra k
k=input('Unesi vrednosti parametra k u obliku vektora k=X0/Xd=');
%Izracunavanje odnosa struja kratkog spoja:
Ik1z_Ik3=3./(2+k);
Ik2z_Ik3=sqrt(3)./(1+2*k).*sqrt(k.^2+k+1);
%Podesavanje prikaza rezultata
disp('REZULTATI')
Rezultat=[k' Ik1z_Ik3' Ik2z_Ik3'];
disp('       k     Ik1z_Ik3  Ik2z_Ik3');
disp('    -----------------------------')
disp(Rezultat);
%generisanje grafika:
plot(k,Ik1z_Ik3,'r',k,Ik2z_Ik3,'b');
xlabel('k=X0/Xd'); ylabel('Odnos struja kratkog spoja');
legend('Ik1z/Ik3','Ik2z/Ik3'); grid on;
return
```

Primena programa za konkretan slučaj daje sledeći rezultat:

```
>> odnosstruja
Unesi vrednosti parametra u obliku vektora k=X0/Xd=0:0.1:5
REZULTATI
       k     Ik1z_Ik3  Ik2z_Ik3
```

```
    ----------------------------
         0    1.5000    1.7321
    0.1000    1.4286    1.5207
    0.2000    1.3636    1.3777
    0.3000    1.3043    1.2763
    0.4000    1.2500    1.2019
    0.5000    1.2000    1.1456
    0.6000    1.1538    1.1022
    0.7000    1.1111    1.0680
    ...
```



Uvoženje i izvoženje podataka

Često se ulazni podaci koji se pridružuju promenljivama u skript datoteci ili u komandnom prozoru mogu nalazi u drugim skript datotekama, ili zapisani u drugim računarskim programima kao što je Excel. Tada je potrebno te podatke uvesti u skript datoteku (program) ili komandni prozor MATLAB-a. Isto tako, može se javiti potreba da se podaci iz MATLAB-a prenesu (izvezu) u druge programe. U zavisnosti od forme zapisa podataka, primenjuju se odgovarajuće komande za njihov uvoz.

-Najjednostavniji način je kada su ulazni podaci zapisani u drugoj skript datoteci, u obliku skalara, vektora ili matrice. Tada se oni uvoze jednostavnim upisivanjem imena datoteke u koju su smešteni. Sve promenjive koje su definisane u toj datoteci se automatski prihvataju u skript datoteci, odnosno komandnom prozoru. Te promenljive se mogu menjati i dodeljivati im se druge vrednosti kao i ostalim promenljivama u programu u koji su uvežene.

Primer 4.9: U toku dana su na jednoj lokaciji na svaki sat merene vrednosti brzine vetra, temperature i solarne iradijacije. Izmerene vrednosti su date u sledećoj tabeli.

t (h)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
v(m/s)	10.8	10.7	11.3	10.9	11.5	11.6	10.7	10.9	10.1	10.4	10.7	11.6	11.6	11.9	11.5	11.3	11.5	11.7	12.2	11.7	11.5	11.3	11.3	11.3
Ta(°C)	15	15	14	13	14.5	15	15.2	17	17	17.5	18	19	20	21	21	21	21	21	20	19	18	17	16	15.5
Is (W/m^2)	0	0	0	0	90	220	250	450	650	680	820	850	830	850	680	600	250	200	150	70	50	0	0	0

a) Odrediti maksimalne i minimalne vrednosti izmerenih veličina kao i vreme (čas) u kome su izemerene te vrednosti. Odrediti srednje vrednosti ovih veličina u posmatranom periodu.
b) Grafički prikazati izmerene veličine na tri grafikona u okviru jednog grafičkog prozora (podeliti grafički prozor na tri dela). Brzinu vetra prikazati kao kontinualni grafik, temperaturu ambijenta kao trakasti vertikalni grafik i solarnu iradijaciju kao grafik diskretnih vrednosti. Označiti ose odgovarajućim veličinama.

Ulazne podatke definisati u posebnoj skript datoteci u obliku vektora koji sadrže odgovarajuće vrednosti izmerenih veličina. Ulazne podatke uvesti u glavni program koji rešava postavljeni zadatak. Ulaznu datoteku snimiti pod imenom `merenja`, a glavni program pod imenom `primer49`

Ulazna skript datoteka `merenja` u kojoj su zapisane ulazne promenljive (izmerene vrednosti veličina iz tabele):

```
%ULAZNA DATOTEKA merenja ZA PROGRAM primer49
%t - vreme
%v - brzina vetra u m/s
%Ta - temperatura ambijenta u stepenima C
%Is - solarna iradijacija u W/m^2
%-----------------------------------------------------------------------
t=[1:24];
v=[10.8 10.7 11.3 10.9 11.5 11.6 10.7 10.9 10.1 10.4 10.7 11.6 11.6 …
11.9 11.5 11.3 11.5 11.7 12.2 11.7 11.5 11.3 11.3 11.3];
Ta=[15 15 14 13 14.5 15 15.2 17 17 17.5 18 19 20 21 21 21 21 21 20 19 …
18 17 16 15.5];
Is=[0 0 0 0 90 220 250 450 650 680 820 850 830 850 680 600 250 200 150 70 50 0 0 0];
```

Glavni program `primer49`:

```
%PROGRAM primer49 ZA RESAVANJE PRIMERA 4.9
%-----------------------------------------------------------------
%Uvoz ulaznih podataka (merenih velicina iz tabele)
merenja; %izvrsava se skript datoteka merenja koja sadrzi t, v, Ta i Is
%Maksimalnih, minimalnih i srednjih vrednsoti izmerenih velicina:
[vmax,ivmax]=max(v);[vmin,ivmin]=min(v);vsr=mean(v);
[Tamax,iTamax]=max(Ta);[Tamin,iTamin]=min(Ta);Tasr=mean(Ta);
[Ismax,iIsmax]=max(Is);[Ismin,iIsmin]=min(Is);Issr=mean(Is);
%Podesavanje formata prikaza rezultata:
fprintf('Maksimalna brzina vetra je %g m/s. Izmerena je u %g h.\n',vmax,ivmax);
fprintf('Minimalna brzina vetra je %g m/s. Izmerena je u %f h.\n',vmin,ivmin);
fprintf('Srednja brzina vetra u datom periodu je %g m/s.\n',vsr);
fprintf('Maksimalna temperatura je %g stepeni C. Izmerena je u %g h. \n',Tamax,iTamax);
fprintf('Minimalna temperatura je %g stepeni C. Izmerena je u %g h. \n',Tamin,iTamin);
fprintf('Srednja temperatura u posmatranom periodu je %f stepeni C.\n',Tasr);
fprintf('Maksimalna iradijacija je %g W/m^2. Izmerena je u %g h. \n',Ismax,iIsmax);
fprintf('Minimalna iradijacija je %g W/m^2. Izmerena je u %g h. \n',Ismin,iIsmin);
fprintf('Srednja vrednost iradijacije u toku dana je %g W/m^2.\n',Issr);
%Kreiranje grafikona:
subplot(3,1,1),plot(t,v);xlabel('t (h)');ylabel('v (m/s)');title('Brzina vetra');grid on;
subplot(3,1,2),bar(t,Ta);xlabel('t (h)');ylabel('Ta (Stepeni C)');
title('Temperatura ambijenta');grid on;
subplot(3,1,3),stem(t,Is);xlabel('t (h)');ylabel('Is (W/m^2)');
title('Solarna iradijacija');grid on;
```

-Podaci mogu, umestu u formi vektora ili matrica, biti zapisani u ASCII formatu. Tada se za njihovo uvoženje u druge skript datoteke ili komandni prozor koristi komanda `load`. Nakon komande `load` sledi ime datoteke u kojoj su smešteni podaci koje treba uvesti.

Primer 4.10: Primer 4.9 uraditi za slučaj kada su ulazni podaci u ulaznoj datoteci `merenja` snimljeni u ACSII formatu.

Ulazna datoteka `merenja` u slučaju zapisa podataka u ACSII formatu:

```
%ULAZNA DATOTEKA ZA PROGRAM primer49
%t - vreme
%v - brzina vetra u m/s
%Ta - temperatura ambijenta u stepenima C
%Is - solarna iradijacija u W/m^2
%-----------------------------------------------------------------------------
         %  t      |     v      |       Ta      |       Is      |
            1          10.8          15             0
            2          10.7          15             0
            3          11.3          14             0
            4          10.9          13             0
            5          11.5        14.5            90
            6          11.6          15           220
            7          10.7        15.2           250
            8          10.9          17           450
            9          10.1          17           650
           10          10.4        17.5           680
           11          10.7          18           820
           12          11.6          19           850
           13          11.6          20           830
           14          11.9          21           850
           15          11.5          21           680
           16          11.3          21           600
           17          11.5          21           250
           18          11.7          21           200
           19          12.2          20           150
           20          11.7          19            70
           21          11.5          18            50
           22          11.3          17             0
           23          11.3          16             0
           24          11.3        15.5             0
```

Glavni program `primer49` kada su ulazni podaci u ulaznoj datotecei merenja zapisani u ACSII formatu:

```
%PROGRAM primer49 ZA RESAVANJE PRIMERA 4.9
%--------------------------------------------------------------------------
%Uvoz ulaznih podataka (merenih velicina iz tabele)
load merenja.m; %ucitavanje datoteke sa podacima u ACSII formatu

%Adresiranjem datoteke merenja se definisu promenljive t, v, Ta i Is
t=merenja(:,1);v=merenja(:,2);Ta=merenja(:,3);Is=merenja(:,4);
  .
  .   (ostatak programa je identičan kao u Primeru 4.9)
  .
```

-MATLAB može uvoziti podatke iz drugih programa, kao što je Excel. Podaci se iz Excela uvoze komandom `xlsread`, koja podatke učitane iz proračunske tabele dodeljuje promeljivoj kao niz. Pri tome ova Excel datoteka mora takođe biti u tekućem direktorijumu MATLABA. Najjednostavniji oblik ove komande je:

```
Matlab_Promenljiva=xlsread('Ime_Excel_Datoteke')
```

ili

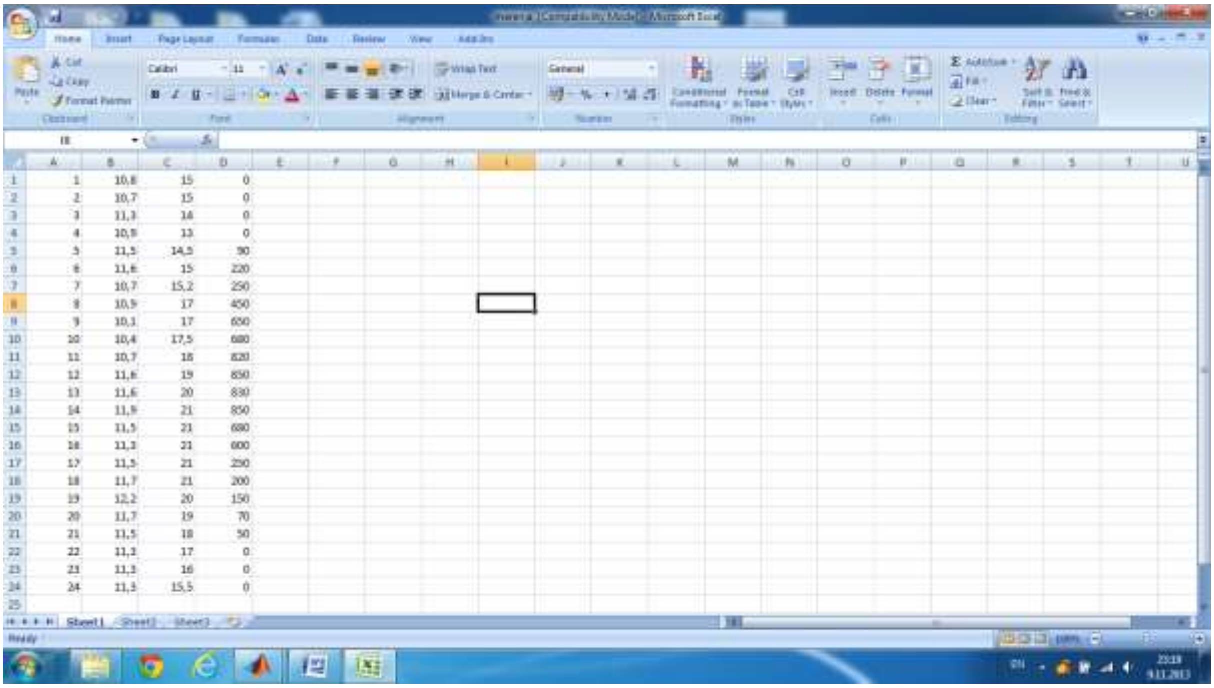
```
Matlab_Promenljiva=xlsread('Ime_Excel_Datoteke', 'Ime_Lista' )
```

Podaci se iz MATLAB-a mogu izvoziti u Excel pomoću komande `xlswrite`:

```
Matlab_Promenljiva=xlswrite('Ime_Excel_Datoteke', 'Ime_Promenljive' )
```

Primer 4.11: Primer 4.9 uraditi za slučaj kada su ulazni podaci u ulaznoj datoteci `merenja` snimljeni u Excel tabeli.

Ulazna datoteka `merenja` u formi Excel tabele:



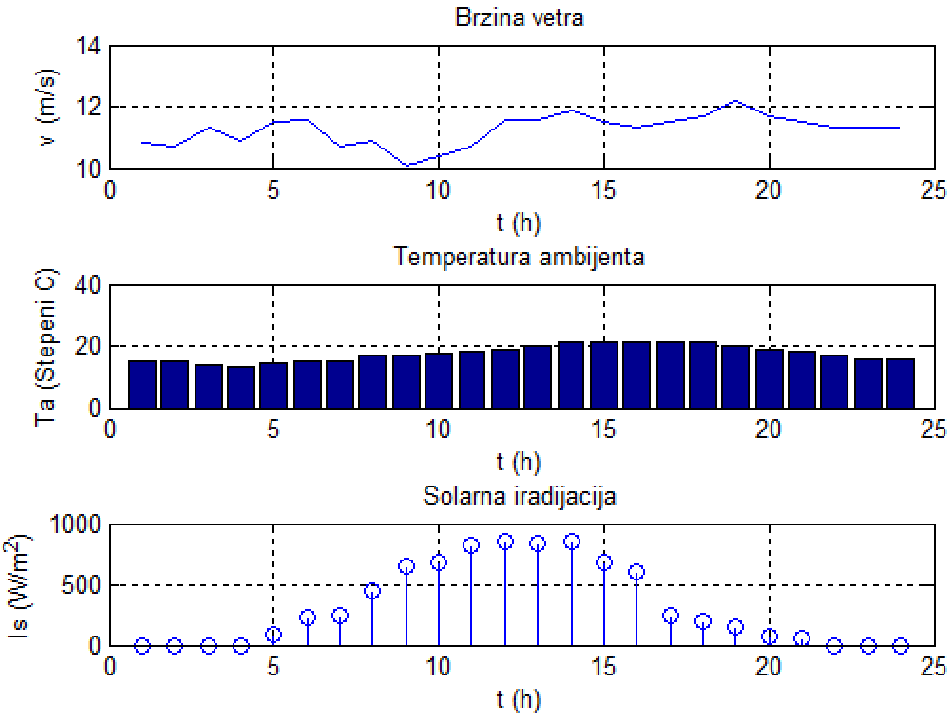
	A	B	C	D
1	1	10,8	15	0
2	2	10,7	15	0
3	3	11,3	14	0
4	4	10,8	13	0
5	5	11,5	14,5	90
6	6	11,6	15	220
7	7	10,7	15,2	250
8	8	10,9	17	450
9	9	10,1	17	650
10	10	10,4	17,5	680
11	11	10,7	18	820
12	12	11,6	19	850
13	13	11,6	20	830
14	14	11,9	21	850
15	15	11,5	21	680
16	16	11,3	21	600
17	17	11,5	21	250
18	18	11,7	21	200
19	19	12,2	20	150
20	20	11,7	19	70
21	21	11,5	18	50
22	22	11,2	17	0
23	23	11,3	16	0
24	24	11,3	15,5	0

Glavni program `primer49` kada su ulazni podaci u ulaznoj datotecei `merenja` zapisani u Excel tabeli:

```
%PROGRAM primer49 ZA RESAVANJE PRIMERA 4.9
%----------------------------------------------------------------------
%Uvoz ulaznih podataka iz Excel tabele merenja:
merenja=xlsread('merenja');
%Adresiranjem datoteke merenja se definisu promenljive t, v, Ta i Is
t=merenja(:,1);v=merenja(:,2);Ta=merenja(:,3);Is=merenja(:,4);

  .
  .   (ostatak programa je identičan kao u Primeru 4.9)
  .
```

U sva tri slučaja (Primeri 4.9-4.11) glavni program `primer49` se izvršava na isti način. Rezultati koji se dobijaju u komandnom prozoru su:

```
>> primer49
Maksimalna brzina vetra je 12.2 m/s. Izmerena je u 19 h.
Minimalna brzina vetra je 10.1 m/s. Izmerena je u 9.000000 h.
Srednja brzina vetra u datom periodu je 11.25 m/s.
Maksimalna temperatura je 21 stepeni C. Izmerena je u 14 h.
Minimalna temperatura je 13 stepeni C. Izmerena je u 4 h.
Srednja temperatura u posmatranom periodu je 17.529167 stepeni C.
Maksimalna iradijacija je 850 W/m^2. Izmerena je u 12 h.
Minimalna iradijacija je 0 W/m^2. Izmerena je u 1 h.
Srednja vrednost iradijacije u toku dana je 320.417 W/m^2.
```



4.2 *Funkcijske datoteke (korisničke funkcije - programi)*

Funkcijske datoteke omogućavaju stvaranje novih funkcija. Mnogo funkcija je već programirano u MATLAB-u. To su njegove matične (ugrađene) funkcije, kao što su `sin(x)`, `sqrt(x)`, `abs(x)` itd. Funkcijska datoteka se još naziva i korisnička funkcija ili funkcijski program. To je MATLAB-ov program koji je korisnik napisao i snimio kao funkcijsku datoteku. Ova funkcija se može upotrebljavati kao bilo koja ugrađena funkcija. Funckijske datoteke se mogu upotrebljavati kao potprogrami unutar većih programa. Na taj način se veliki računarski programi prave od manjih delova koji se mogu nezavisno testirati.

Kreiranje, snimanje i izvršavanje funkcijske datoteke

Funkcijske datoteke se pišu i uređuju u prozoru Editor, kao i script datoteke. Za razliku od skript datoteka, pri kreiranju funkcijskih datoteka prva linija mora počinjati sentencom `function`. U opštem slučaju to je izraz sledećeg oblika:

```
function [y1,y2,…]=ime_funkcije(x1,x2,…)
```

Posle ovog reda, koji se naziva red s definicijom funkcije, sledi program (telo funkcije) - niz MATLAB komandi i izraza.

`y1,y2,…` - Imena izlaznih promenjlivih koji se dobijaju kada se funkcija pozove. Kada postoji samo jedna izlazna promenljiva, tada se ona može pisati bez uglastih zagrada.
`ime_funkcije` - Ime funkcije, odnosno funkcijske datoteke,
`x1,x2,…` - Ulazne promenljive (argumenti) funkcije.

Promenljive funkcijske datoteke su lokalne promenljive. U funkcijsku datoteku mogu se iz komandnog prozora proslediti samo one promenljive koje su ulazni argumenti funkcije.
U funkcijskim datotekama važe i sve ulazne i izlazne karakteristike skript datoteka. To znači da će na ekranu biri prikazane sve promenljive kojima je, u nizu komandi i izraza koji sledi posle reda s definicijom funkcije, dodeljena vrednost, ukoliko na kraju njihovog reda ne stoji tačka zarez. Za

interaktivno unošenje podataka može se upotrebiti komanda `input`, a komande `disp`, `fprintf`, i `plot` za prikazivanje na ekranu, snimanje na disk, odnosno crtanje slika, baš kao i u skript datoteci.
Funkcijske datoteke se moraju snimiti pre upotrebe. Veoma je važno da se datoteka snimi pod imenom koje je identično imenu funkcije u redu s definicijom. Samo tako će funkcija moći da se poziva (upotrebljava) navođenjen tog imena i odgovarajućeg argumenta u zagradi.
Funkcijska datoteka (funkcija) koju je definisao korisnik upotrebljava se isto kao ugrađena MATLB-ova funkcija. Može se pozivati iz komandnog prozora, skript datoteka i drugih funkcija. Da bi se funkcija mogla koristiti, direktorijum u koji je snimljena mora biti tekući ili naveden u putanji pretraživanja, kao kod skript datoteka. Funkcijska datoteka se može korisiti i kao deo matematičkog izraza ili kao argument druge funkcije. U svim slučajevima korisnik mora tačno znati kakvi su ulazni, a kakvi izlazni argumenti.

Primer 4.12: Napisati program u obliku funkcijske datoteke koji izračunava ekvivalentni otpor n paralelno vezanih otpornika ($\frac{1}{R_{ek}}=\frac{1}{R_1}+\frac{1}{R_2}+...+\frac{1}{R_n}$). Ime i argument funkcije neka budu `REK = rek(R)`. Primeniti program slučaj kada su paralelno vezani otpornici: 50 Ω, 75 Ω, 300 Ω, 60 Ω, 500 Ω, 180 Ω, 36 Ω.

```
function REK=rek(R)
%Funkcija za izracunavanje ekvivalentnog otpora n paralelnih otpornika
%-------------------------------------------------------------------------
% Ulazni argument R je vektor koji sadrzi vrednosti paralelnih otpora
% Izlazni argument je REK - ekvivalentni otpor paralelne veze otpora
REK=(sum(1./R))^(-1);
```

Izvršavanje funkcije `rek` u komandnom prozoru na tri načina:

```
>> R=[50,75,300,60,500,180,36];          %Prvi nacin
>> REK=rek(R)

REK =

   11.2782

>> REK=rek([50,75,300,60,500,180,36])     %Drugi nacin

REK =

   11.2782
                                          %Treci nacin
>> fprintf('Ekvivalentni otpor je %f Oma. \n',rek([50,75,300,60,500,180,36]))
Ekvivalentni otpor je 11.278195 Oma.
>>
```

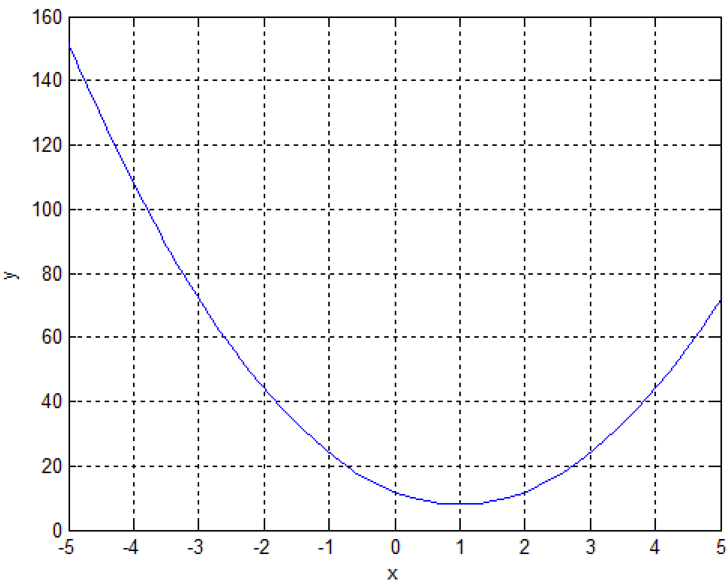
Primer 4.13: Napisati korisničku funkciju koja izračunava lokalni maksimum ili minimum kvadratne funkcije oblika $f(x)=ax^2+bx+c$. Ime i argumenti funkcije neka budu `[xm,ym]=maxmin(a,b,c)`. Ulazni argumenti su konstante `a,b` i `c,` a izlazni koordinate `xm` i `ym` maksimuma ili minimuma.
Primeniti program na funkciju: $f(x)=4x^2-8x+12$. Primenom komande `fplot` nacrtati ovu funkciju u opsegu $-5\le x\le 5$ kako bi ste utvrdili da li je primenom funkcije `maxmin` dobijen njen maksimum ili minimum.

Funkcija `minmax`:

```
function [xm,ym]=maxmin(a,b,c)
%Korisnicka funkcija za izracunavanje max ili min kvadratnih funkcija
%f(x)=a*x^2+b*x+c
%---------------------------------------------------------------------
%Uslov da funkcija ima max ili min je da joj je prvi izvod jednak nuli:
%fp(x)=2*a*x+b = 0 => xm=-b/(2*a); ym=f(xm)=a*xm^2+b*xm+c;
%--------------------------------------------------------------------
%Kod:
xm=-b/(2*a);
ym=a*xm^2+b*xm+c;
```

Rezultat primene korisničke funkcije `minmax` i grafik navedene kvadratne funkcije

```
>> [xm,ym]=maxmin(4,-8,12)

xm =

     1


ym =

     8

>> fplot('4*x^2-8*x+12',[-5,5]), xlabel('x'),ylabel('y'), grid on
```



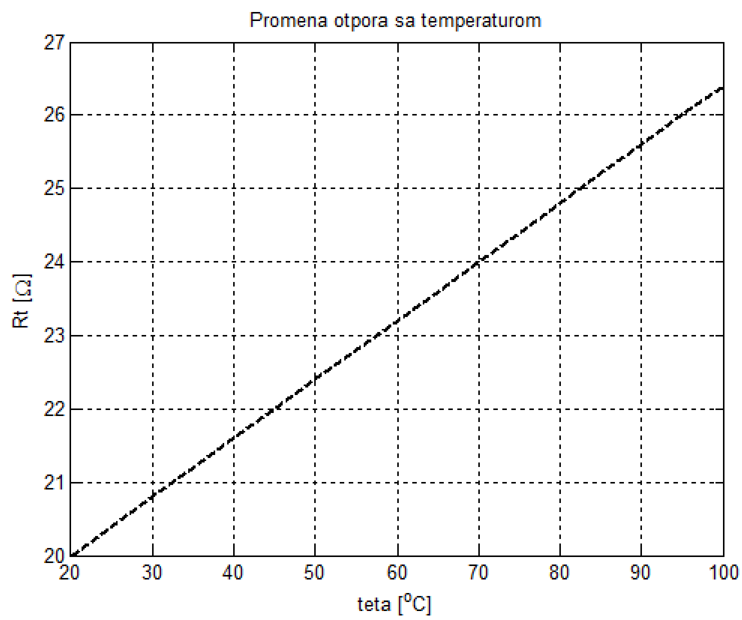
Primer 4.13: Napisati program u obliku funkcijske datoteke (korisničku funkciju) za izračunavanje vrednosti i crtanje grafika promene aktivnog otpora provodnika u funkciji temperature. Ime i argumeti funkcije neka budu `Rt=Rteta(R20,teta.`(Promena otpora bakarnih i aluminijumskih provodnika sa temperaturom izračunava se pomoću izraza: $R_\theta = R_{20^\circ} \cdot \left[1 + 0{,}004 \cdot \left(\theta - 20^\circ\right)\right]$, gde je R_{20° otpornost provodnika na 20 °C, a θ tekuća temperatura). Primeniti program za: R_{20° =5Ω u opsegu temperature θ od 20°C do 100°C, sa korakom od 5°C.

```
function Rt=Rteta(R20,teta)
% Program za izracunavanje vrednosti otpora u f-ji temperature provodnika
```

```
%---------------------------------------------------------------------------
%Ulazni argumenti su:
%R20 - otpor na 20 stepeni
%teta - temperatura u obliku skalara ili vektora
%Izlazlazni argument je: Rt - vrednost otpora na zadatoj tempereturi
%---------------------------------------------------------------------------
%Kod funkcije:
Rt=(1+0.004.*(teta-20)).*R20;
%Crtanje grafika promene otpora sa temperaturom
plot(teta,Rt,'--k','linewidth',2);
xlabel('teta [^oC]');ylabel('Rt [\Omega]');
title('Promena otpora sa temperaturom'); grid on;
%Kreiranje prikaza rezultata u obliku tabele pomocu komande disp:
disp(' ');
disp('REZULTAT:');
disp('     teta       Rt    ');
disp('   (Step C)    (Omi)  ');
disp('--------------------');
disp([teta' Rt']);
```

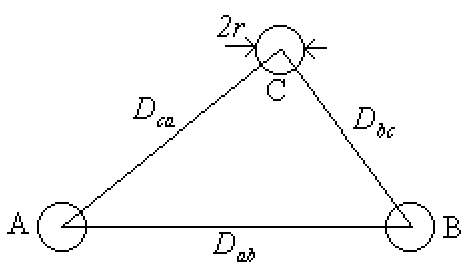
Rezultati izvršavanja programa u komandnom prozoru:

```
>>
Rt=Rteta(20,[20:5:100])

REZULTAT:
     teta       Rt
   (Step C)    (Omi)
--------------------
   20.0000    20.0000
   25.0000    20.4000
   30.0000    20.8000
   35.0000    21.2000
   40.0000    21.6000
   45.0000    22.0000
   50.0000    22.4000
   55.0000    22.8000
   60.0000    23.2000
   65.0000    23.6000
   70.0000    24.0000
   75.0000    24.4000
   80.0000    24.8000
   85.0000    25.2000
   90.0000    25.6000
   95.0000    26.0000
  100.0000    26.4000
```



Primer 4.14: Napisati program u obliku funkcijske datoteke za izračunavanje podužne induktivnosti vazdušnog voda. Ime i argumenti funkcijskog programa neka budu `lv=lvoda(Dab,Dbc,Dca,r)`.

$$l_V = 2 \cdot 10^{-4} \ln\left(\frac{\sqrt[3]{D_{ab} D_{bc} D_{ca}}}{0{,}778r}\right),$$



Ulazni podaci su rastojanja između pojednih provodnika u vodu (D_{ab}, D_{bc}, D_{ca}) i poluprečnik provodnika voda r. Primeniti program za konkretan primer: $D_{ab} = 1{,}4m; D_{bc} = 1{,}3m; D_{ca} = 1{,}5m;$ i r=4,8mm

```
function lv=lvoda(Dab,Dbc,Dca,r)
% Program za izracunavanje poduzne induktivnosti vazdusnog voda
```

```
%-------------------------------------------------------------
% Ulazni argumenti funkcije su:
% Dab,Dbc,Dca-rastojanja izmedju provodnika voda u [m] i
% r-poluprecnik provodnika voda u [m]
% Izlazni argument je lv - poduzna induktivnost voda u H/m
lv=2*10^(-4)*log((Dab*Dbc*Dca)^(1/3)/(0.778*r));
```

Primena funkcije `lvoda`:

```
>> lv=lvoda(1.4,1.3,1.5,4.8e-3)

lv =

    0.0012
```

4.3 Upravljanje tokom programa

Kod jednstavnih programa, komande se izvršavaju jedna za drugom redosledom kako su zapisane u komandnom prozoru, skript datoteci ili korisničkoj funkciji. Do sada su svi programi koji (u prethodnim poglavljima) bili takvi. Međutim, u mnogim lsučajevima potrebni su složeniji programi u kojima komande ne moraju obavezno da se izvršavaju redosledom kojim su upisane, ili je potrebno da se pojedine komande (ili grupe komandi) izvršavaju s različitim ulaznim promenljivih ili da se njihovo izvršavanje ponovi više puta unutar programa.
MATLAB ima više alatki koje omogućavaju upravljanje tokom izvršavanja programa. To su relacioni i logički operatori, uslovni iskazi i petlje.

Relacioni operatori : `<,  >,  <=,  >=,  = =,  ~ =`

Relacioni operatori se koriste kao aritmetički operatori unutar matematičkih izraza za poređenje dva broja ili izraza i utvrđuje da i je iskaz tačan ili netačan. Kada se porede dva broja ili izraza, rezultat je 1 (logička istina) ukoliko je izraz poređenja istinit; u suprotnom je 0 (logička nula). Rezultat se može upotrebiti u drugom matematičkom izrazu, za adresiranje vektora, ili u drugim komandama za upravljanje tokom programa.

Primer 4.15: Upotreba relacionih operatora.

```
>> 4<6

ans =

     1

>> 4>6

ans =

     0

>> 2==5

ans =

     0
```

```
>> 2~=5

ans =

     1

>> a=[1 0 1 0 1 0];
>> b=[1 1 1 1 1 1];
>> a==b

ans =

     1     0     1     0     1     0
```

Logički operatori: `&` AND, `|` OR, `~` NOT

Operandi logičkih operatora su brojevi. Svaki broj različit od nule smatra se logičkom vrednsošću `true`, dok se nula smatra logičkom vrednsošću `false`. Kao i relacioni operatori koriste se u drugim matematičkim operacijama, za adresiranje nizova i u drugim komandama za upravljanje tokom programa (npr. kod komande `if`, `while`...). Kao i relacioni operatori, logički operatori se mogu primenjivati i na skalarne i na vektore (matrice).

Primer 4.16: Upotreba logičkih operatora.

```
>> 3&8

ans =
     1

>> a=5|0

a =
     1

>> ~25

ans =
     0

>> x=[2 3 0 -1 6]l

>> x=[2 3 0 -1 6];
>> y=[3 5 5 1 -2];

>> x&y

ans =
     1     1     0     1     1

>> x|y

ans =
     1     1     1     1     1
```

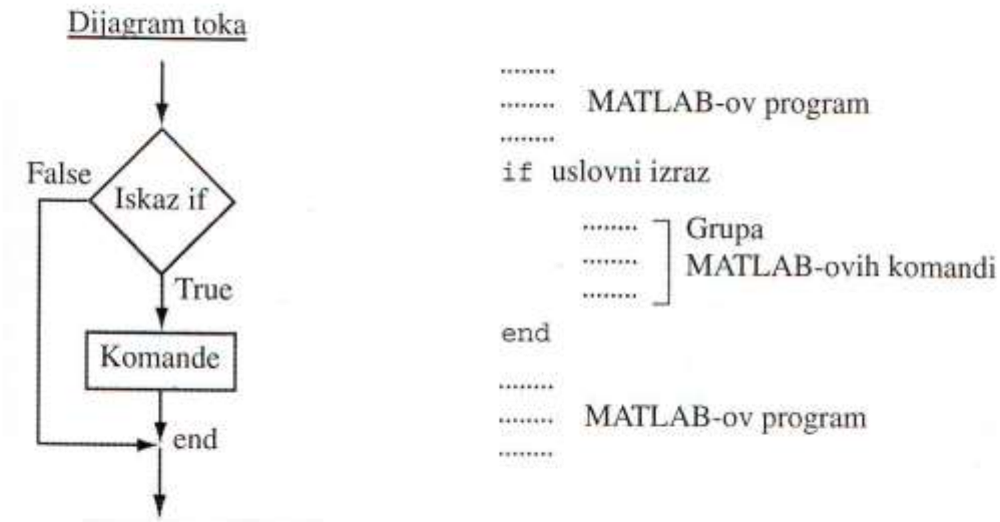
MATLAB ima ugađene logičke funkcije. To su:

`and(A,B)`	ekvivalentno `A&B`
`or(A,B)`	ekvivalentno `A|B`
`not(A)`	ekvivalentno `~A`
`all(A)`	Vraća 1 ako su svi elementi vektora A različiti od nule. Vraća 0 ako je samo jedan elemenat vektor A nula.
`any(A)`	Vraća 1 ako je makar jedan elemenat vektora A različiti od nule.
`find(A)`	Ako je A vektor, funkcija find vraća indekse elemenata koji su različiti od nule
`find(A>d)`	Vraća adrese elemenata vektora A koji su veći od skalara d

Uslovni iskazi

Uslovni iskaz je komanda koja omogućava MATLAB-u da odlučuje da li će izvršiti grupnu komandu koja sledi iskazu za uslovno izvršavanje, ili će te komande preskočiti. U uslovnom iskazu se zadaje uslovni izraz. Ako je rezultat tog izraza true (tačno), izvršava se grupa komandi koja neposredno sledi. Ako je rezultat izraza u uslovnom iskazu false (netačno), računar preskače tu grupu komandi. U MATLAB-u se koristi nekoliko struktura uslovnih iskaza:
`if-end`, `if-else-end`, `if-elseif-else-end`, i uslovni iskaz `switch-case`

Struktura `if - end`



Primer 4.17: Radnik je plaćen određeni iznos po satu za rad do 40 sati (nedeljno), a prekovremeni rad se plaća 50% više. Napišite program u obliku skript datoteke koji će obračunavati zaradu radnika. Program zahteva od korisnika da unese ukupan broj sati i iznos satnice, a zatim prikazuje ukupnu zaradu. Primeniti program za
a) broj radnih sati 35 i cenu satnice od 500 dinara i
b) broj radnih sati 50 i satnicu 600 dinara.

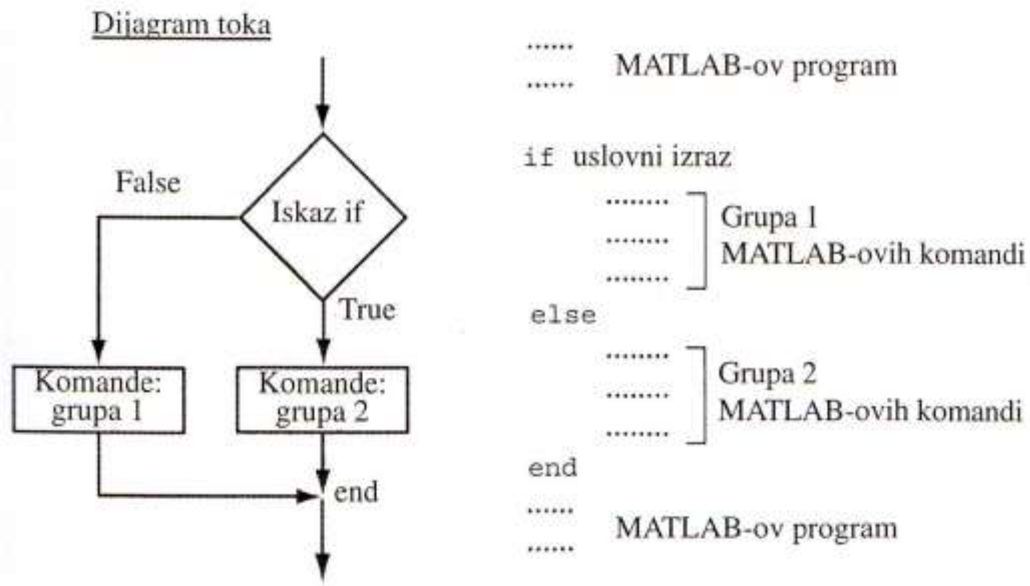
Program `zarada`:

```
%Program za obracun zarade radnika
%------------------------------------------------
s=input('Unesite broj radnih sati: ');
c=input('Unesite cenu satnice u RSD: ');
Zarada=s*c;
if s>40
    Zarada=Zarada+(s-40)*0.5*c;
end
fprintf('Zarada radnika je %g RSD. \n',Zarada)
```

Rezultat primene programa na konkretnim primerima:

```
>> zarada
Unesite broj radnih sati: 35
Unesite cenu satnice u RSD: 500
Zarada radnika je 17500 RSD.
>> zarada
Unesite broj radnih sati: 50
Unesite cenu satnice u RSD: 600
Zarada radnika je 33000 RSD.
```

Struktura `if-else-end`



Primer 4.18: Donji deo rezervoara vodotornja je u obliku valjka a gornji deo u obliku obrnute zarubljene kupe. Unutar rezervoara nalazi se plovak koji pokazuje nivo vode. Za $0 \le h \le 19m$, količina vode je $V = 12.5^2 \pi h$. Za $19 < h \le 33m$, količina vode se izračunava prema izrazu:
$V = 12.5^2 \cdot \pi \cdot 19 + \frac{1}{3}\pi(h-19)(12.5^2 + 12.5 \cdot r_h + r_h^2)$

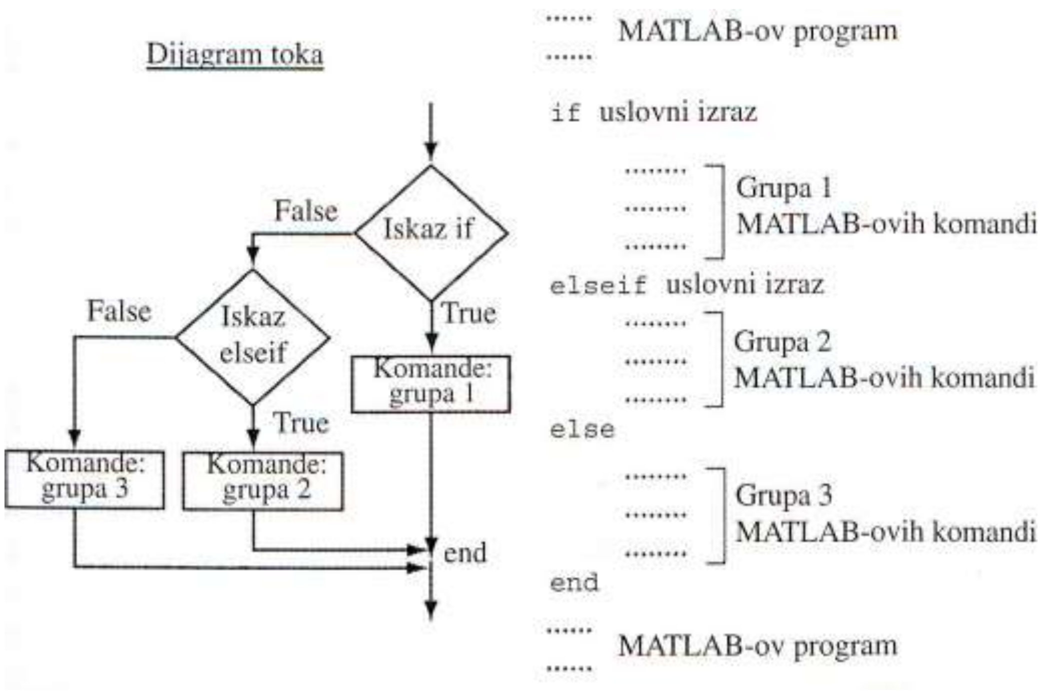
Gde je $r_h = 12.5 + \frac{10.5}{14}(h-19)$. Napisati funkcijski program za izračunavanje količine vode u rezervoaru u zavisnosti od položaja plovka (visine h).

```
function V=kolvode(h)
%Program kolvode izracunava kolicinu vode u vodotornju
%-------------------------------------------------------------------
%Ulazni argument je nivo vode h u m
%Rezultat je kolicina (zapremina) vode V u m^3
if h<=19
    V=12.5^2*pi*h;
else
    rh=12.5+10.5/14*(h-19);
    V=12.5^2*pi*19+pi*(h-19)*(12.5^2+12.5*rh+rh^2)/3;
end
```

Primena programa `kolvode` za h=14 m i za h=27,3 m

```
>> V1=kolvode(14)

V1 =

  6.8722e+003

>> V1=kolvode(27.3)

V1 =

  1.5767e+004
```

Struktura `if-elseif-else-end`



Primer 4.19: Napisati program u obliku skript datoteke za rešavanje kvadratne jednačine $ax^2 + bx + c = 0$. Neka ime datoteke bude `kvadkoreni`. Pošto se program pokrene, on treba da zahteva od korisnika da zada vrednosti konstanti *a*, *b* i *c*.

Objašnjenje: Da bi izračunao rešenja jednačine, program izračunava diskriminantu $D = b^2 - 4ac$. Ako je $D > 0$, program prikazuje poruku "Jednačina ima dva rešenja" a zatim ih prikazuje u sledećem redu. Ako je $D = 0$, program prikazuje poruku "Jednačina ima jedno rešenje" a zatim ga prikazuje u sledećem redu. Ako je $D < 0$, program prikazuje poruku "Jednačina nema realna rešenja"

Primeniti program za rešavanje sledećih jednačina:

$x^2 + 3x - 1 = 0$; $4x^2 + 10x + 6{,}25 = 0$; $15x^2 + 8x + 2 = 0$.

Program `kvadkoreni`:

```
%Program kvadkoreni za resavanje kvadratne jednacine
%-----------------------------------------------------------
a=input('Unesite konstantu a=');
b=input('Unesite konstantu b=');
c=input('Unesite konstantu c=');
D=b^2-4*a*c;
if D>0
    disp('Jednacina ima dva resenja')
    x1=-b/2/a-sqrt(D)/2/a;
    x2=-b/2/a+sqrt(D)/2/a;
    fprintf('Rešenja kvadratne jednačine su x1=%g i x2=%g. \n',x1,x2);
elseif D==0
    disp('Jednacina ima jedno resenje')
    x1=-b/2/a;
    fprintf('Rešenje kvadratne jednačine je x1=%g. \n',x1);
else
    disp('Jednacina nema realna resenja')
end
return
```

Rezultati primene programa na konkretne primere:

```
>> kvadkoreni
Unesite konstantu a=1
Unesite konstantu b=3
Unesite konstantu c=-1
Jednacina ima dva resenja
Rešenja kvadratne jednačine su x1=-3.30278 i x2=0.302776.
>> kvadkoreni
Unesite konstantu a=4
Unesite konstantu b=10
Unesite konstantu c=6.25
Jednacina ima jedno resenje
Rešenje kvadratne jednačine je x1=-1.25.
>> kvadkoreni
Unesite konstantu a=15
Unesite konstantu b=8
Unesite konstantu c=2
Jednacina nema realna resenja
```

Primer 4.20: Napraviti program pod imenom `kspoj` u obliku skript datoteke koji izračunava struje trofaznog kratkog spoja, jednofaznog kratkog spoja i dvofaznog kratkog spoja bez zemljospoja na sabirnicama jednog EES. Izrazi za struje kratkog spoja u zavisnosti od tipa su:

$I_{k3} = \frac{U_{nf}}{Z_d}$ trofazni; $I_{k1z} = \frac{3U_{nf}}{2Z_d + Z_0}$ jednofazni; $I_{k2} = \frac{\sqrt{3}U_{nf}}{2Z_d}$ dvofazni bez zemljospoja;

Uputstvo: ulazni podaci su ekvivalentne impedanse direktnog i nultog redosleda (Z_d, Z_0), i nominalni fazni napon sistema. Preporučljivo je da te vrednosti unosite sa tastature. Takođe i tip kratkog spoja definisati sa tastature.

Primeniti program za konkretne vrednosti: $Z_d = 32\Omega; Z_0 = 55\Omega; U_{nf} = 110/\sqrt{3}$ kV.

Program `kspoj`:

```
% Program kspoj za izracun. efekt. vrednosti struja kratkog spoja u EES
% Ulazni podaci:
Zd=input('Unesite vrednost za Zd=');
Z0=input('Unesite vrednost za Z0=');
Un=input('Unesite vrednost nominalnog napona Un=');
tip=input('Definisite tip kratkog spoja tip=');
% Oznake za tip k.spoja: 1 -jednofazni zemljospoj;
%                        2-dvofazni kratak spoj bez zemlejospoja,
%                        3-trofazni kratak spoj
if tip==1
    Ik=sqrt(3)*Un/(2*Zd+Z0);
elseif tip==2
    Ik=Un/(2*Zd);
elseif tip==3
    Ik=Un/sqrt(3)/Zd;
else
    disp('Pogresno ste uneli tip kratkog spoja');
end
fprintf('Struja kratkog spoja je: %f. \n',Ik)
return
```

Iskaz `switch - case`

Ovaj uslovni iskaz omogućava da se za izvršavanje izabere jedna od više mogućih grupa komandi. Struktura tog programa je:

```
........
        MATLAB-ov
........

switch izraz switch
        case vrednost 1
        .........
                grupa komandi 1
        .........
        case vrednost 2
        .........
                grupa komandi 2
        .........
        case vrednost 3
        .........
                grupa komandi 3
        .........
        otherwise
        .........
                grupa komandi 4
        .........
end
........
        MATLAB-ov
........
```

Primer 4.21: Korišćenjem iskaza `switch - case` kreirati skript datoteku koja će u zavisnosti od unetog meseca prikazivati poruku o godišnjem dobu.

Program `goddoba`:

```
%Program goddoba za prikazivanje god. doba u zavisnosti od unetog meseca
%---------------------------------------------------------------------------
mesec=input('Unesi mesec (1-12):');
%Uslovni iskaza switch-case:
switch mesec
    case {3 4 5}
        disp('Prolece');
    case {6 7 8}
        disp('Leto');
    case {9 10 11}
        disp('Jesen');
    case {12 1 2}
        disp('Zima');
    otherwise
        disp('Pogresan unos')
end
```

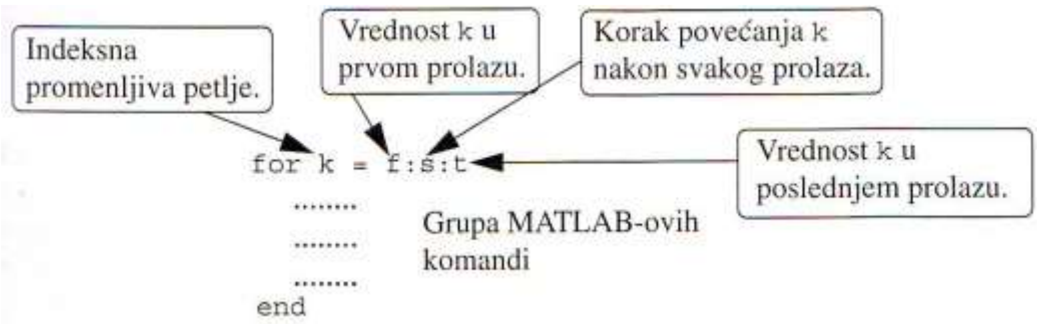
Rezultati izvršavanja programa u komandnom prozoru:

```
>> goddoba
Unesi mesec (1-12):4
Prolece
>> goddoba
Unesi mesec (1-12):7
Leto
>> goddoba
Unesi mesec (1-12):9
Jesen
>> goddoba
Unesi mesec (1-12):12
Zima
>> goddoba
Unesi mesec (1-12):13
Pogresan unos
>>
```

Petlje

Petlje predstavljaju još jedan način da se upravlja tokom programa. U petlji se izvršavanje komande, ili grupe komandi, ponavlja više puta zaredom. Svako izvršavanje petlje zove se prolaz. U svakom prolazu se barem jednoj promenljivoj, ili svim promenljivim unutar petlje dodeljuju nove vrednosti. MATLAB podržava petlje `for – end` u kojima se izvršavanje komande ili grupe komandi ponavlja zadati broj puta, i `while – end` koje se koriste kada ukupan broj prolaza nije unapred poznat, već se petlja izvršava dok je ispunjen zadati uslov.

Petlje: `for-end`



Primer 4.22: Napisati program za izračunavanje sume prvih n brojeva (celih brojeva).

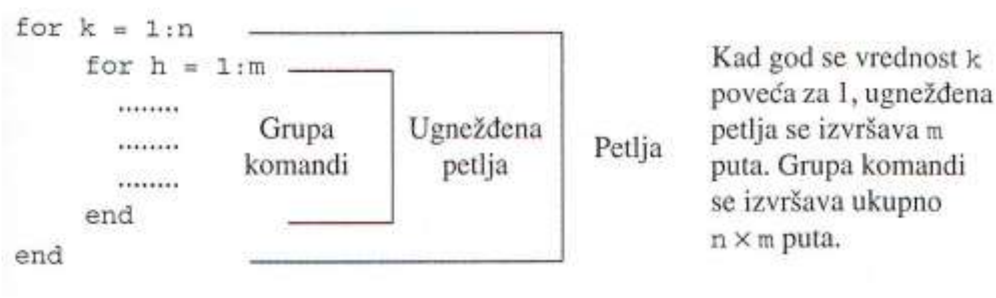
Program `suman`:

```
%Program za izracunavanje sume prvih n brojeva
%---------------------------------------------
n = input('Unesi ceo broj n=');
S = 0;
for k=1:n
S = S + k;
end
fprintf('Suma prvih n brojeva je %g \n',s);
```

Primena programa `suman` za n=7:

```
>> suman
Unesi ceo broj n=7
Suma prvih n brojeva je 28
```

Ugnežđene petlje



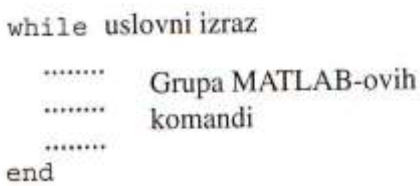
Primer 4.23: Napisati korisničku funkciju za generisanje kvadratne matrice proizvoljnih dimenzija, čiji su svi dijagonalni elementi jednaki 5, elementi u prvoj vrsti jednaki 1, dok su svi ostali elementi matrice jednaki nuli.

```
function [A]=specmatrica(n)
%Program specmatrica za generisanje matrice zadatog oblika
%-----------------------------------------------------------
disp(' ');
for k=1:n
    for kk=1:n
        if k==kk
            A(k,kk)=5;
        elseif k==1
            A(k,kk)=1;
        else
            A(k,kk)=0;
        end
    end
end
return
```

Primena:

```
>> A=specmatrica(4)

A =
     5     1     1     1
     0     5     0     0
     0     0     5     0
     0     0     0     5
```

Petlje: `while-end`



Primer 4.24:

```
x=0; %pocetna vrednost x je 1
while x<=15 %naredna komanda se izvrsava samo ako je x<=15
    x=x+3 %u svakom prolazu se vednosti x dodaje 3
end
```

4.4 Zadaci za vežbanje

Zadatak 4.1 Za početni ulog *U* i kamatnu stopu *k*, stanje *S* na računu nakon *n* godina iznosi:

$$S = U\left(1 + \frac{k}{100}\right)^n$$

Kamate se pripisuju godišnje. Napisati skript datoteku za izračunavanje stanja na štednom računu na kraju svake godine, u prvih *n* godina. Ulazne promenljive definisati u skript datoteci pomoću komande `input`. Rezultate prikazati tabelarno koristeći komandu `disp`, tako što će se u prvoj koloni prikazati godine (od 1 do n) a u drugoj odgovarajuće stanje na štednom računu. Isti rezultat prikaziti i grafički (grafik *S* u funkciji godina). Označiti ose na odgovarajući način.
Program primeniti za konkretan slučaj kada je: početni ulog 1000 €, a kamatna stopa 3,5% i period oročenja 10 godina.

Zadatak 4.2 Koristeći uslovni iskaz if-end i relacione operatore, napisati program u obliku skript datoteke za određivanje prosečnog učešća jednog stana u maksimalnom jednovremenom opterećenju stambenog objekta u kome se koristi centralno grejanje, koje se izračunava u zavisnosti od broja stanova prema izrazu:

$$P_{\max,1} = 8{,}5 \cdot \left(0{,}25 + \frac{0{,}75}{\sqrt{n}}\right) \quad \text{za } n \le 20$$

$$P_{\max,1} = \frac{1}{n} \cdot 5{,}1 \cdot n^{0{,}88} \quad \text{za } n \ge 20$$

Snaga se dobija u KW.
Nacrtati promenu snage $P_{\max,1}$ sa promenom broja stanova u opsegu $1 \le n \le 30$.

Zadatak 4.3 Napraviti program u obliku funkcijske datoteke (korisničku funkciju) za izračunavanje i grafičko prikazivanje sledeće funkcije:

$$y = \frac{2}{\sqrt{1 + (1{,}5t)^2}} \sin\left(1{,}5t + arctg(-1{,}5t)\right)$$

Primeniti program za $-10 \le t \le 10$ sa korakom od 0,1. Nakon izvršavanja funkcie u komandnom prozoru odrediti maksimalnu vrednost funkcije, kao i vrednost nezavisno premenjive *t* u kojoj se postiže maksimum funkcije.

Zadatak 4.4 Napraviti program u obliku skript datoteke za izračunavanje i grafičko prikazivanje sledeće funkcije:

$$y = \begin{cases} 4e^{x+2} & za - 6 \le x \le -2 \\ x^2 & za - 2 \le x \le -2{,}5 \\ (x + 6{,}5)^{1/3} & za - 2{,}5 \le x \le 6 \end{cases}$$

Treba odrediti i maksimalnu vrednost funkcije, kao i vrednost nezavisno premenjive *x* u kojoj se postiže maksimum funkcije.
Prikazivanje numeričkih rezultata na ekranu podesiti pomoću komandi `disp` i `fprintf`.

Grafik funkcije treba da bude kontinualni, sa isprekidanom linijom crne boje, debljine 3. Ose označiti na odgovarajući način.

Zadatak 4.6 Napraviti korisničku funkciju za određivanje dozvoljenog napona dodira u zavisnosti od vremena trajanja kvara, koji se računa prema sledećim izrazima:

$$U_{doz} = 1000V \quad \text{za} \quad t \le 0{,}075s$$

$$U_{doz} = \frac{75}{t}V \quad \text{za} \quad 0{,}075s \le t \le 1{,}153s$$

$$U_{doz} = 65V \quad \text{za} \quad t \ge 1{,}153s$$

Nacrtati funkciju $U_{doz}(t)$, za vrednosti $0 \le t \le 2s$ sa inkrementalnim priraštajem Δt=0,005s.
(Koristiti uslovni petlju `for-end` i uslovni iskaz `if-elseif-else-end`).

Zadatak 4.7 Napraviti program u obliku skript datoteke za izračunavanje zbira prvih *n* članova reda: $\sum_{k=1}^{n} \frac{(-1)^k \cdot k}{2^k}$. Izvršiti program za *n*=4 i *n*=20.

Zadatak 4.8 Napraviti korisničku funkciju za generisanje matrice dimenzija *m*x*n* u kojoj je vrednost svakog elementa razlika njegovih indeksa (rednog broja vrste i kolone). Primeniti program za matricu dimenzija 4x6.

Zadatak 4.9 Napisati korisničku funkciju koja pomoću uslovnog iskaza `if` i petlje `while` izračunava zbir pozitivnih elemenata nekog vektora. Primeniti program za vektor V=[1,-5,8,9,-6,0,4,-8].

4.1.

```
U=input('Unesite pocetni ulog u evrima, U=');
k=input('Unesite kamatnu stopu u procentima, k=');
n=input('Unesite period orocenja u godinama, n=');
n=1:n;
S=U*(1+k/100).^n;
disp(' ');
disp('Stanje na racunu:');
disp(' ');
disp('     n       S ');
disp(' (godina) (evra) ');
disp('--------------------------');
disp([n' S']); plot(n,S), xlabel('Godine'), ylabel('Stanje na racunu
(Evra)'), grid on;
```

4.2.

```
function [Pmax1]=pmax1(n)
%Program za izracunavanje prosecnog ucesca jednog stana u max jedn. opter.
zgrade
n=input('Unesi broj stanova, n=');
if n<=20
Pmax1=8.5*(0.25+0.75./sqrt(n));
else
Pmax1=1./n.*5.1.*n.^0.88;
end
end

>> Pmax1=pmax1(30)
Unesi broj stanova, n=30
Pmax1 =
3.3909
>> plot([1:30],pmax1),xlabel('Broj stanova'),ylabel('Pmax1 (kW)'), grid on
```

Unesi broj stanova, n=1:30

4.3.

```
function y=zad43(t)
%Funkcijski program za resavanje Zadatka zadatka 4.3
y=2./sqrt(1+(1.5*t).^2).*sin(1.5*t+atan(-1.5*t));
plot(t,y), xlabel('t'), ylabel('y'), grid on;

>> y=zad43([-10:0.1:10]);
>> [ymax,tmax]=max(y)
ymax =
0.72343
tmax =
117
```

4.4.

```
x=[-6:0.1:6];
dimx=length(x);
for k=1:dimx
    if x(k)<=-2
        y(k)=4*exp(x(k)+2);
    elseif -2<x(k)<=2.5
        y(k)=x(k)^2;
    else
        y(k)=(x(k)+6.5)^(1/3);
    end
end
[ymax,indymax]=max(y);
xmax=x(indymax);
fprintf('Maksimalna vrednost funkcije y je %g, koja se ima pri vrednosti x
od %g.\n',ymax,xmax);
plot(x,y,'--k','linewidth',3), xlabel('x'), ylabel('y'), grid on
```

4.6

```
function Udoz=udoz(t)
%Zadatak 4.6
for k=1:length(t)
if t(k)<=0.075
Udoz(k)=1000;
elseif t(k)>0.075 & t(k)<=1.153
Udoz(k)=75./t(k);
else
Udoz(k)=65;
end
end

>> t=[0:0.005:2];
>> Udoz=udoz(t);
>> plot(t,Udoz), xlabel('t (s)'), ylabel('Udoz (V)'), title('Dozvoljeni napon dodira'), grid on;
```

4.7

```
function s=zad47(n)
%Zadatak 4.7
s=0; k=0;
while k<n
    k=k+1;
    s=s+(-1)^k*k/2^k;
end

>> s= zad47 (4)
```

4.8.

```
function sm=zad48(m,n)
%Zadatak 4.8
for v=1:m
    for k=1:n
        sm(v,k)=v-k;
    end
end

>> sm=zad48(4,6)
```

4.9

```
function sp=zad49(v)
%Zadatak 4.9
dimv=length(v);
sp=0;
k=0;
while k<dimv
    k=k+1;
    if v(k)>0
        sp=sp+v(k);
    end
end
```

>> sp=zad49([1,-5,8,9,-6,0,4,-8])

5. POLINOMI I APROKSIMIRANJE PODATAKA

5.1 Polinomi

Polinomi su matematički izrazi koji se često koriste za modelovanje problema u prirodnim i tehničkim naukama. U mnogo slučajeva, jednačina do koje se dođe tokom rešavanja zadataka jeste polinom. Rešenje zadatka je nula ili koren tog polinoma.
Opšti oblik polinoma je:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0$$

Koeficijenti a_n, a_{n-1},...,a_1, a_0 su realni brojevi. n je ceo pozitivan broj i predstavlja red polinoma.
U MATLAB-u se polinomi predstavljaju vektorom vrstom čiji su elementi koeficijenti a_n, a_{n-1},...,a_1, a_0. Prvi element je koeficijent uz x sa najvišim stepenom. Vektor mora da sadrži sve koeficijente, uključujući i one jednake nuli. Na primer:

Polinom	**Predstavljanje u MATLAB-u**
$5x+2$	p=[5 2]
$x^3 - 2x^2 + 7$	p=[1 -2 0 7]
$4x^5 + 3x^3 - 10x^2 + 2x - 8.5$	p=[4 0 3 -10 2 -8.5]

Vrednost polinoma u tački x se može izračunati pomoću funkcije `polyval`

```
polyval(p,x)
```

gde je `p` vektor koji sadrži koeficijente polinoma a `x` broj ili promenljiva za koji treba izračunati vrednost polinoma.
U slučaju da je argument `x` vektor ili matrica, vrednost polinoma se izračunava za svaki element vektora ili matrice `x`, tako da je rezultat takođe vektor ili matrica koji sadrži odgovarajuće vrednosti polinoma.

Rešenja polinoma su vrednosti argumenta (x) za koje je vrednost polinoma jednaka nuli, pa se često zovu i nule ili koreni polinoma. U MATLAB-u se za nalaženje nula polinoma koristi funkcija `roots`.

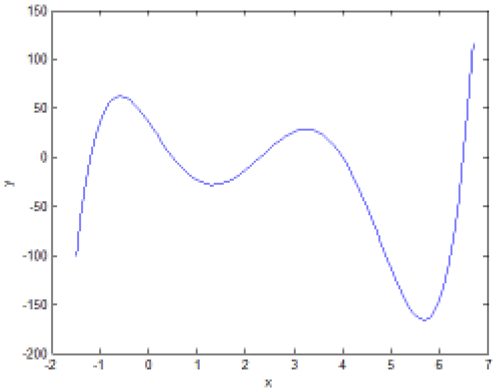
```
r=roots(p)
```

gde je `r` vektor koji sadrži rešenja (nule) polinoma, a `p` vektor koji sadrži koeficijente polinoma.

Primer 5.1 Za polinom $f(x) = x^5 - 12.1x^4 + 40.59x^3 - 17.015x^2 - 71.95x + 35.88$
a) Izračunati $f(5)$
b) Odrediti nule polinoma (korene)
c) Nacrtati grafik polinoma za $-1.5 \le x \le 6.7$

```
>> p=[1,-12.1,40.59,-17.015,-71.95,35.88];
>> polyval(p,5)

ans =

 -112.9950
```

```
>> roots(p)

ans =

    6.5000
    4.0000
    2.3000
   -1.2000
    0.5000

>> x=-1.5:0.1:6.7;
>> y=polyval(p,x);
>> plot(x,y), xlabel('x'), ylabel('y')
```



Kada su rešenja (nule) polinoma poznata, pomoću komande `poly` se mogu izračunati koeficijenti polinoma. Ova komanda ima sledeći oblik:

`p=poly(r)`

gde je p vektor vrsta s koeficijentima polinoma, a r vektor sa rešenjima (nulama) polinoma.

Primer 5.2 Ako su nule polinoma date vektorom r=[-3 1 0.3], odrediti koeficijente tog polinoma. Kog je reda taj polinom?

```
>> r=[-3 1 0.3];
>> p=poly(r)

p =

    1.0000    1.7000   -3.6000    0.9000
```

Shodno dobijenim koeficijentima, traženi polinom ima sledeći oblik: $f(x) = x^3 + 1.7x^2 - 3.6x + 0.9$

Sabiranje (oduzimanje) polinoma se vrši tako što se saberu (oduzmu) vektori koeficijenata polinoma. Ako polinomi nisu istog stepena, kraći vektor se mora dopuniti nulama.

Množenje dva polinoma se vrši pomoću funkcije `conv`, koja ima sledeći oblik:

`c=conv(a,b)`

gde je `c` vektor koeficijenata polinoma koji je rezultat množenja polinoma koji su predstavljeni vektorima `a` i `b`.
Pri množenju, polinomi ne moraju biti istog reda. Tri ili više polinoma se množe uzastopnom primenom funkcije `conv`.

Deljenje polinoma se ostvaruje pomoću funkcije `deconv`, koja ima sledeći oblik:

```
[q,r]=deconv(u,v)
```

gde je `q` vektor koeficijenata polinoma koji je količnik deljenja, `r` je vektor koeficijenata polinoma koji je ostatak deljenja, `u` je vektor koeficijenata polinoma koji je brojilac a `v` je vektor koeficijenata polinoma koji je imenilac.

Primer 5.3 Dati su polinomi $f_1(x) = 2x^3 + 9x^2 + 7x - 6$ i $f_2(x) = 3x^2 - 2x + 4$. Odrediti zbir, razliku, proizvod i količnik ova dva polinoma.

```
>> p1=[2,9,7,-6];
>> p2=[3,-2,4];
>> p=p1+[0 p2]

p =

     2    12     5    -2

>> m=p1-[0 p2]

m =

     2     6     9   -10

>> c=conv(p1,p2)

c =

     6    23    11     4    40   -24

>> [q,r]=deconv(p1,p2)

q =

    0.6667    3.4444


r =

         0         0   11.2222  -19.7778
```

Izvodi polinoma se dobijaju pomoću funkcije `polyder.` Može se upotrebiti i za izračunavanje izvoda proizvoda dva polinoma i količnika dva polinoma.

`k=polyder(p)`	Izvod jednog polinoma.
`k=polyder(a,b)`	Izvod proizvoda polinoma `a` i `b`
`[n,d]=polyder(u,v)`	Izvod količnika polinoma `u` i `v`. Vektori `n` i `d` predstavljaju koeficjente brojioca i imenioca izvoda količnika.

Primer 5.4 Dati su polinomi $f_1(x) = 3x^2 - 2x + 4$ i $f_2(x) = x^2 + 5$. Naći izvode ova dva polinoma, izvod proizvoda ova dva polinoma i izvod količnika ova dva polinoma.

```
>> f1=[3 -2 4];
>> f2=[1 0 5];
>> df1=polyder(f1)

df1 =

     6    -2
```

```
>> df2=polyder(f2)

df2 =

     2     0

>> df1f2=polyder(f1,f2)

df1f2 =

    12    -6    38   -10

>> [n d]=polyder(f1,f2)

n =

     2    22   -10


d =

     1     0    10     0    25
```

Jedina razlika kod zadavanja komande za izvod proizvoda i količnika polinoma je u broju izlaznih argumenata. MATLAB računa izvod proizvoda dva polinoma ako je zadat jedan izlazni argument. Ako su zadata dva izlazna argumenta izračunava se izvod količnika dva polinoma.

5.2 Aproksimiranje podataka krivom

Aproksimiranje podataka krivom je postupak pronalaženja funkcije koja predstavlja matematički model skupa tačaka koje predstavljaju određene podatke. S obzirom da postoji veliki broj različitih tipova funkcija (linearne, polinomske, stepene, eksponencijalne, itd) pronalaženje odgovarajuće funkcije za aproksimiranje određenog skupa podataka krivom može biti prilično složen zadatak.
U MATLABU se za aproksimiranje podataka krivom polinomskog tipa koristi funkcija `polyfit`. Ova funkcija koristi metodu najmanjih kvadrata, po kojoj se koeficijenti polinoma određuju minimizacijom zbira kvadrata razlika u svim tačkama.
Osnovni oblik funkcije `polyfit` je:

`p=polyfit(x,y,n)`

gde je `p` vektor koeficijenata polinoma koji aproksimira podatke; `x` je vektor s horizontalnim koordinatama tačaka (nezavisna promenljiva); `y` je vektor s vertikalnim koordinatama tačaka (zavisna promenljiva); `n` je stepen polinoma sa kojim želimo da aproksimiramo te podatke.

Primer 5.5. U jednom eksperimentu je u toku 24 časa, na svaka 2 časa merena struja jednog potrošača. Zabeležene su sledeće vrednosti:

T [h]	0	2	4	6	8	10	12	14	16	18	20	22
I [A]	25	30	32	40	46	51	54	60	67	73	80	88

Aproksimirati funkciju $I = f(T)$ polinomom pevog reda (linearnom funkcijom).

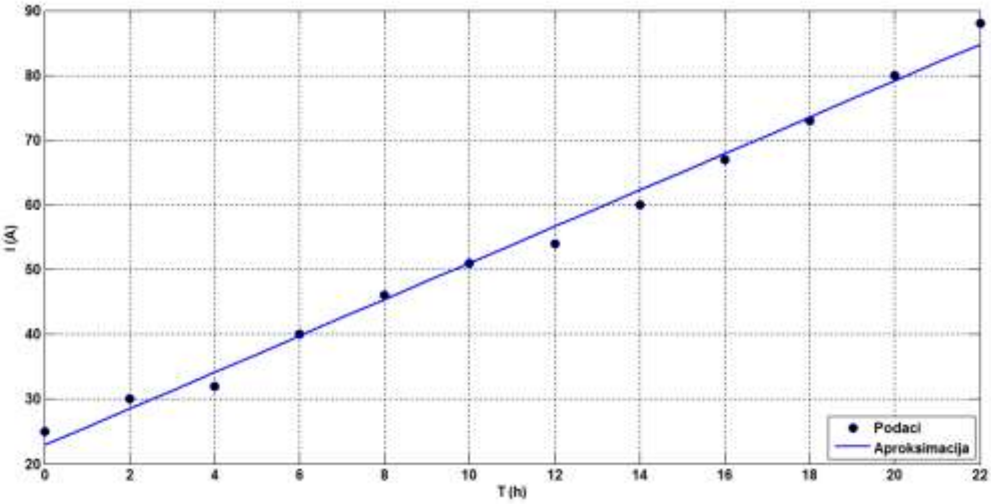
```
>> T=0:2:22;
>> I=[25 30 32 40 46 51 54 60 67 73 80 88];
>> IfT=polyfit(T,I,1)
```

```
IfT =

    2.8147   22.8718
```

Funkcija kojom se aproksimiraju izmerene struje je: $I(T) = 2.8147 \cdot T + 22.8718$

Npr. za T=9 h, dobija se: I(T)= 48.2041 A.

Na slici su prikazane izmerene vrednosti i prava koja ih aproksimira.



Primer 5.6 Za eksperimentalno određene parove vrednosti troškova goriva i snaga agregata datih u tabeli, odrediti koeficijente polinoma drugog reda kojim se aproksimira zavisnost troškova goriva od snage elektrane $C = f(P)$.

$C[NJ/h]$	1100	1580	2060
$P[MW]$	100	150	200

```
>> C=[1100,1580,2060];
P=[100,150,200];
CfP=polyfit(P,C,2)

CfP =

   -0.0000    9.6000  140.0000
```

5.3 Interpoliranje

Interpoliranje je izračunavanje vrednosti između dve poznate tačke. Jednodimenzionalno interpoliranje je ono kod koga svakoj tački odgovara jedna nezavisna (x) i jedna zavisna promenljiva (y). U MATLAB-u se jednodimenzionalno interpoliranje obavlja pomoću funkcije `interp1`

```
yi=interp1(x,y,xi,'metoda')
```

gde je `yi` interpolirana vrednsot, `x` je vector s horizontalnim koordinatama tačaka, `y` je vektor s vertikalnim koordinatama tačaka (zavisna promenljiva), `xi` je horizontalna koordinata tačke koja se interpolira. Metoda interpolacije je opciona komanda. Može biti `'linear'` – koristi linearno interpoliranje splajnom, `'spline'` – koristi kubno interpoliranje splajnom itd.

Primer 5.7 U tabeli su date tačke koje predstavljaju vrednosti funkcije $f(x)=1.5^x\cos(2x)$. Primenom metode linear i spline izračunati vrednosti y između datih tačaka. Nacrtati grafik za svaku metodu interpolacije. Na svakom grafiku prikazati početne tačke, krivu funkcije i krivu koja predstavlja metodu interpolacije.

x	0	1	2	3	4	5
y	1.0	-0.6242	-1.4707	3.2406	-0.7366	-6.3717

```
x=0:1:5;
y=[1 -0.6242 -1.4707 3.2406 -0.7366 -6.3717];
xi=0:1:5;

yilin=interp1(x,y,xi,'linear');
yispl=interp1(x,y,xi,'spline');

yfun=1.5.^xi.*cos(2*x);

subplot(1,2,1), plot(x,y,'*',xi,yilin,xi,yfun,'--')
subplot(1,2,2), plot(x,y,'*',xi,yispl,xi,yfun,'--')
```

5.4 Zadaci za vežbanje

Zadatak 5.1 Dat je polinom $f(x)=x^4-4x^2+1.2x+3$. Pomoću funkcije `polyval`:

a) Izračunati vrednost polinoma za x=4
b) Izračunati vrednost polinoma za $-3\le x\le 3$. Nacrtati grafik polinoma u datom opsegu nezavisno promenljive x.

Zadatak 5.2 Dati su polinomi $f_1(x)=4x^4+9x^3-2x^2-5x+6$ i $f_2(x)=x^2+2x+4$.

a) Odrediti korene (nule) ova dva polinoma
b) Odrediti zbir, razliku, proizvod i količnik ova dva polinoma.
c) Odrediti izvode polinoma, izvod proizvoda polinima i izvod količnika polinoma.

Zadatak 5.3 Dati su podaci u tabeli.

x	-5	-4	-2.2	-1	0	1	2.2	4	5	6	7
y	0.1	0.2	0.8	2.6	3.9	5.4	3.6	2.2	3.3	6.7	8.9

a) Pomoću funkcije `polyfit` aproksimirati podatke iz tabele pomoću polinoma prvog stepena. Nacrtati grafik tačaka i polinoma.
b) Pomoću funkcije `polyfit` aproksimirati podatke iz tabele pomoću polinoma trećeg stepena. Nacrtati grafik tačaka i polinoma.

Zadatak 5.4 U rednom RC kolu napunjen kondenzator nepoznate kapacitivnosti C se prazni kroz otpornik otpornosti $R=2000\Omega$. Dok se kondenzator prazni, napon na otporniku se meri u intervalima od 1 s tokom perioda od 10 s. Izmerene vrednosti napona na kondenzatoru su prikazane u tabeli.

t (s)	1	2	3	4	5	6	7	8	9	10
U (V)	9.4	7.31	5.15	3.55	2.81	2.4	1.26	0.97	0.74	0.58

Nacrtati grafik zavisnosti napona od vremena i odrediti kapacitivnost kondenzatora aproksimiranjem podataka iz tabele eksponencijalnom funkcijom koristeći funkciju `polyfit(x,log(y),1)`

(Pomoć: Vremenska promena napona na kondenzatoru u rednom RC kolu se odvija po eksponencijalnoj funkciji: $U = U_0 e^{-\frac{t}{RC}}$, gde je U_0 početni napon, R otpornost otpornika, a C kapacitivnost kondenzatora. Da bi se za aproksimiranje podataka eksponencijalnom krivom mogla upotrebiti funkcija `polyfit`, potrebno je eksponencijalnu funkciju napisati u formi linearne jednačine za `ln(U)` i `t` u sledećem obliku: $\ln(U) = -\frac{1}{RC}t + \ln(U_0)$. Ova jednačina ima oblik $y = mx + b$. Poređenjem prethodna dva izraza, zaključuje se da je t nezavisno promenljiva x, a $\ln(U)$ zavisna promenljiva y. Koeficijenti m i b se izračunavaju pomoću funkcije `polyfit`, a zatim se pomoću njihovih vrednosti izračunavaju $C = -\frac{1}{Rm}$ i $U_0 = e^b$)

Zadatak 5.5 Broj stanovnika Kine između 1940. godine i 2000. godine dat je u sledećoj tabeli:

Godina	1940	1950	1960	1970	1980	1990	2000
Broj stanovnika (u milionima)	537	557	682	826	981	1135	1262

a) Aproksimirati podatke pomoću kvadratne funkcije. Pomoću te funkcije proceniti broj stanovnika 1995. godine.
b) Interpolirati podatke metodom linear i splajn. Proceniti broj stanovnika 1995. godine pomoću obe metode.

U svim slučajevima nacrtati grafik tačaka podataka (označene kružićima) i aproksimiranu odnosno u tački (b) interpoliranu krivu.

(Napomena: Kina je 1955. godine imala tačno 614.4 miliona stanovnika.)

5.1.

```
>> % Definisanje polinoma
>> f=[1 0 -4 0 1.2 0 3]

f =

    1.0000     0    -4.0000    0     1.2000    0     3.0000

>> % a) Izracunavanje vrednosti polinoma za x=3
>> f3=polyval(f,4)

f3 =

       3094.20

>> % b) Odredjivanje nula (resenja-korena) polinoma
>> nulepolinoma_f=roots(f)

nulepolinoma_f =

   1.8394
  -1.8394
  -1.1399
   1.1399
        0 + 0.8261i
        0 - 0.8261i

>> % c)

>> xopseg=[-3:0.1:3];
>> fx=polyval(f,xopseg) ;
>> plot(xopseg,fx),xlabel('x'),ylabel('f');grid on
```

5.2.

```
>> f1=[4 9 -2 -5 6];
>> f2=[1 2 5];
>> % a)
>> n1=roots(f1)

n1 =

  -2.0000
  -1.3745
   0.5623 + 0.4791i
   0.5623 - 0.4791i

>> n2=roots(f2)

n2 =
```

```
  -1.0000 + 2.0000i
  -1.0000 - 2.0000i

>> % b)
>> Zf1f2=f1+[0 0 f2]

Zf1f2 =

     4     9    -1    -3    11

>> Rf1f2=f1-[0 0 f2]

Rf1f2 =

     4     9    -3    -7     1

>> Pf1f2=conv(f1,f2)

Pf1f2 =

     4    17    36    36   -14   -13    30


>> [kol ost]=deconv(f1,f2)
kol =

     4     1   -24


ost =

     0     0     0    38   126

>> % c)
>> Df1=polyder(f1)

Df1 =

    16    27    -4    -5

>> Df2=polyder(f2)

Df2 =

     2     2

>> Df1f2=polyder(f1,f2)

Df1f2 =

    24    85   144   108   -28   -13

>> [db di]=polyder(f1,f2)

db =
```

```
     8    33   116   136   -32   -37


di =

     1     4    14    20    25
```

5.3.

```
>> x=[-5 -4 -2.2 -1 0 1 2.2 4 5 6 7];
>> y=[0.1 0.2 0.8 2.6 3.9 5.4 3.6 2.2 3.3 6.7 8.9];
>> % a)
>> p1=polyfit(x,y,1)

p1 =

    0.5566    2.7694

>> % b)
>> p3=polyfit(x,y,3)

p3 =

    0.0219   -0.0481    0.0078    3.0839

>> % c)
>> y1=polyval(p1,x)

y1 =

   -0.0137    0.5430    1.5449    2.2128    2.7694    3.3261    3.9940
4.9959    5.5526    6.1092    6.6658

>> y3=polyval(p3,x)

y3 =

   -0.8994    0.8793    2.6004    3.0060    3.0839    3.0656    3.1021
3.7503    4.6638    6.1394    8.3086

>> plot(x,y,'b',x,y1,'--',x,y3,'*'),xlabel('x'),ylabel('y'),grid on
```

5.4.

```
>> t=1:10;
>> U=[9.4 7.31 5.15 3.55 2.81 2.4 1.26 0.97 0.74 0.58];
>> plot(t,U), xlabel('t[s]'), ylabel('U[V]'), title('U=f(t)'), grid on
>> R=2000;
>> mn=polyfit(t,log(U),1)

mn =

  -0.319478020319050   2.597066186470100
```

```
>> C=-1/(R*mn(1,1))

C =

  0.001565052893156

>> U0=exp(mn(1,2))

U0 =

 13.424295824527725
```

5.5.

```
>> god=[1940:10:2000];
>> brst=[537 557 682 826 981 1135 1262];
>> %a)
>> pa=polyfit(god,brst,2)
pa =
  1.0e+005 *

    0.0000   -0.0032    2.9961

>> brst1995=polyval(pa,1995)
brst1995 =
  1.1972e+003
>> %b)
>> godi=[1940:5:2000]; %zadavanje tacaka u kojima se interpol. funkc.
>> blin=interp1(god,brst,godi,'linear') %linearna interpolacija

blin =

  1.0e+003 *

    0.5370    0.5470    0.5570    0.6195    0.6820    0.7540    0.8260
0.9035    0.9810    1.0580    1.1350    1.1985    1.2620

>> blin1995=interp1(god,brst,1995,'linear') %proc. br. st. 1995

blin1995 =

  1.1985e+003

>> bspl=interp1(god,brst,godi,'spline')  %splajn interpolacija

bspl =

  1.0e+003 *

    0.5370    0.5272    0.5570    0.6131    0.6820    0.7531    0.8260
0.9024    0.9810    1.0596    1.1350    1.2036    1.2620

>> bspl1995=interp1(god,brst,1995,'spline')  %proc. br. st. 1995

bspl1995 =
```

```
  1.2036e+003

>> % c) Crtanje grafika
>> plot(god,brst,'o',god,polyval(pa,god),'b',godi,blin,'--',godi,bspl,'*-
'), xlabel('godina'),ylabel('broj stanovnika'),legend('podaci','kvadratna
aproksimacija','linearna interpolacija','splajn interpolacija');grid on
```

6. NUMERIČKA ANALIZA POMOĆU MATLABA

Numeričke metode se često koriste za rešavanje matematičkih zadataka u nauci i tehnici, kada je teško, pa čak i nemoguće doći do tačnih rešenja. MATLAB sadrži opsežnu biblioteku funkcija za rešavanje raznih kategorija matematičkih zadataka primenom numeričkih metoda. Ovde je pokazana primena tih funkcija, bez ulaženja u detalje o numeričkim metodama na kojima se one zasnivaju, a koje su predmet izučavanja u okviru matematičkih predmeta.

6.1 Rešavanje jednačine s jednom nepoznatom

Jednačina s jednom nepoznatom se u opštem slučaju može napisati u obliku *f*(*x*) = 0. Rešenje jednačine je vrednost *x* za koju funkcija preseca *x* osu (tj. za koju je vrednsot funkcije nula), što znači da u toj tački vrednost funkcije menja znak.
U MATLAB-u se nula funkcije može izračunati pomoću komande (funkcije) `fzero`, koja ima sledeći oblik:

```
x = fzero('funkcija',x0)
```

gde je:
`x` – rešenje, koje je skalarna vrednsot
`'funkcija'` – funkcija za koju se traži rešenje. Može se zadati na tri načina:

(1) Upisivanjem matematičkog izraza u obliku znakovnog niza
(2) Definisanje funkcije kao funkcijske datoteke, a ime funkcije se potpom zada kao znakovni niz
(3) Funkcija se može napisati kao lokalna funkcija a zatim se njeno ime unese u obliku znakovnog niza.

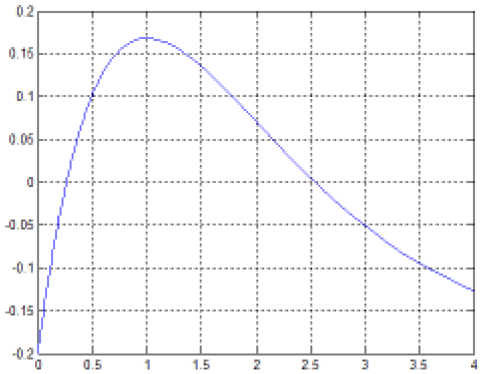
`x0` – vrednost *x* u blizini mesta gde funkcija preseca koordinatnu osu (početno pogađanje rešenja). Dobar način utvrđivanja početnog rešenja je crtanje krive funkcije.

U mnogim primenama u nauci i inženjerstvu može se proceniti opseg rešenja. Kada funkcija ima više od jednog rešenja, često se dešava da samo jednoo rešenje ima fizičkog smisla.

Primer 6.1 Naći rešenje jednačine $xe^{-x} = 0.2$.

Rešenje:
Najpre je potrebno prethodnu jednačinu napisati u obliku funkcije *f*(*x*) = 0, tj., $xe^{-x} - 0.2 = 0$. Radi procene početnog pogađanja rešenja, nacrta se funkcija pomoću komande fplot:

```
>> fplot('x*exp(-x)-0.2',[0,4]), grid on
```



Sa slike se vidi da funkcija preseca x osu na dva mesta. Da bi se pronašla rešenja funkcije, sada se dva puta primenjuje funkcija fzero, jednom za početno pogađanje 0.5 a drugi put za početno pogađanje 2.5.

```
>> x1=fzero('x*exp(-x)-0.2',0.5)

x1 =

    0.2592

>> x2=fzero('x*exp(-x)-0.2',2.5)

x2 =

    2.5426
```

Komanda fzero ima i dodatne opcije. Jedna od njih je `[x fval]=fzero('funkcija',x0)` koja dodeljuje vrednost funkcije u tački `x` promenljivoj `fval.`

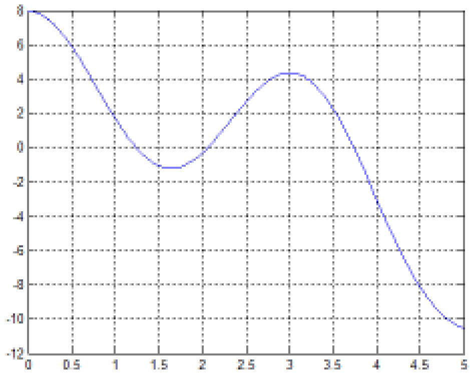
Primer 6.2 Rešiti jednačinu $\cos(x) = 2x^3$

```
>> resenje=fzero('2*x^3-cos(x)',1)

resenje =

    0.7214
```

Primer 6.3 Pronaći prva tri pozitivna rešenja jednačine $4\cos(2x) - e^{0.5x} + 5 = 0$

Rešenje:
Najpre treba nacrtati funkciju u opsegu nezavisno promenljive x od 0 do 5, kako bi se procenila približna rešenja (početna pogađanja):

```
>> fplot('4*cos(2*x)-exp(0.5*x)+5',[0 5]),grid on
```



Sada se primenjuje funkcija `fzero` za određivanje nula funkcije, izborom početnih rešenja u okolini presečnih tačaka sa grafika:

```
>> x1=fzero('4*cos(2*x)-exp(0.5*x)+5',1)

x1 =

    1.2374

>> x2=fzero('4*cos(2*x)-exp(0.5*x)+5',2)

x2 =

    2.0665

>> x3=fzero('4*cos(2*x)-exp(0.5*x)+5',4)

x3 =

    3.7375
```

Primer 6.4 Kutiju mase *m*=20 kg vuče uže koje je pod uglom θ u odnosu na ravan na kutije. Sila potrebna za pomeranje kutije data je formulom: $F = \dfrac{\mu m g}{\cos\theta + \mu \sin\theta}$, gde je koeficijent trenja μ=0.45 i g=9.81 m/s^2. Izračunati ugao θ ako je vučna sila 92 N.

Rešenje:
Najpre je potrebno prethodnu jednačinu napisati u obliku *f*(*x*) = 0, odnosno:
$F\cos\theta + F\mu\sin\theta - \mu m g = 0$, a zatim primeniti funkciju `fzero`:

```
>> teta=fzero('92*cos(x)+92*0.45*sin(x)-0.45*20*9.81',pi/6)

teta =

    0.9279

>> tetaustepenima=teta*180/pi

tetaustepenima =

   53.1653
```

6.2 Pronalaženje minimuma i maksimuma funkcije

Često je potrebno pronaći minimum ili maksimum funkcije oblika *y*=*f*(*x*). Vrednost *x* koja odgovara lokalnom minimum ili maksimumu funkcije određuje se tako što se izračuna nula izvoda funkcije. Vrednost *y* se zatim izračunava zamenom tako dobijene vrednsoti *x* u funkciju *y*=*f*(*x*).
U MATLAB-u se minimum funkcije s jednom promenljivom određuje pomoću komande `fminbnd`, koja ima sledeći oblik:

```
x = fminbnd('funkcija',x1,x2)
```

gde je `x` vrednost *x* za koju funkcija ima minimum. `x1` i `x2` su granice opsega u okviru koga se traži minimum funkcije.

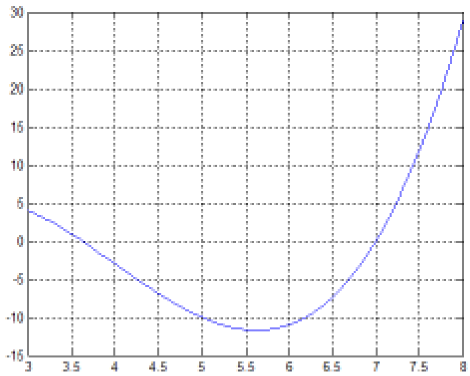
Vrednost funkcije na mestu minimuma može se odrediti primenom druge forme komande fminbnd:

```
[x fval]=fminbnd('funkcija',x1,x2)
```

Primer 6.5 Naći lokalni minimum funcije $f(x)=x^3-12x^2+40x-35$ u intervalu $3\le x\le 8$.

```
>> [x fval]=fminbnd('x^3-12*x^2+40*x-35',3,8)

x =

    5.6330


fval =

  -11.7093
```

Ako se promeni ova funkcija nacrta u navedenom opsegu, dobija se:



Komanda `fminbnd` se može upotrebiti i za pronalaženje maksimuma funkcije. To se radi tako što se funkcija pomnoži sa -1 i pronađe minimum.

Primer 6.6 Naći maksimum funkcije $f(x)=xe^{-x}-0.2$ u intervalu $0\le x\le 8$

```
>> [x fval]=fminbnd('-x*exp(-x)+0.2',0,8)

x =

    1.0000
```

```
fval =

   -0.1679
```

6.3 Numeričko integraljenje

Primenom operacije integrala se ostvaruju raznovrsna matematička izračunavanja. Neka od njih su izračunavanje površine, zapremine, brzine na osnovu ubrzanja, rada na osnovu sile, energije na osnovu snage, itd. Integraljenje jednostavnih funkcija se može obaviti analitički, ali kada su funkcije složenije, koristi se numeričko integraljenje. Operacija integraljenja se može obavljati nad funkcijom ili nad skupom tačaka koje predstavljaju određene podatke, dobijenih npr. merenjem neke veličine.
Numeričko integraljenje se obavlja tako što se ukupna površina obuhvaćena podintegralnom funkcijom i granicama integrala podeli na male segmente, čija se površina može prosto izračunati kao površina pravougaonika, a zati se izvrši sabiranje svih segmenata koji čine tu površinu. Razlika između pojedinih metoda numeričkog integraljenja leži u načinu deljenja površine na segmente i u načinu izračunavanja površina segmenata.
U MATLAB-u se za numeričko integraljenje koriste funkcije (komande): `quad` i `trapz`. Komanda `quad` se koriste za integraljenje funkcija, dok se `trapz` koristi kada se integrali skup tačaka.

Komanda `quad` koristi prilagodljivu Simpsonovu metodu integraljenja. Ima sledeći oblik:

```
q = quad('funkcija',a,b)
```

gde je `q` vrednost integrala podintegralne funkcije `'funkcija'` u granicama integrala `a` i `b`. Podintegralna funkcija se mora napisati za argument *x* koji je vektor, tj. primeniti operacije nad pojedninačnim elementima, tako da funkcija `quad` izračunava vrednost funkcije za svaki element vektora *x*.

Primer 6.7 Izračunati sledeći integral primenom numeričke integracije: $\int_0^8 \left(xe^{-x^{0.8}} + 0.2\right)dx$

```
>> quad('x.*exp(-x.^0.8)+0.2',0,8)

ans =

    3.1604
```

Primer 6.8 Izračunati sledeći integral: $\int_0^5 \frac{1}{0.8x^2 + 0.5x + 2}\,dx$

```
>> resenje=quad('(0.8*x.^2+0.5*x+2).^(-1)',0,5)

resenje =

    0.8774
```

Komanda `trapz` se koristi za integraljenje funkcije zadate u obliku skupa tačaka koje predstavljaju određene podatke. Primenjuje se metoda trapeza. Oblik ove komande je sledeći:

```
q = trapz(x,y)
```

gde su `x` i `y` vektori *x* odnosno *y* koordinata tačaka. Vektori moraju biti iste dužine.

6.4 Obične diferencijalne jednačine

Vrlo mali broj diferencijalnih jednačina se može rešiti analitički. Pomoću numeričkih metoda može se naći *rešenje* gotovo svake diferencijalne jednačine. MATLAB ima obimnu biblioteku funkcija za rešavanje diferencijalnih jednačina. Ovde je opisano rešavanje jednostavnih diferencijalnih jednačina prvog reda.
Obična diferencijalna jednačina (ODJ, eng. *ordinary differential equation*, *ODE*) sadrži nezavisnu promenljivu, zavisnu promenljivu i izvode zavisne promenljive. Ovde će biti razmatrane jednačine prvog reda koje imaju sledeći oblik:

$$\frac{dy}{dt} = f(t,y)$$

gde je t nezavisna promenljiva, a y je zavisna promenljiva. Rešenje je funkcija $y = f(t)$ koje ispunjava uslove jednačine. Pošto više funkcija može ispunjavati uslove date ODJ, za rešavanje zadatka potrebni su dodatni podaci. Dodatni podaci su vrednost funkcije (zavisne promenljive) za određenu vrednost nezavisne promenljive.

Postupak rešavanja ODJ prvog reda:

Korak 1: Napiše se zadatak u standardnom obliku.

$$\frac{dy}{dt} = f(t,y) \text{ za } t_0 \le t \le t_f \quad \text{uz } y = y_0 \text{ u } t = t_0$$

Kao što je rečeno, za rešavanje ODJ prvog reda potrebna su tri podatka: jednačina izvoda y u odnosu na t, interval nezavisne promenljive i početna vrednost y. Rešenje je vrednost y kao funkcija t između t_0 i t_f.

Primer zadatka za rešavanje:

$$\frac{dy}{dt} = \frac{t^3 - 2y}{t} \text{ za } 1 \le t \le 3 \quad \text{uz } y = 4.2 \text{ u } t = 1$$

Korak 2: Napiše se funkcijska datoteka koja izračunava $\frac{dy}{dt}$ za date vrednsoti t i y. Za navedeni primer, funkcijska datoteka bi bila:

```
function dydt=ODJpr1(t,y)
dydt=(t^3-2*y)/t;
```

Korak 3: Izabere se metoda rešavanja.

Metode rešavanja u MATLAB-u su funkcije.

Ime funkcije	Opis
`ode45`	Pogodna za jednostavne zadatke. Zasniva se na metodi Runge-Kuta

`ode23`	Pogodna za jednostavne zadatke. Često brža ali manje tačna od ode45
`ode113`	Pogodna za jednostavne zadatke. Radi u više koraka.
`ode15s`	Pogodna za teže zadatke. Koristi se kada ode45 ne uspeva.
`ode23s`	Pogodna za teže zadatke. Koristi se kada ode15s ne uspeva.
`ode23t`	Pogodna za umereno teške zadatke.
`ode23tb`	Pogodna za teže zadatke. Često je efikasnija od ode15s.

Korak 4: Rešava se zadatak

```
[t,y] = imefunkcije('imedat',topseg,y0)
```

gde su:
`imefunkcije` - ime numeričke metode koja se koristi (npr. `ode45`, `ode23`, itd.)
`'imedat'` – ime funkcijske datoteke koja izračunava $\frac{dy}{dt}$ za date vrednsoti t i y.
`topseg` – vektor koji zadaje interval rešenja.
`y0` – početna vrednost y (vrednost y u početnoj tački intervala).
`[t,y]` – Rešenje ODJ. `t` i `y` su vektori. Broj tačaka i razmak između početne i završne tačke zavise od ulaznog vektora `topseg`.

Na primer, rešenje zadatka navedenog u koraku 1:

$$\frac{dy}{dt} = \frac{t^3 - 2y}{t} \text{ za } 1 \le t \le 3 \text{ uz } y = 4.2 \text{ u } t = 1$$

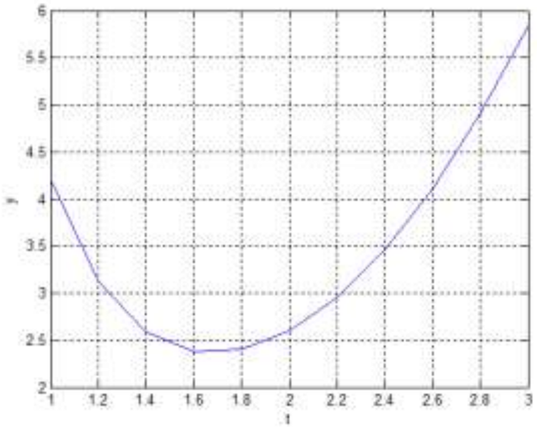
Može se dobiti na sledeći način:

```
>> [t y]=ode45('ODJpr1',[1:0.2:3],4.2)

t =

    1.0000
    1.2000
    1.4000
    1.6000
    1.8000
    2.0000
    2.2000
    2.4000
    2.6000
    2.8000
    3.0000


y =

    4.2000
    3.1234
    2.5896
    2.3817
```

```
     2.4010
     2.6000
     2.9560
     3.4592
     4.1069
     4.9006
     5.8444
```

Grafik krive rešenja se može jednostavno nacrtati pomoću komande plot:

```
>> plot(t,y),xlabel('t'),ylabel('y'),grid on
```



Primer 6.9 rešiti sledeću diferencijalnu jednačinu:

$$\frac{dy}{dt}=\frac{t^2-2t+3}{y^2} \text{ za } 0.5\le t\le 3 \text{ uz } y(0.5)=2$$

Nacrtati krivu rešenja.

Rešenje:

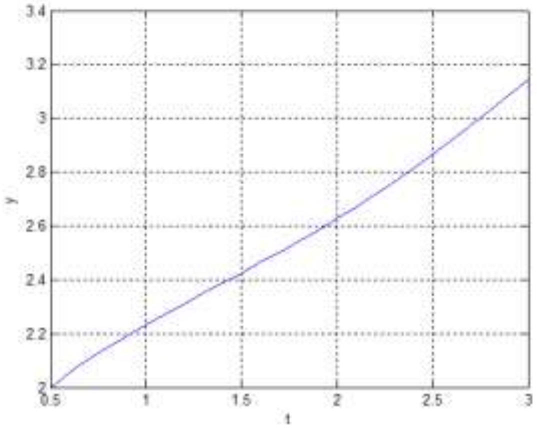
Najpre se formira funkcijska datoteka koja izračunava $\frac{dy}{dt}$ za date vrednosti t i y.

```
function dydt=ODJpr9(t,y)
dydt=(t^2-2*t+3)/y^2;
```

Sada se primenjuje funkcija za numeričko rešavanje diferencijalne jednačine:

```
>> [t y]=ode45('ODJpr9',[0.5:0.1:3],2)
t =
     0.5000
     0.6000
     0.7000
     0.8000
     0.9000
     1.0000
     1.1000
     1.2000
     1.3000
```

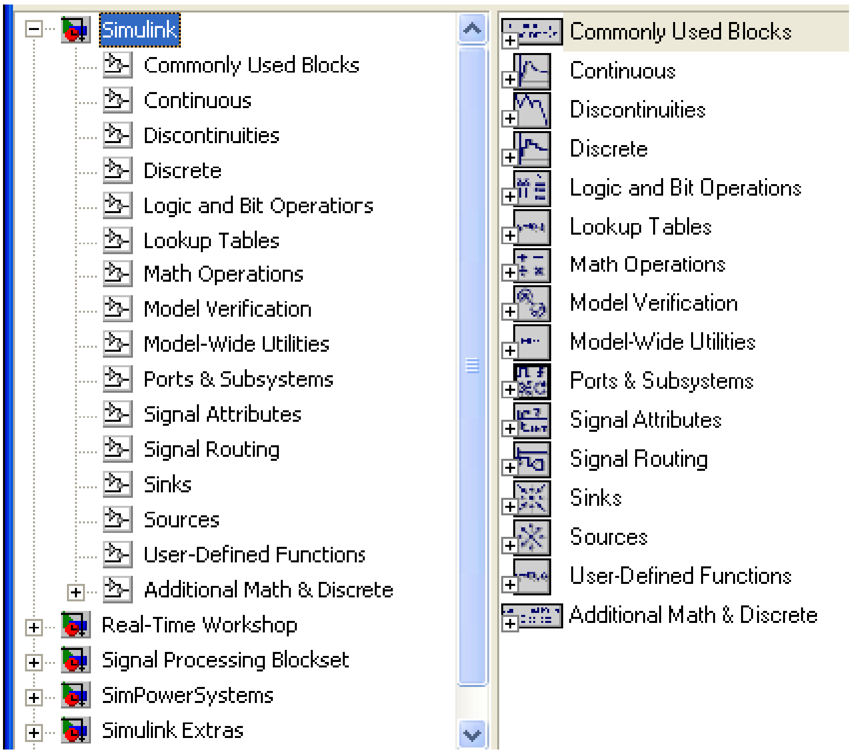
```
    1.4000
    1.5000
    1.6000
    1.7000
    1.8000
    1.9000
    2.0000
    2.1000
    2.2000
    2.3000
    2.4000
    2.5000
    2.6000
    2.7000
    2.8000
    2.9000
    3.0000


y =

    2.0000
    2.0536
    2.1028
    2.1485
    2.1914
    2.2324
    2.2719
    2.3104
    2.3484
    2.3863
    2.4244
    2.4630
    2.5023
    2.5426
    2.5840
    2.6268
    2.6710
    2.7168
    2.7643
    2.8134
    2.8643
    2.9170
    2.9715
    3.0278
    3.0858
    3.1456

>> plot(t,y),xlabel('t'),ylabel('y'),grid on
```



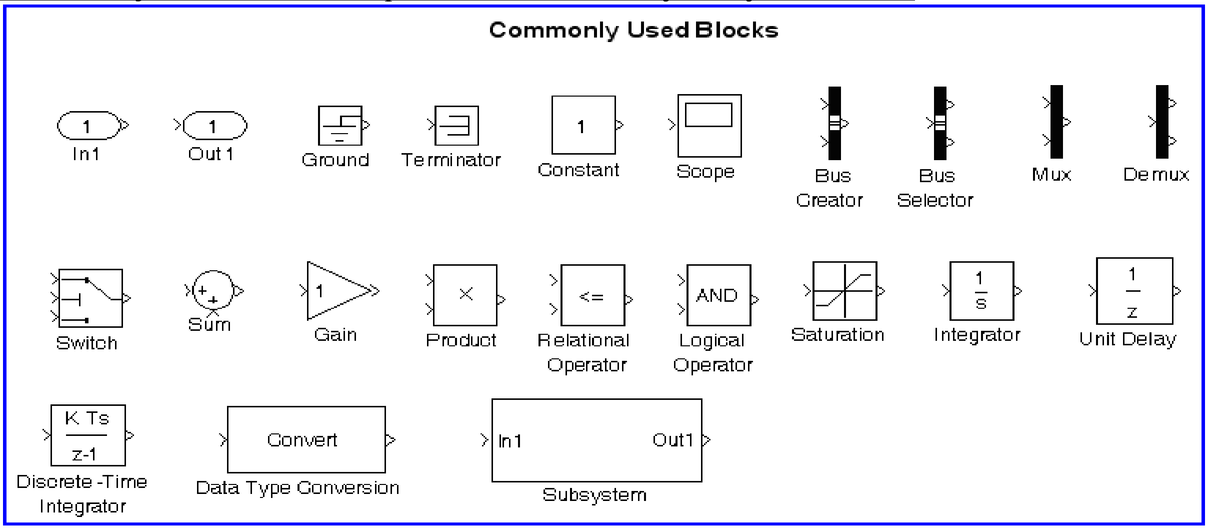
7. SIMULINK

SIMULINK je modul integrisan u MATLAB. Služi za simulaciju dinamike linearnih, nelinearnih, vremenski kontinualnih ili diskretnih multivarijabilnih sistema sa koncentrisanim parametrima. Elementi sistema se predstavljaju pomoću blokova. Svaki blok predstavlja matematički model određenog elementa. Povezivanjem blokova gradi se sistem, koji se naziva SIMULINK model. Simulacija dinamike sistema se ostvaruje pomoću SIMULINK ili MATLAB funkcija za numeričko rešavanje diferencijalnih jednačina, kojima je opisan sistem, odnosno SIMULINK model. Pri kreiranju SIMULINK modela, automatski mu se pridružuje m-datoteka, koja prestavlja niz MATLAB i SIMULINK komandi i funkcija.

7.1 Osnovne grupe blokova u SIMULINK-u



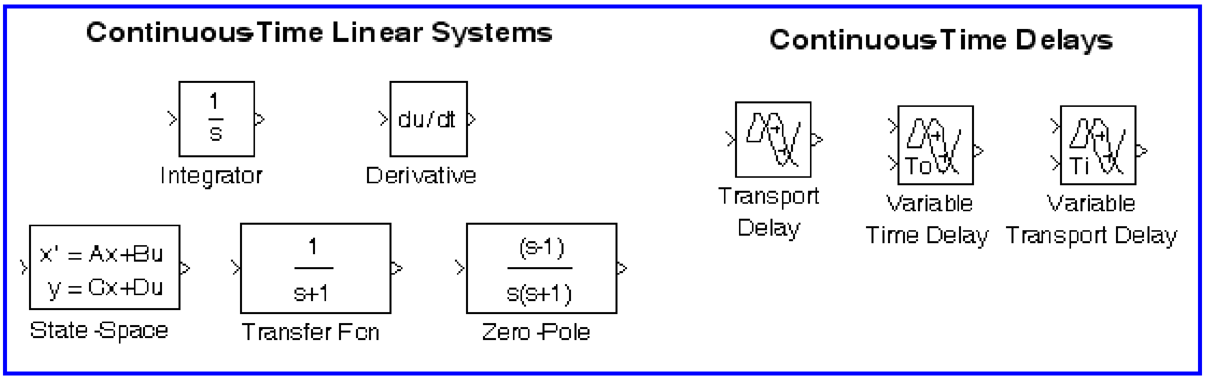
Commonly Used Blocks – Opšti blokovi, blokovi koji se najčešće koriste.



Gde su:
In1 – ulaz u posmatrani sistema
Out1 – izlaz iz posmatranog sistema
Ground – uzemljuje ulazni port koji nije povezan sa drugim blokovima
Terminator – povezuje izlaz bloka koji nije povezan sa drugim blokovima (zatvara blok)
Constant – generator konstantne vrednosti (realne ili kompleksne)
Scope – prozor za grafičko prikazivanje rezultata u funkciji vremena trajanja simulacije (nema vidljive oznake na osama)
Mux – multiplekser
Demux – demultiplekser
Switch – izlaz je prvi ili treći ulaz u zavisnoti od vrednosti na drugom ulazu
Sum – sabirač

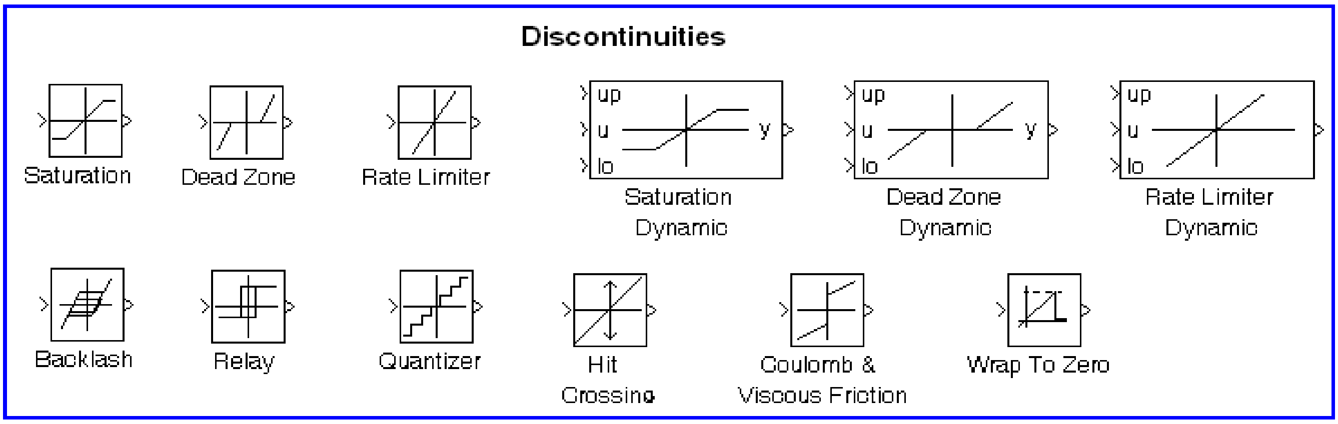
Gain – pojačavač
Product – množač
Relational Operator – relacioni operator
Logical Operator – logički operator
Saturation – zasićenje (ograničava ulazni signal između gornje i donje granice)
Integrator – integrator
Unit Delay – jedinično kašnjenje
Discrete-Time Integrator – diskretni integrator
Data Type Conversion – pretvaranje i skaliranje ulaznih signala zadati tip podataka na izlazu
Subsystem – deo ukupnog sistema

Continuous – Kontinualni sistemi

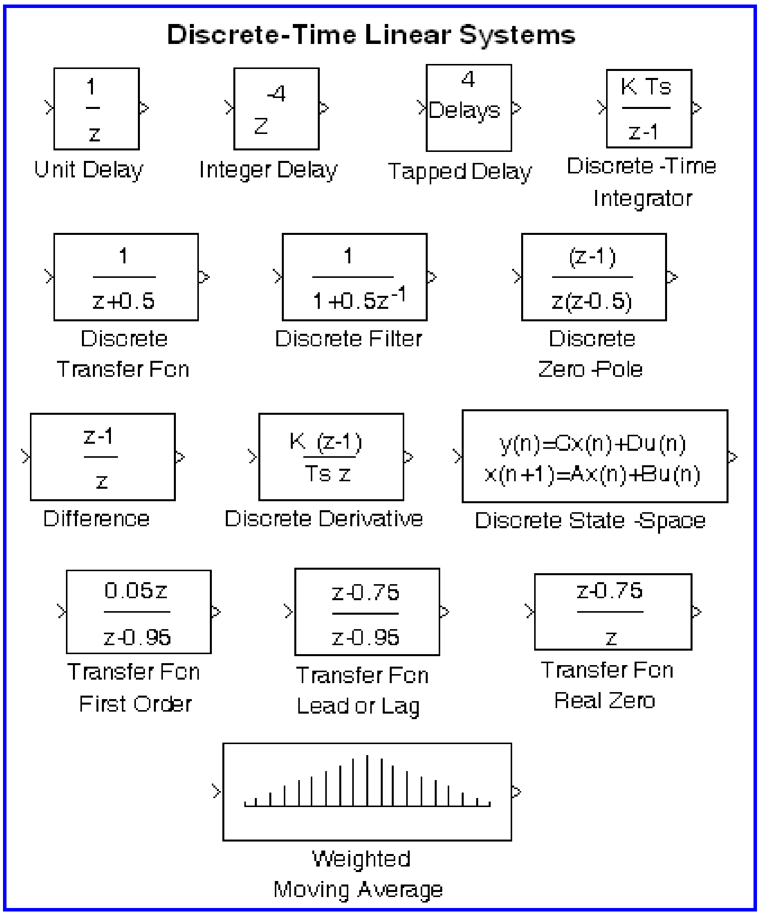


Derivate – numerički diferencijator
State Space – sistem definisan preko modela u prostoru stanja
Transfer Fcn – transfer funkcija; sistem definisan preko broioca i imenioca prenosne f-je.
Zero Pole – sistem definisan preko nula, polova i pojačanja prenosne funkcije

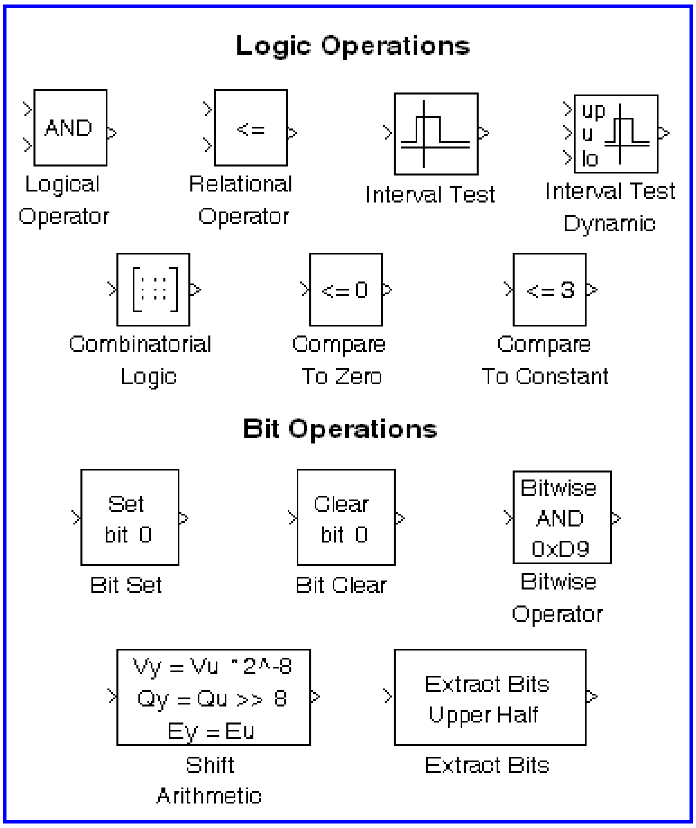
Discontinuities – Diskontinualni sistemi



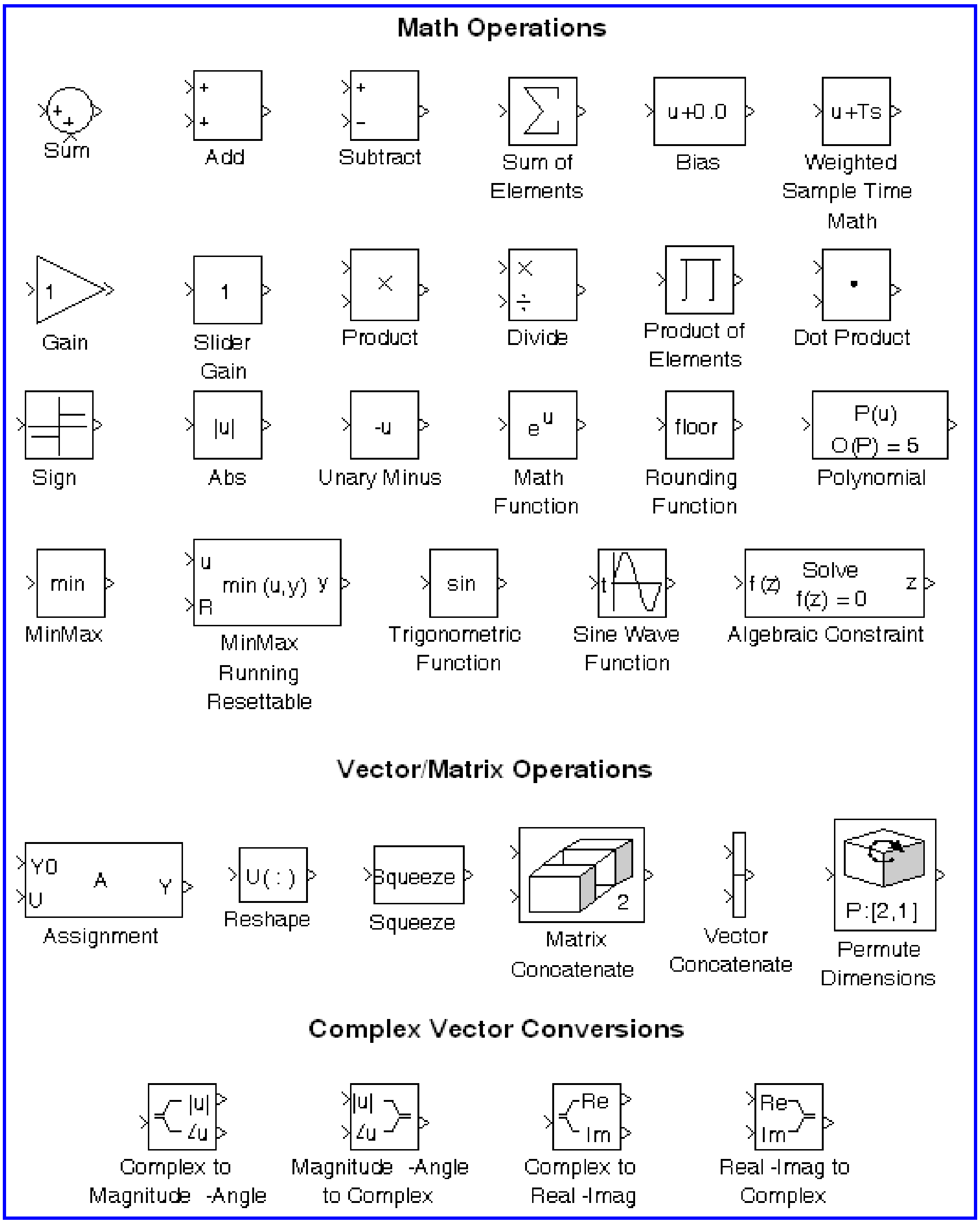
Discrete – Diskretni sitemi



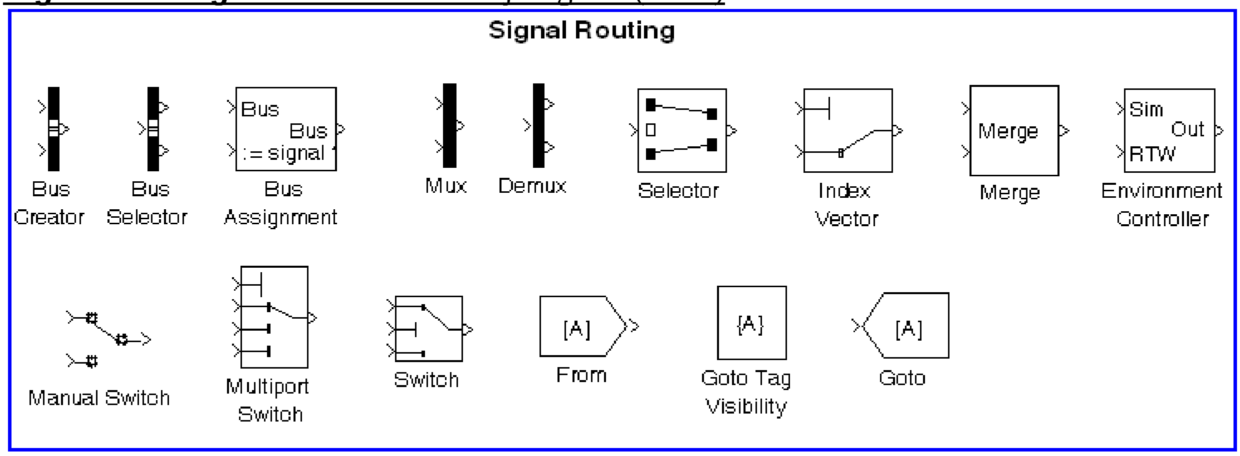
Logic and Bit Operations – Logički i bit operatori



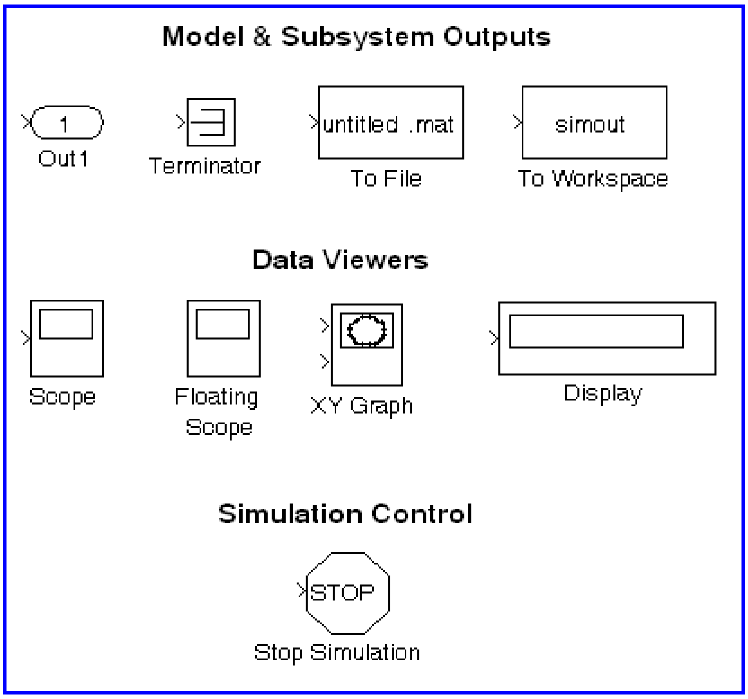
Math Operations – Matematički operatori



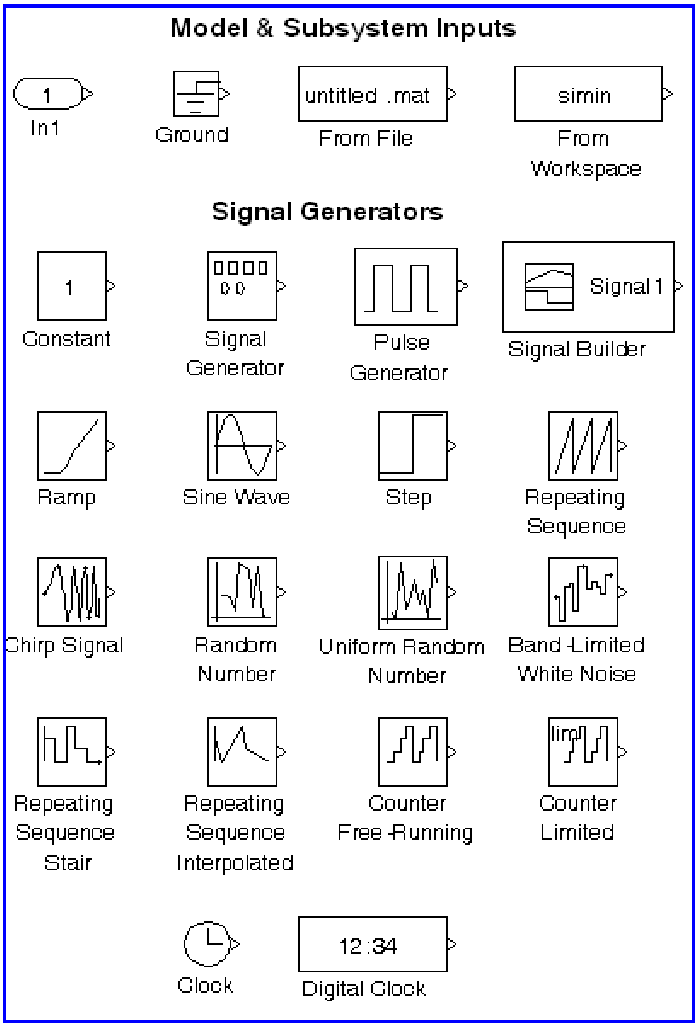
Signal Routing – Blokovi za trasiranje signala (ruteri)



Sinks – Blokovi za prikazivanje rezultata simulacije



Sources – Blokovi za generisanje ulaznih signala

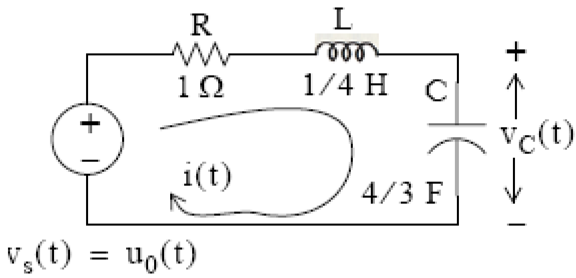


7.2 Formiranje SIMULINK modela

SIMULINK modeli se formiraju u odgovarajućem prozoru koji se otvara kada se izabere: **File, New, Model** iz menija komandnog prozora MATLAB-a. Posle otvaranja prozora, u prozor se ubacuju potrebni blokovi iz odgovarajuće grupe blokova. Blokovi se uzimaju tako što se prvo otvori prozor ogovarajuće grupe blokova, zatim se strelica miša dovede do odgovarajućeg bloka, pritisne se levi taster miša i blok jednostavno prevuče u novootvoreni prozor SIMULINK modela. Kada se svi potrebni blokovi smeste u prozor SIMULINK modela, vrši se njihovo povezivanje. Zatim se definišu parametri blokova u dijalogu za definisanje parametara, koji se otvara dvostrukim pritiskom miša na blok. Nakon toga se iz menija **Simulation** bira opcija **Parameters** i vrši podešavanje parametara simulacije, što podrazumeva izbor jedne od raspoloživih metoda za numeričko rešavanje običnih diferencijalnih jednačina kojima je opisan model, kao i podešavanje koraka integracije, vremena simulacije, itd. Na kraju, izborom opcije **Simulation**, **Start** počinje simulacija.

Primer 7.1 Za električno kolo na slici je pri početnim uslovima $i_L(0^-) = 0$, i $v_C(0^-) = 0{,}5$ V određena diferencijalna jednačina promena napona na kondenzatoru:

$$\frac{d^2 v_C}{dt^2} + 4\frac{dv_C}{dt} + 3v_C = 3u_0(t) \qquad t > 0.$$



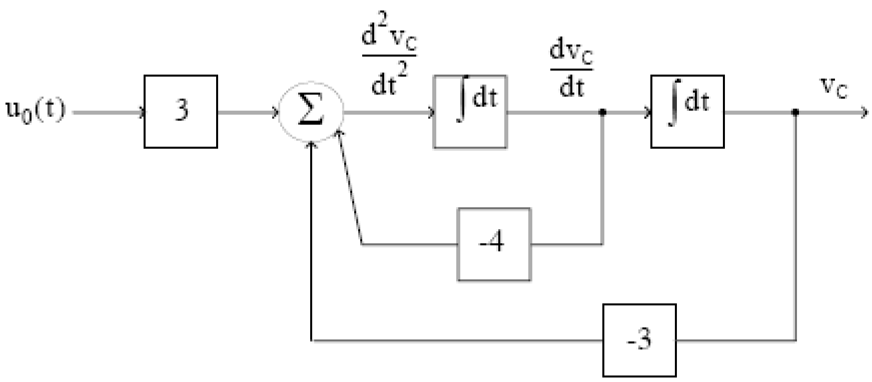
Napraviti SIMULINK model za rešavanje ovog kola.

Rešenje:

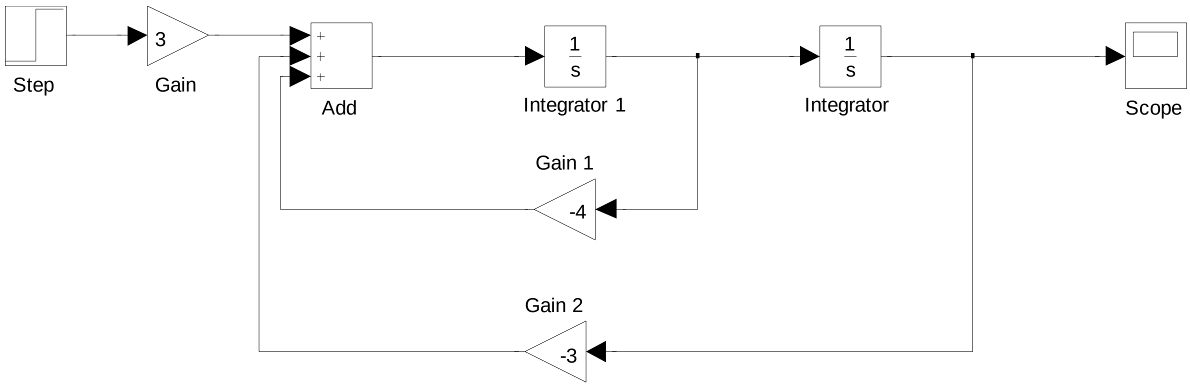
Gornja diferencijalna jednačina se može napisati u sledećem obliku:

$$\frac{d^2 v_C}{dt^2} = -4\frac{dv_C}{dt} - 3v_C + 3u_0(t) \qquad t > 0$$

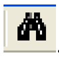
Ova jednačina se može predstaviti pomoću blok dijagrama na sledećoj slici:

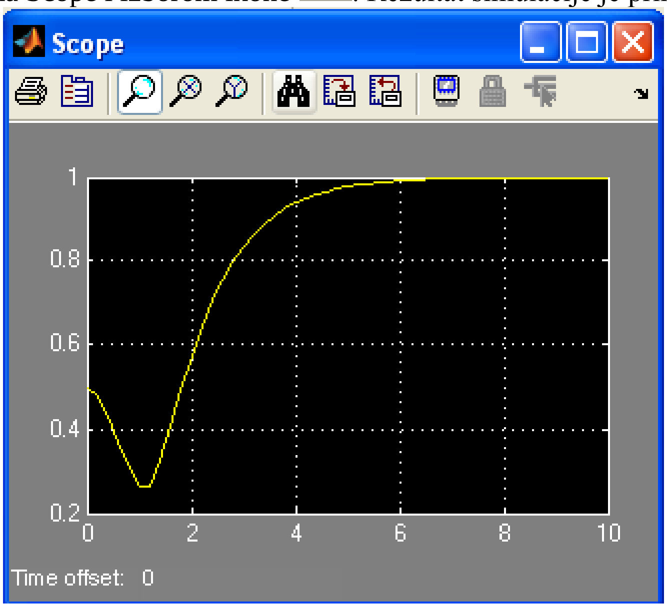


Sada se može formirati SIMULINK model primenom odgovarajućih blokova, prema prethodnom dijagramu.

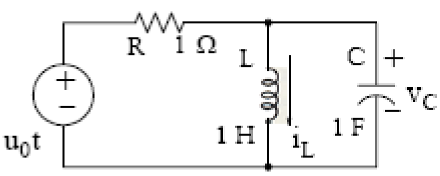


Blok Step se uzima iz grupe blokova Sources; Blokovi Gain i Add se uzimaju iz krupe blokova Math Operators; Blok Integrator se uzima iz grupe blokova Continuous; Blok Scope iz grupe blokova Sinks.

Nakon formiranja SIMULINK modela podešavaju se parametari blokova. Prvi početni uslov (0) se upisuje u prvi integrator a drugi početni uslov (0,5) u drugi integrator. Nakon toga se podese parametri simulacije (prihvataju se automatski podešeni parametri). Zatim se model sačuva pod određenim imenom (opcija save) i tek tada se može izvršiti simulacija. Simulacija se vrši pritiskom na ikonu  koja označava Start simulation ili iz menija Simulation odabere Start. Rezultat simulacije se prikazuje dvostrukim pritiskom na Scope i izborom ikone . Rezultat simulacije je prikazan na sledećoj slici.

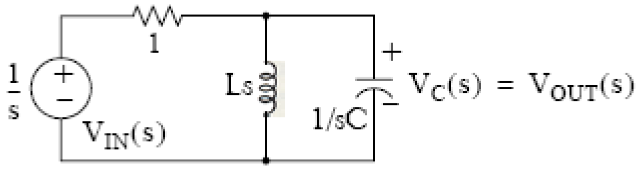


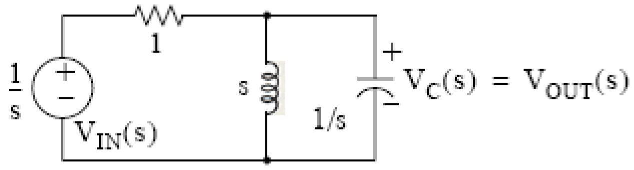
Primer 7.2 Za električno kolo na slici, formirati SIMULINK model za simulaciju i prikazivanje promene napona na kondenzatoru. Ulazni napon je jedinična odskočna funkcija vremena (step funkcija). Početni islovi u kolu su $i_L(0)=0$ i $u_C(0)=0$. Koristiti transfer prenosnu funkciju (Transfer Fcn). Za prikazivanje rezultata izabrati blok Scope.



Rešenje:

Prvo se električno kolo prevede u s – domen:



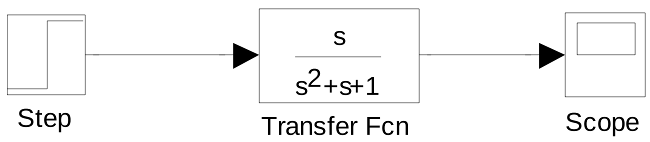


$$V_{OUT}(s) = \frac{(s \cdot 1/s)/(s + 1/s)}{(s \cdot 1/s)/(s + 1/s) + 1} \cdot V_{IN}(s) = \frac{s}{s^2 + s + 1} \cdot V_{IN}(s)$$

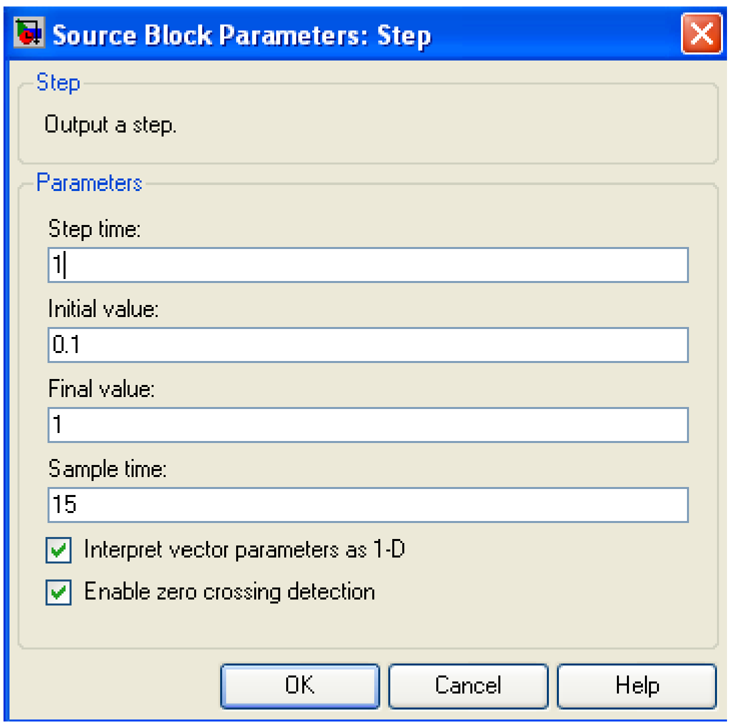
Tako da je prenosna ili transfer funkcija:

$$G(s) = \frac{V_{OUT}(s)}{V_{IN}(s)} = \frac{s}{s^2 + s + 1}$$

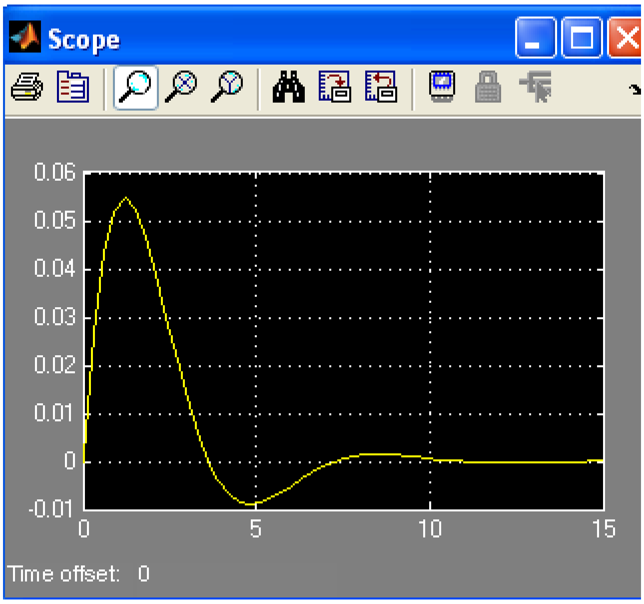
Sada se formira SIMULINK model:



Pošto u trenutku t = 0, step funkcija nije definisana, onda se parametri Step bloka podešavaju na sledeći način:

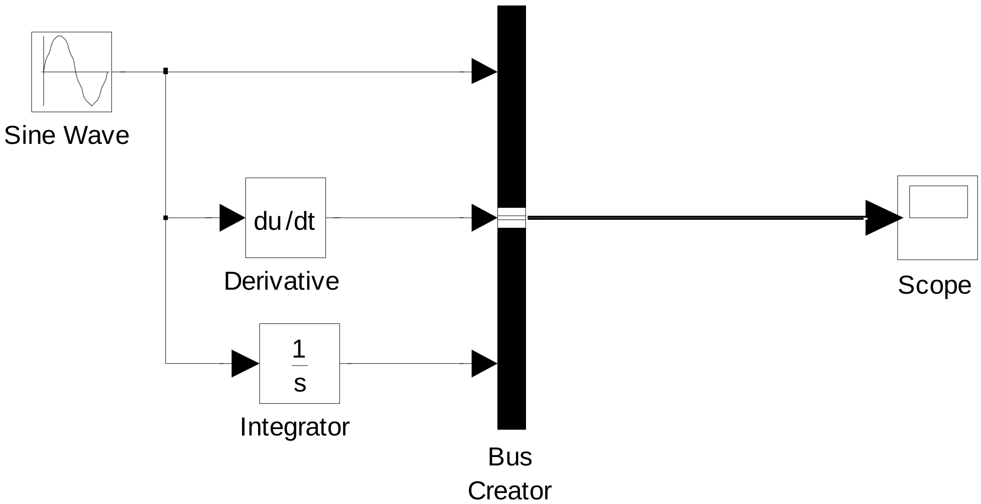


Pri podešavanju parametara simulacije, za vreme trajanja simulacije treba upisati 15 s. Nakon toga, pokrene se simulacija i dobija se sledeći rezultat:

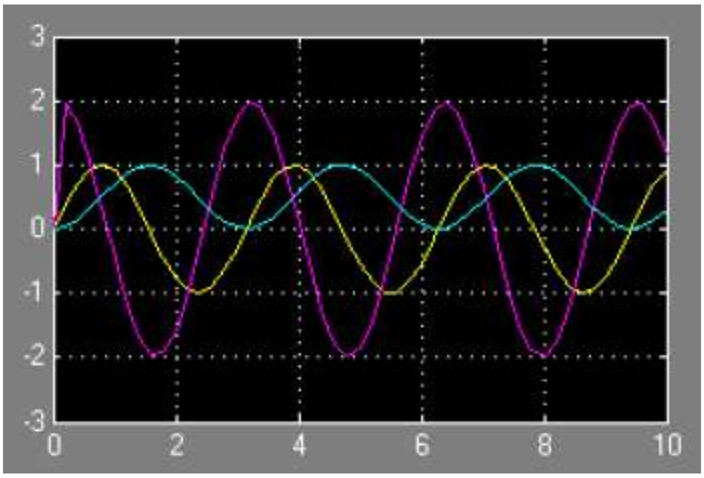


Primer 7.3 Kretirati SIMULINK model za kombinovanje istovremeno prikazivanje funkcija: $\sin 2t$, $\frac{d}{dt}\sin 2t$ i $\int \sin 2t dt$.

Rešenje:



U ulaznom bloku Sine Wave se definise funkcija $\sin 2t$. Rezultat simulacije je grafik sve tri funkcije:



Primer 7.4 Za sisteme čije su prenosne funkcije:

a) $\frac{Y(s)}{X(s)} = \frac{5}{15s^2 + 10s + 1}$; b) $\frac{Y(s)}{X(s)} = \frac{10s + 1}{15s^2 + 10s + 1}$; c) $\frac{Y(s)}{X(s)} = \frac{10s^2 + 5s + 1}{15s^2 + 10s + 1}$;

formirati SIMULINK model za određivanje odziva na različite vrste ulaznog signala (step, constant, ramp, sine wave...). Za prikazivanje rezulatata simulacija koristiti blokove: Scope, To Workspace i XY Gaph.

Rešenje: