

Jezik Matlab

- programski jezik visoke razine
- zapis podataka u obliku *matrica/polja*
- kontrola toka programa, funkcija, strukture podataka, ulaza/izlaza i objektno orijentiranih dijelova programa

Matematičke funkcije

- zbirka matematičkih funkcija od jednostavnih sve do veoma kompleksnih (npr. interpolacija, polinomske funkcije, računanje s matricama, Besselove funkcije, brza Fourierova transformacija, svojstvene vrijednosti itd.)

Radno okruženje

- zbirka alata s kojima korisnik izvodi funkcije

Grafički sustav

- prikaz 2D i 3D grafike visoke kvalitete
- obrada slike, animacija, alati za grafičko korisničko sučelje (GUI)

Alati za izradu aplikacija (**Application Development Tools**)

- izrada programa u programskim jezicima C i Fortran koje se mogu uključiti u MATLAB projekte
- pozivanje rutina iz MATLAB ("dinamičko povezivanje")
- pozivanje MATLAB - a kao jedinice za računanje
- alati za izradu i distribuciju samostalnih verzija MATLAB aplikacija (MATLAB compiler, C/C++ Math)

TOOLBOXES

- kolekcija gotovih funkcija prilagođenih za posebne aplikacije koje daju dodatne mogućnosti MATLAB-u i Simulink-u
- funkcije su ugrađene u MATLAB jezik i mogu se jednostavno pregledavati i modificirati ("otvoreni kod")

Npr.

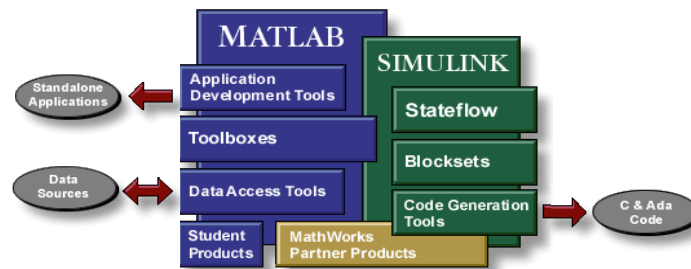
- Simboličko računanje
- Podešavanje krivulja („fitanje“)
- Identificiranje sustava
- Optimiranje procesa
- Neuronske mreže, itd.

Alati za akviziciju i pristup podacima (Data Access Tools)

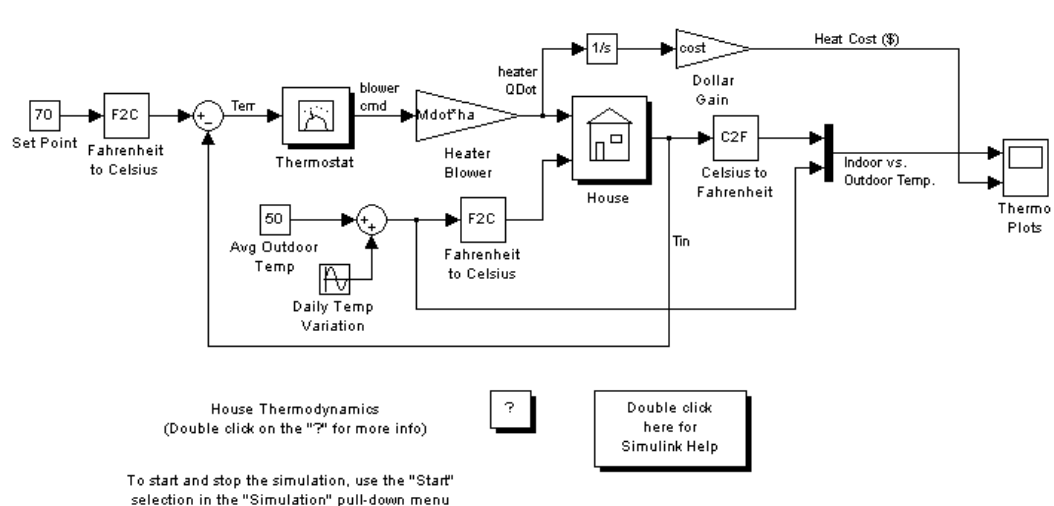
- izravan pristup podacima iz vanjskih jedinica i drugih softverskih paketa
- komunikacija i upravljanje uređajima za akviziciju podataka
- povezivanje s drugim programima (npr. MS Excel)

MATLAB Studentska verzija + MATLAB priručnik

- dostupni po povoljnoj cijeni za studente



- interaktivni sustav za *simuliranje* nelinearnih dinamičkih sustava (isporučuje se zajedno s MATLABom)
- *grafičko okruženje*, modeliranje i prikaz jednostavnim blok dijagramima
- linearni, nelinearni, vremenski kontinuirani, diskretni i viševeličinski sustavi
- vizualni prikaz rezultata simulacije, animacije i zapis u datoteke
- promjena simulacijskih parametara moguća i tijekom same simulacije



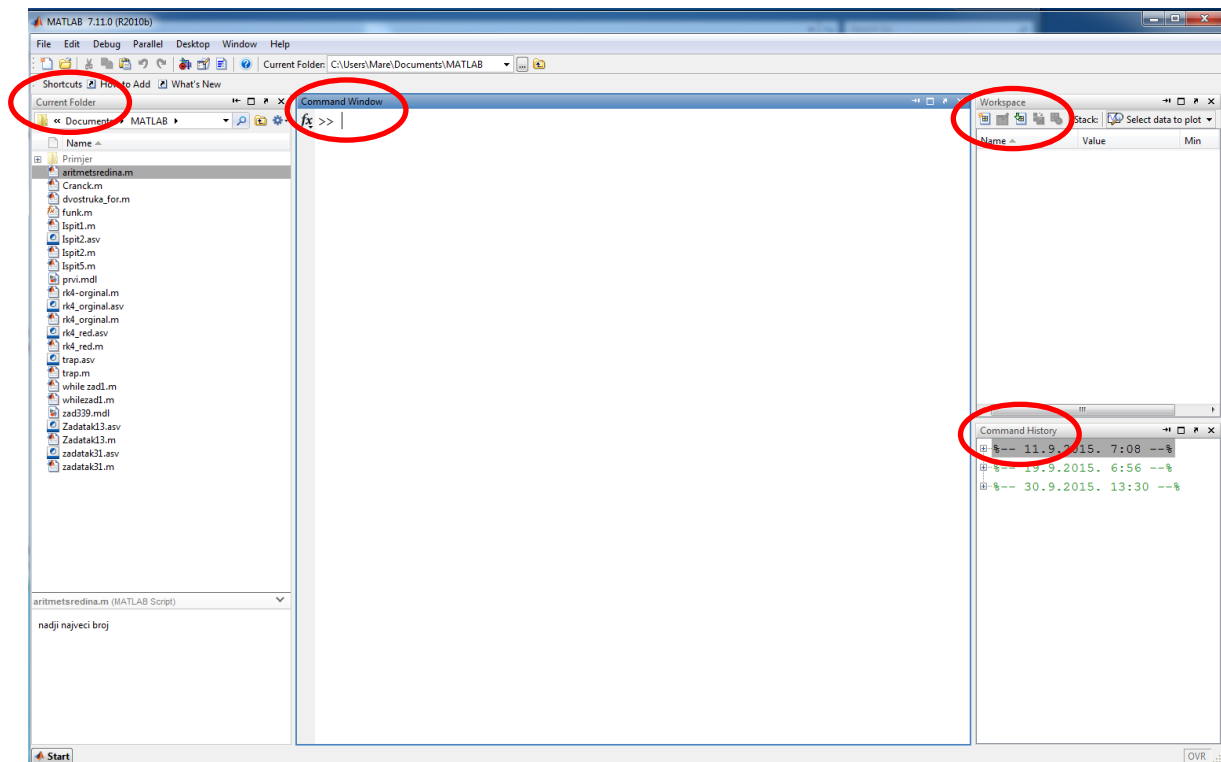
Blocksets

- ugrađeni dodaci Simulink-a koji podržavaju posebne primjene (npr. obrada digitalnih signala, posebne primjene itd.)

Real-Time Workshop

- generiranje C koda iz blok dijagrama i pokretanje aplikacija na različitim sustavima u realnom vremenu

Matlab sučelje



Varijable

Varijable su u osnovi memorijske lokacije čiji se sadržaj može mijenjati tokom izvođenja programa.

Varijable po **tipu** mogu biti cjelobrojne, decimalne, kompleksne, znakovne....

U Matlabu se znak “=” naziva operatorom pridruživanja koji pridružuje vrijednost varijabli.

Njegovo značenje je različito od znaka jednakosti u matematici. Lijevo od operatora pridruživanja može biti samo jedno ime varijable.

`var = 10;` Definiramo varijablu i pridružimo joj neku vrijednost

`var + 10`

`a = var + 1` Pridružujemo varijabli a vrijednost varijable var uvećanu za 1

`var = var + 1` Pridružujemo varijabli var vrijednost varijable var uvećanu za 1

Matrice

Unos matrica

Matrice se u MATLAB mogu unositi na nekoliko načina:

- direktni unos elemenata
- učitavanje matrica iz vanjske datoteke s podacima
- generiranje funkcija pomoću ugrađenih funkcija
- kreiranje matrica s vlastitim M-datotekama

Pri unosu matrice kao liste elemenata pridržavamo se slijedećih pravila:

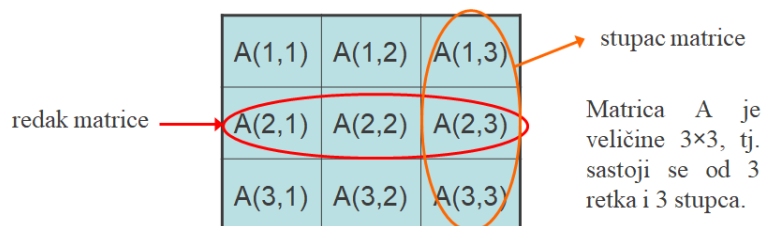
- odjeljujemo elemente reda s prazninama ili zarezom
- kraj svakog reda označavamo sa točkom-zarezom ;
- cijela lista elemenata nalazi se unutar uglatih zagrada, [] .

Za unos matrice, jednostavno upišemo:

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

MATLAB strukturirano prikazuje matricu A:

```
A =
    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1
```



Subscripts

Elementi reda i i stupca j matrice A označavaju se s $A(i, j)$. Npr., $A(4, 2)$ je broj u četvrtom redu i drugom stupcu. U našoj matrici $A(4, 2)$ je 15.

$A(4, 2) =$

15

Moguće je izračunati sumu elemenata četvrtog stupca od A pišući

$A(1, 4) + A(2, 4) + A(3, 4) + A(4, 4)$

što daje

```
ans =
    34
```

Moguće je i pozvati elemente matrice sa jednim indeksom, $A(k)$. To je uobičajeni način pozivanja vektora u obliku reda i stupca. Isto se može primijeniti na dvodimenzijску matricu, pri čemu se polje smatra jednim dugačkim stupčastim vektorom koji se sastoji od stupaca originalne matrice. Prema tome, $A(8)$ je drugi način pozivanja vrijednosti 15 smještene u $A(4, 2)$.

Ako se pokuša pozvati element koji je izvan dimenzija matrice, MATLAB javlja grešku.

Operator Colon

Jedan od najvažnijih MATLABovih operatora je dvotočka (engl. *colon*) :
Javlja se u nekoliko različitih oblika. Izraz

```
1:10
```

je vektor u obliku retka koji sadrži cijele brojeve od 1 do 10

```
1      2      3      4      5      6      7      8      9     10
```

Za dobivanje razmaka/koraka koji nije jediničan, navodi se korak. Npr.,

```
100:-7:50
```

daje

```
100     93     86     79     72     65     58     51
```

i

```
0:pi/4:pi
```

daje

```
0      0.7854      1.5708      2.3562      3.1416
```

Ako se dvotočka navodi u indeksu odnosi se na dio matrice

```
A(1:k, j)
```

je prvih k elemenata j tog stupca od A . Npr.,

```
A(1:3, 2)
```

Naredba

```
sum(A(1:4, 4))
```

računa sumu četvrtog stupca. Ali, postoji bolji način. Operator Colon sam po sebi poziva sve elemente u redu ili stupcu matrice, a ključna riječ *end* odnosi se na zadnji red ili stupac. Prema tome

```
sum(A(:,end))
```

računa sumu svih elemenata u zadnjem stupcu od A.

```
ans =
    34
```

Operatori

Oznaka	Operacija	engl.
+	Zbrajanje	Addition
-	Oduzimanje	Subtraction
*	Množenje	Multiplication
/	Dijeljenje	Division
\	Recipročno dijeljenje	Left division
^	Potencija	Power
'	Kompleksno konjugirano transponiranje	Complex conjugate transpose
()	Određivanje redoslijeda izvođenja operacija	Specify evaluation order

Elementarne matematičke funkcije

abs(x)	apsolutna vrijednost
exp(x)	e^x
log(x)	$\ln x$
log10(x)	$\log x$
rem(x,y)	ostatak (engl. <i>remainder</i>) nakon dijeljenja x/y
round(x)	zaokružuje na najbliži cijeli broj (integer)
sign(x)	daje predznak argumenta (vrijednosti -1, 0, 1)
sqrt(x)	kvadratni korijen
A^b	a^b
a:b	a/b ili b\a

Sintaksa

- varijable su "case sensitive", razlikuju se velika i mala slova, mogu imati do 19 karaktera i moraju započeti slovom
- tekst iza % uzima se kao komentar
- na kraju svakog reda stavlja se ";", ako želimo ispis stavlja se ", "
- ako je jednadžba preduga piše se "..." i pritisne *Enter*, a zatim se nastavlja pisati u novom redu

Ctrl + c kad se pritisne ova kombinacija tipaka prekida se izvođenja programa

Matrice

MATLAB ima četiri osnovne funkcije za stvaranje jednostavne matrice:

zeros	sve nule
ones	sve jedinice
rand	jednoliko raspoređeni slučajni elementi
randn	normalno distribuirani slučajni elementi

Nekoliko primjera:

```
Z = zeros(2,4)
```

```
Z =
    0    0    0    0
    0    0    0    0
```

```
F = 5*ones(3,3)
```

```
F =
    5    5    5
    5    5    5
    5    5    5
```

```
N = fix(10*rand(1,10))
```

```
N =
    4    9    4    4    8    5    2    6    8
0
```

```
R = randn(4,4)
```

```
R =
    1.0668    0.2944   -0.6918   -1.4410
    0.0593   -1.3362    0.8580    0.5711
   -0.0956    0.7143    1.2540   -0.3999
   -0.8323    1.6236   -1.5937    0.6900
```

Radni prostor

Radni prostor (*workspace*) je područje memorije dostupno iz MATLAB komandne linije. Naredbe `who` i `whos`, prikazuju trenutni sadržaj radnog prostora. Naredba `who` daje kratku listu, dok `whos` uz to daje veličinu i neke dodatne informacije.

Pogledajmo što daje naredba `whos` na radnom prostoru koji sadrži rezultate neke od primjera u ovom priručniku. Vidimo nekoliko različitih struktura podataka u MATLAB-u.

```
whos
```

Name	Size	Bytes	Class
A	4x4	128	double array
D	5x3	120	double array
M	10x1	3816	cell array

S	1x3	442	struct array
h	1x11	22	char array
n	1x1	8	double array
s	1x5	10	char array
v	2x5	20	char array

Grand total is 471 elements using 4566 bytes.

Da bi obrisali sve postojeće varijable iz radnog prostora unesite

```
clear
```

Save

Naredba `save` čuva sadržaj radnog prostora u MAT-datoteci koja se može pročitati naredbom `load` prilikom slijedećeg rada u MATLAB-u. Npr.

```
save primjer
```

snima cijeli sadržaj radnog prostora u datoteku `primjer.mat`.

Ako je potrebno, mogu se snimiti samo određene varijable navodeći njihova imena nakon imena datoteke. Varijable se spremaju u binarnom formatu koji MATLAB može čitati brzo. Ako je potrebno datotekama pristupiti izvan MATLAB, može se naznačiti neki drugi format.

<code>-ascii</code>	8-digit tekst format
<code>-ascii -double</code>	16-digit tekstalni format
<code>-ascii -double -tabs</code>	dijeli polje elemenata s tabovima
<code>-v4</code>	stvara datoteku za MATLAB 4 verziju
<code>-append</code>	dodaje podatke na već postojeći MAT-file.

`save ime_datoteke varijable -ascii -tabs` (koristi ASCII oblik s 8 znamenki)

`save ime_datoteke varijable -ascii -double -tabs` (koristi ASCII oblik s 16 znamenki)

Load

`Load` naredba čita binarnu datoteku koja sadrži matrice stvorene pri prethodnom radu s MATLABom pomoću naredbe `Save` ili čita tekst datoteku koja sadrži numeričke podatke. Tekst datoteka treba biti organizirana kao pravokutna tablica brojeva, odvojena prazninama s jedni redom u liniji i jednakim brojem elemenata u svakom redu. Ovako izgleda tekst datoteka stvorena izvan MATLABa:

```
16.0  3.0  2.0  13.0
 5.0 10.0 11.0  8.0
 9.0  6.0  7.0 12.0
 4.0 15.0 14.0  1.0
```


Spremanjem datoteke pod imenom npr.

```
save magik.dat
```

Zatim naredbom

```
load magik.dat
```

čita se datoteka i stvara varijabla `magik` koja je ustvari gore napisana matrica.

Script File ili M-file

To su jednostavne tekst datoteke koje sadrže MATLAB naredbe, koje se mogu lako otvarati i izvoditi, a pri čemu MATLAB slijedi niz naredbi u datoteci. Izraz *M-file* kaže nam da imena script datoteka moraju završavati s ekstenzijom ".m", npr. `primjer.m`

Za stvaranje M-datoteke otvara se *New* iz izbornika *File* i odabire se *M-file*. Na taj način otvara se MATLABov tekst editor u kojeg ćemo upisivati naredbe.

Nakon što takvu datoteku spremimo na disk, jednostavno je pokrećemo utipkavanjem njenog imena u MATLAB promptu, ali pri tom moramo biti u direktoriju gdje je datoteka pohranjena:

```
» primjer
```

Sve varijable iz M-datoteke postaju dio radnog prostora MATLABa.

Naredbe MATLABa za rad s datotekama:

Naredba	Opis
<code>what</code>	popis M-file-ova u direktoriju
<code>dir , ls</code>	popis svih datoteka u direktoriju
<code>delete test</code>	briše M-file <code>test.m</code>
<code>cd path</code>	prelazi u zadani direktorij
<code>cd , pwd</code>	prikazuje trenutne direktorije
<code>which test</code>	prikazuje put do datoteke <code>test.m</code>

Primjer, kreirajte matricu A koja sadrži ova četiri redka (studenti)

```
A =
    16.0    3.0    2.0   13.0
     5.0   10.0   11.0    8.0
     9.0    6.0    7.0   12.0
     4.0   15.0   14.0    1.0
```

Pohranite datoteku pod imenom `magik.m`.

Nakon toga naredbom `magik` učitajte datoteku s varijablom A koja sadrži ovu matricu.

Jednostavna polja

Unos polja ima slijedeći oblik:

```
varijabla=[ niz brojeva odvojenih zarezom ili razmakom ]
```

npr.

```
A = [ 6 4 6 3; 5 sqrt(2) 4.2 5; 5 sin(0.5) 4 3; 4 15 14 1.3 ]
```

Ovaj izraz predstavlja matricu 4×4, pojedini redovi međusobno su odvojeni točkazarezom.

Funkcije polja primjenjuju funkciju na svaki pojedini element polja.

Polja koja sadrže linearno razmaknute elemente mogu se formirati na slijedeći način:

first:increment:last
(prvi član: prirast : zadnji član)

a može i pomoću funkcije

```
linspace(first, last, number_of_points)
```

Polja koja sadrže logaritamski razmaknute elemente mogu biti formirana uporabom funkcije

```
logspace(first_exp, last_exp, number_of_points)
```

Osnovne matematičke operacije između skalara i polja primjenjuju se na sve elemente u polju, npr.

$x-2$

oduzima dvojku od svih elemenata u x .

Osnovne matematičke operacije između polja vrijedi ako su polja iste veličine, množenje i dijeljenje simbolički se prikazuje kao $.*$ i $./$ ili $.\$

Polja mogu imati bilo koju pravokutnu formu. Polje s jednim redom obično se naziva retčasti vektor. Polje s jednim stupcem naziva se stupčasti vektor. Polje s više redova i stupaca nazivaju se matrice. Retčasti vektor može se pretvoriti u stupčasti i obratno uporabom operacije točka-apostrof, npr. ako je x red, $x.'$ je stupac.

Točka-apostrof označava transponiranje. Samo apostrof označava kompleksno konjugirano transponiranje.

Relacijski i logički operatori

Rezultat ovih izraza je 1 za istinu i 0 za neistinu.

Relacijski operator	Opis
<	manje nego - less than
<=	manje nego ili jednako - less than or equal to
>	veće nego - greater then
>=	veće nego ili jednako - greater than or equal to
==	jednako - equal to
~=	nije jednako, različito - not equal to

Logički operator	Opis
&	I AND
	ILI OR
~	NE NOT

Petlje i kontrola toka

Struktura (Control Flow)	Opis
for n= početak:kraj naredba end	pri svakoj iteraciji n dodjeljuje i-tom stupcu polja i izvodi naredbe
while izraz naredbe end	izvodi naredbe sve dok svi elementi izraza nisu istina ili različiti od nule
if izraz naredbe ako je izraz istinit elseif izraz naredbe ako je izraz istinit else naredbe ako su izrazi neistiniti end	izvodi se tako da se jedna grupa naredbi izvodi ako je izraz istinit ili različit od nule, druga grupa naredbi izvodi se ako je izraz neistinit ili nula
break	prekida izvođenje for i while petlja

Važno je shvatiti kako relacijski operatori i `if` naredba rade s matricama. Kad se želi provjeriti jednakost dviju varijabli, može se uzeti

```
if A == B, ...
```

Ovo je MATLAB kod, i radi ono što se i očekuje kad su A i B skalari. Ali kad su A i B matrice, `A == B` ne testira *jesu li* oni jednaki, nego provjerava *gdje* su jednaki; rezultat je nova matrica gdje 0 i 1 prikazuje jednakost element po element. Zapravo, ako A i B nisu istih dimenzija, onda `A == B` javlja pogrešku.

For

For petlja ponavlja grupu izraza unaprijed određen broj puta. Izraz završava s odgovarajućom end naredbom.

```
for n = 3:32
    r(n) = rank(magic(n));
end
```

Naredbe između for i end izvode se jednom za svaki stupac u polju. Pri svakoj iteraciji n se dodjeljuje slijedećem stupcu polja, tj. za vrijeme i-tog prolaza kroz petlju, npr.

```
for n=1:10
    x(n)=sin(n*pi/10);
end
```

Petlje je dobro pisati uvučeno radi čitljivosti, posebno kad ih ima više.

```
for i = 1:m
    for j = 1:n
        H(i,j) = 1/(i+j);
    end
end
```

While

Petlja while ponavlja grupu naredbi ovisno o logičkom uvjetu. Petlja završava odgovarajućom end naredbom.

Ovdje je prikazan program koji prikazuje upotrebu naredbi while, if, else, i end, prilikom traženja nule polinoma pomoću metoda bisekcija.

```
a = 0; fa = -Inf;
b = 3; fb = Inf;
while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if sign(fx) == sign(fa)
        a = x; fa = fx;
    else
        b = x; fb = fx;
    end
end
x
```

Rezultat je korijen polinoma $x^3 - 2x - 5$, naime

```
x =
    2.09455148154233
```

Break

Izraz `break` omogućava raniji izlaz iz `for` ili `while` petlje. U višestrukim petljama `break` izlazi samo iz unutarnje petlje.

M-File funkcije

Pošto su funkcije u MATLAB-u vrlo moćan alat pri proračunu kompleksnih problema, postoji mogućnost kreiranja vlastitih funkcija u formi M-datoteka pohranjenih na računalu.

M-file funkcija je slična običnom M-file-u, tekst datoteci s ekstenzijom **.m**

```
function y= ime_funkcije(x1,x2, ...,xn)  
    y= ...
```

Ime funkcije i ime datoteke moraju biti identični. Npr., funkcija `primjer` mora biti pohranjena u datoteci `primjer.m`

MATLAB traži M-file funkcije na isti način kao što radi i kod običnih M-datoteka. Npr, ako upišemo `primjer` u promptu, prvo traži varijablu s tim imenom. Ako je nema, onda je traži kao ugrađenu funkciju. Na kraju provjerava u trenutnom direktoriju postoji li `primjer.m`. Ako se ne nalazi ni ovdje provjerava sve MATLAB direktorije da bi našao `primjer.m`

Format

Naredba `Format` upravlja s numeričkim formatom vrijednosti prikazanih u MATLAB-u. Naredba djeluje samo na prikaz brojeva, a ne i kako će MATLAB računati i spremati ih.

Pogledajmo nekoliko različitih formata, zajedno s izlazom dobivenim za vektor `x` čije su komponente različite veličine

```
x = [4/3 1.2345e-6]  
  
format short  
  
    1.3333    0.0000  
  
format short e  
  
    1.3333e+000    1.2345e-006  
  
format short g  
  
    1.3333    1.2345e-006  
  
format long  
  
    1.333333333333333    0.00000123450000
```

```
format long e
    1.3333333333333333e+000    1.2345000000000000e-006

format long g
    1.333333333333333    1.2345e-006

format bank
    1.33    0.00

format rat
    4/3    1/810045

format hex
    3ff5555555555555    3eb4b6231abfd271
```

Slijedeća naredba potiskuje prazne linije koje se javljaju pri ispisu i tako omogućuje prikaz više informacija na ekranu.

```
format compact
```

Crtanje i grafički prikazi

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

Naredba `plot` automatski određuje granice osi, označuje pojedine točke i povezuje ih ravnom linijom. Različite opcije naredbe `plot` omogućavaju crtanje više funkcija ili niza podataka na istom grafu, uporabu različitih tipova i boja krivulja, mogućnost označavanja samo točaka bez linija. Osi se mogu imenovati, staviti naslov iznad slike, stavljati crtice na osi itd.

npr.

```
plot(x,y,x,z) - crtanje dviju funkcija y i z u ovisnosti od x na istom grafu
```

```
plot(x,y,x,y, '+') - crta funkciju y dva puta jednom linije, a drugi puta "+"
```

```
grid - crtanje mreže isprekidanih linija unutar grafa
```

```
xlabel('tekst ispod osi x')
```

```
ylabel('tekst ispod osi y')
```

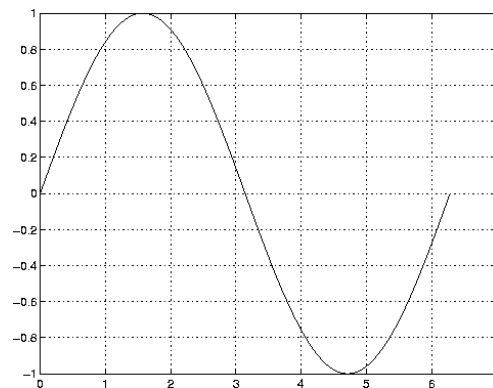
```
title('naslov iznad grafa')
```

Crtanje grafova

Funkcija `plot` ima različite oblike, što ovise o ulaznom argumentu. Ako je `y` vektor `plot(y)` daje linearni graf `y` u ovisnosti o indeksu elementa `y`, Ako se postavie dva vektora kao argumenti, `plot(x, y)` crta graf `y` vs. `x`.

Da bi nacrtali funkciju sinus na segmentu od 0 do 2. Napisat ćemo:

```
t = 0:pi/100:2*pi;
y = sin(t);
plot(t,y)
```

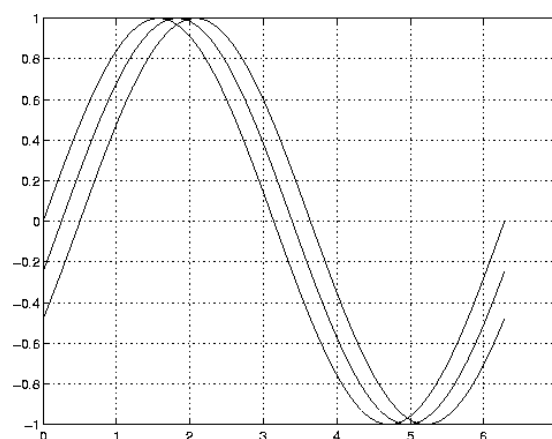


Izrazi koji omogućuju crtanje tri funkcije od `t`, pri čemu je svaka krivulja drugačije boje:

```
y2 = sin(t-.25);
y3 = sin(t-.5);
plot(t,y,t,y2,t,y3)
```

Moguće je odrediti boju, stil linije i markere (npr. kružići ili križići) naredbom

```
plot(x,y,'color_style_marker')
```



`color_style_marker` je 1-, 2- ili 3-znakovni karakter (odvojen s jednostrukim navodnicima) koji se sastoji od boje, stila linije i vrste markera:

* Color = 'c', 'm', 'y', 'r', 'g', 'b', 'w', and 'k'.

(cyan, magenta, yellow, red, green, blue, white, and black)

* Linestyle = '-' puna linija, '- -' isprekidana, ':' točkasta, '- .' točkacrtta, i 'none' bez crtanja linije

* Uobičajeni markeri su '+', 'o', '*', i 'x'.

Npr., naredba

```
plot(x, y, 'y:+')
```

crta žutu točkastu liniju i svaku točku označava križićem, ako je definirani marker a nije linija MATLAB crta samo markere.

Grafički prozor

Funkcija plot automatski otvara novi prozor s grafičkim prikazom ako već ne postoji otvoreni prozori na ekranu. Da bi otvorili novi prozor za grafički prikaz i da bi postao tekućim:

```
figure
```

Da bi učinili postojeći prozor s grafičkim prikazom tekućim:

```
figure(n)
```

pri čemu je n broj slika na traci s naslovom.

Dodavanje crteža na postojeći grafički prikaz

Naredba hold daje mogućnost dodavanja crteže na postojeći grafički prikaz:

```
hold on
```

MATLAB ne uklanja postojeći grafički prikaz, nego dodaje nove podatke na aktivni. Npr. slijedeći niz naredbi prvo stvara konturni prikaz peaks funkcije, a zatim dodaje pseudoobojan prikaz iste funkcije:

```
[x, y, z] = peaks;  
contour(x, y, z, 20, 'k')  
hold on  
pcolor(x, y, z)  
shading interp
```

Naredba hold on omogućuje da se pcolor prikaz kombinira s contour prikazom na jednoj slici.

Crtanje 3D grafova

U sljedećem primjeru funkcija `peaks` će biti prikazana najčešćim naredbama za trodimenzionalno crtanje MATLABu.

```
[x, y, z] = peaks;  
plot3(x, y, z)
```

Naredba `plot3` u suštini radi isto što i `plot` funkcija – spaja setove koordinata linijama, ali u ovome slučaju koordinate u tri smijera x , y i z . Kao i kod `plot` funkcije bitno je da su sve tri varijable vektori iste dužine.

```
scatter3(x, y, z)
```

Analogno dvodimenzionalnoj inačici, `scatter3` crta točke na specificiranim lokacijama zadanih vektorima x , y i z .

```
mesh(x, y, z)
```

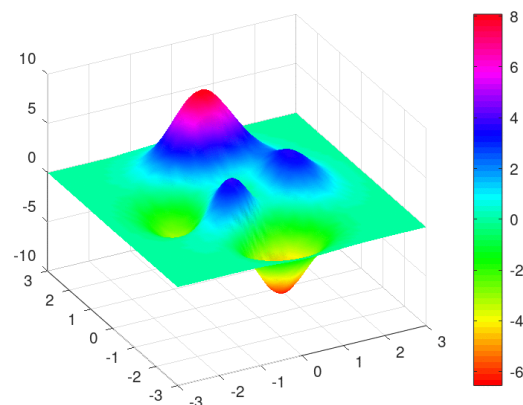
Naredba `mesh` stvara mrežu između točaka tako linijama stvarajući površine - četverokute koji svojim oblikom nagovještavaju trodimenzionalni smijer kretanja funkcije.

```
surf(x, y, z)
```

Naredba `surf` nadogradnja je na prikazanu naredbu `mesh`, gdje su površine obojane te intuitivno prikazuju vrijednosti veće i manje od nule.

Moguće je također dodati drugačiju skalu boja naredbom `colormap`, te odnos broјčane vrijednosti s bojama naredbom `colorbar`. Radi lakšeg prikaza mogu se i maknuti linije naredbom `shading interp`:

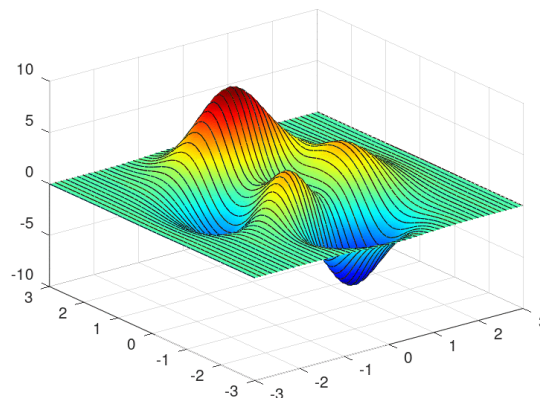
```
[x, y, z] = peaks;  
surf(x, y, z)  
colormap hsv  
colorbar  
shading interp
```



Neke od opcija colorbar naredbe: jet, winter, summer, hsv, turbo, lines, colorcube.

Za 3D crteže također vrijede navedena pravila dodavanja crteža na postojeći graf.

```
[x,y,z] = peaks;
surf(x,y,z)
colormap jet
shading interp
hold on
plot3(x,y,z, 'k')
```



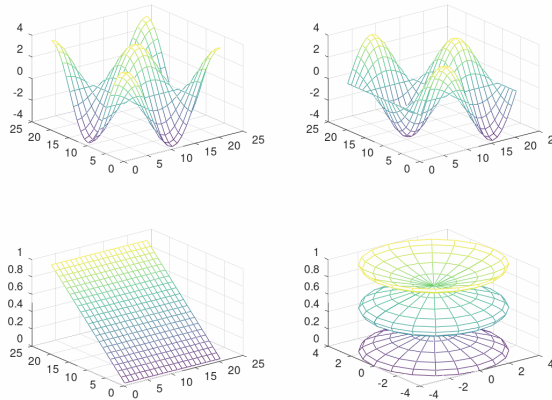
Subplot

Funkcija subplot omogućuje prikaz više grafova u istom prozoru ili tiskanje takvih grafova na jednom papiru:

```
subplot(m,n,p)
```

preloma sliku u $m \times n$ matricu manjih grafova, a p se navodi kao aktivni graf. Grafovi se broje duž prvog reda u prozoru, zatim drugi red itd. Slijedeći primjer pokazuje kako bi nacrtali podatke na četiri grafa unutar prozora:

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1)
mesh(X)
subplot(2,2,2); mesh(Y)
subplot(2,2,3); mesh(Z)
subplot(2,2,4); mesh(X,Y,Z)
```



Još neke mogućnosti:

```
subplot (1,1,1)
```

briše sve osi i vraća na početni postav sa jednim oknom

```
pause
```

privremeno prekida izvođenje programa (pritiskom na bilo koju tipku izvođenje se nastavlja, dok

```
pause (n)
```

zaustavlja prikaz na n sekundi pa potom nastavlja.

Kontrola osi

Funkcija `axis` ima brojne opcije za podešavanje skaliranja, orijentacije i aspektni odnos crteža.

MATLAB obično traži maksimume i minimume danih podataka i prema tome bira odgovarajuće područje na osima. Funkcijom `axis` sami ugađamo granice osi:

```
axis([xmin xmax ymin ymax])
```

`axis` isto tako prihvaća ključne riječi za kontrolu osi, npr.

```
axis square
```

postavlja iste duljine za čitave x i y -osi, a

```
axis equal
```

postavlja iste duljine crtica i na x - i y -osi.

Izraz `grid off` isključuje grid lines, dok ih

`grid on` ponovno uključuje.