Prirodoslovno-matematički fakultet
Matematički odsjek
Sveučilište u Zagrebu

ITERATIVNE METODE

Vježbe 01 - Matlab/Octave

1. kolovoza 2021.

Sastavio: Zvonimir Bujanović

Sadržaj

UVOD

Matlab / Octave

Matlab (MATrix LABoratory) je programsko okruženje za tehničko i znanstveno računanje.

GNU Octave je besplatni, open-source klon Matlaba.

Omogućavaju:

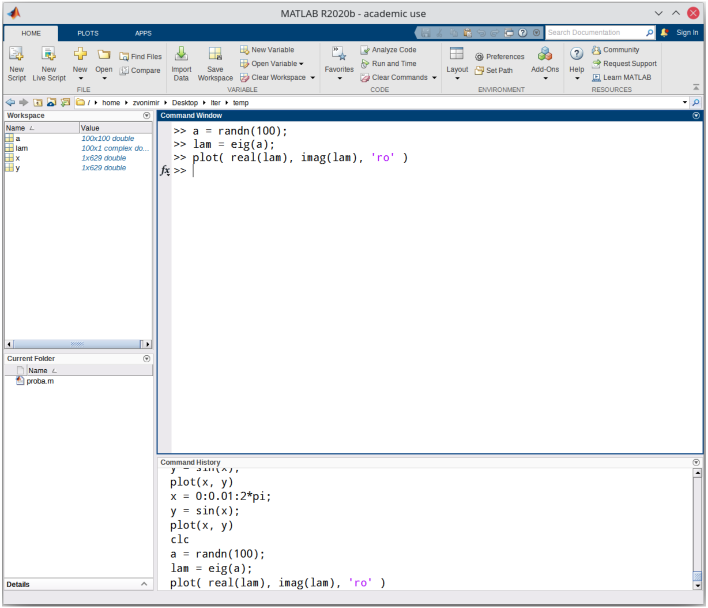
- izvođenje kompleksnih proračuna
- vizualizaciju rezultata
- izvođenje simulacija
- programiranje

Matlab ima vlastiti (interpretirani) programski jezik visokog nivoa koji je jednostavan za korištenje i upotrebljava standardiziranu matematičku sintaksu. Podržava i objektno-orijentirani pristup.

Brojnim *toolboxovima* moguće je jako proširiti funkcionalnost.

RAD U INTERAKTIVNOJ KONZOLI

Korisničko sučelje



Korisničko sučelje

Dijelovi korisničkog sučelja:

- *Command Window* – interaktivna konzola u koju utipkavamo naredbe i vidimo njihov rezultat

Korisničko sučelje

Dijelovi korisničkog sučelja:

- *Command Window* – interaktivna konzola u koju utipkavamo naredbe i vidimo njihov rezultat
- *Workspace* – popis svih varijabli i njihova osnovna svojstva

Korisničko sučelje

Dijelovi korisničkog sučelja:

- *Command Window* – interaktivna konzola u koju utipkavamo naredbe i vidimo njihov rezultat
- *Workspace* – popis svih varijabli i njihova osnovna svojstva
- *Command History* – popis ranije izvršenih naredbi

Korisničko sučelje

Dijelovi korisničkog sučelja:

- *Command Window* – interaktivna konzola u koju utipkavamo naredbe i vidimo njihov rezultat
- *Workspace* – popis svih varijabli i njihova osnovna svojstva
- *Command History* – popis ranije izvršenih naredbi
- *Current Folder* – trenutni direktorij u kojem se nalaze programi dostupni za izvršavanje, vidi ovdje za detalje.

Help

Matlab dolazi sa vrlo iscrpnim sustavom pomoći, pod menijem Help $\rightarrow$ Documentation.

Osim toga, pomoć je dostupna u interaktivnoj konzoli:

```
help eig; % kratki info o naredbi eig, ispiše se u konzoli
doc eig;  % detaljna dokumentacija o naredbi eig
```

Sve dokumentacija je dostupna i online.

Unošenje naredbi

Svaka naredba u Matlabu može i ne mora završavati točka-zarezom.

Naredbe koje ne završavaju točka-zarezom rezultiraju ispisom izračunate vrijednosti na ekran:

```
a = sqrt( -1 )
```

```
a =
    0 + 1.0000i
```

Rezultat naredbi koje završavaju točka-zarezom neće biti ispisan:

```
a = sqrt( -1 );
```

```
|
```

Ekran možemo obrisati naredbom `clc;`

Unos elementarnih varijabli

- `0.314` ili `3.14e-1` – realni broj (`double`)
- `-3+0.5i` – kompleksni broj
- `'x'` – `char`
- `'iter'` – string (1D polje charova)
- `true`, `false` – logička vrijednost (`logical`)

Načini ispisa decimalnih brojeva

Matlab može ispisivati realne brojeve u 4 osnovna formata. Ako utipkamo sljedeće naredbe i ispišemo vrijednost od `a=31.4`, ispisi su:

```
format short
```

```
31.400
```

```
format long
```

```
31.40000000000000
```

```
format short e
```

```
3.1400e+001
```

```
format long e
```

```
3.140000000000000e+001
```

TIPOVI PODATAKA U MATLABU I VARIJABLE

Tipovi podataka

Matlab je *slabo tipiziran* jezik. Nije potrebno deklarirati varijable; pojedine varijable mogu mijenjati svoj tip “u letu”:

```
a = [1 2 3]
a = 'pero'
a.element = 123
```

```
a =
    1     2     3

a =
    pero

a =
    element: 123
```

Tipovi podataka

Bazični tipovi podataka u Matlabu su npr. `single`, `double`, `char`, `logical`, `int32` ...

Ali svaka varijabla je zapravo 2D-matrica nekog od bazičnih tipova.

U Matlabu osim matrica postoje i sljedeći tipovi podataka:

- 3 i više dimenzionalna polja
- Strukture – kao `struct` u C-u.
- Cell-array – 1 ili više dimenzionalno polje elemenata koji mogu biti različitih tipova.
- Objekti – slično kao u C++ (strukture čiji članovi mogu biti i funkcije).

Varijable

Tip varijable možemo ispitati pomoću `is????` funkcija.

```
a = 3 + 2i;
isnumeric( a )
ischar( a )
islogical( a )
isstruct( a )
iscell( a )
isreal( a )
```

Tip varijable `x` možemo ispisati pomoću naredbe `whos x`.
Tipove svih deklariranih varijabli doznajemo pomoću `whos`.

Varijablu `x` možemo "oddeklarirati" pomoću `clear x`.
Sve varijable možemo "oddeklarirati" pomoću `clear`.

Varijable

Deklarirane varijable možemo spremiti u binarnu mat datoteku i kasnije ih ponovno učitati.

Varijable `x`, `yy`, `ZZZ` spremamo u datoteku `var.mat` pomoću

```
save( 'var.mat', 'x', 'yy', 'ZZZ' );
```

Sve deklarirane varijable spremamo u datoteku `var.mat` pomoću

```
save( 'var.mat' );
```

Varijable `x`, `yy`, `ZZZ` učitavamo iz datoteke `var.mat` pomoću

```
load( 'var.mat', 'x', 'yy', 'ZZZ' );
```

Sve varijable koje postoje u datoteci `var.mat` učitavamo sa

```
load( 'var.mat' );
```

MATRICE

Matrice možemo zadati ovako:

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

```
A =
      1      2      3
      4      5      6
```

Možemo posložiti druge vektore i/ili matrice u novu matricu:

```
X = [1 2; 3 4]; Y = [7; 8];
A = [X Y]
B = [X; 7 8]
```

```
A =
      1      2      7
      3      4      8
B =
      1      2
      3      4
      7      8
```

Zadavanje matrica

Postoje brze naredbe za stvaranje nul-matrice (`zeros`), jedinične matrice (`eye`), matrice pune jedinica (`ones`), random matrice sa uniformnom (`rand`) i normalnom (`randn`) distribucijom elemenata:

```
A = zeros( 5 );
B = ones( 3, 4 );
C = eye( 5 );
D = eye( 3, 6 );
E = rand( 6, 3 );
F = randn( 1 );
```

Zadavanje matrica

Postoje i brze naredbe za zadavanje matrica tipa $1 \times n$:

```
3:7
```

```
ans =
    3      4      5      6      7
```

```
3:6:30
```

```
ans =
    3      9     15     21     27
```

```
1.2 : 3.4 : 12.7
```

```
ans =
    1.2000    4.6000    8.0000   11.4000
```

Pristup elementima matrice

- `A(r, s)`
  Element u retku *r* i stupcu *s*.

Pristup elementima matrice

- `A(r, s)`
  Element u retku *r* i stupcu *s*.

- `A(r1:r2, s1:s2)`
  Podmatrica koja uključuje retke od r1 do r2 i stupce od s1 do s2.

Pristup elementima matrice

- `A(r, s)`
  Element u retku *r* i stupcu *s*.

- `A(r1:r2, s1:s2)`
  Podmatrica koja uključuje retke od r1 do r2 i stupce od s1 do s2.

- `r=[3 5]; s=[7 2]; A(r,s)`
  Podmatrica koja sadrži retke 3 i 5 i stupce 7 i 2.

Pristup elementima matrice

- `A(r, s)`
  Element u retku *r* i stupcu *s*.

- `A(r1:r2, s1:s2)`
  Podmatrica koja uključuje retke od r1 do r2 i stupce od s1 do s2.

- `r=[3 5]; s=[7 2]; A(r,s)`
  Podmatrica koja sadrži retke 3 i 5 i stupce 7 i 2.

- `A(:,1:3:8)`
  Podmatrica koja sadrži sve retke i stupce 1, 4 i 7.

Aritmetički operatori

- `A+B`, `A-B`, `A*B` – zbrajanje/oduzimanje/množenje matrica ispravnih dimenzija; oprez: ponašanje je drugačije ako je jedan od operanada skalar!

Aritmetički operatori

- `A+B`, `A-B`, `A*B` – zbrajanje/oduzimanje/množenje matrica ispravnih dimenzija; oprez: ponašanje je drugačije ako je jedan od operanada skalar!

- `A.*B` – množenje element po element, `A(r,s)*B(r,s)`, matrice `A` i `B` su istih dimenzija

Aritmetički operatori

- `A+B`, `A-B`, `A*B` – zbrajanje/oduzimanje/množenje matrica ispravnih dimenzija; oprez: ponašanje je drugačije ako je jedan od operanada skalar!

- `A.*B` – množenje element po element, `A(r,s)*B(r,s)`, matrice `A` i `B` su istih dimenzija

- `A'` – kompleksno konjugirana i transponirana matrica

Aritmetički operatori

- `A+B`, `A-B`, `A*B` – zbrajanje/oduzimanje/množenje matrica ispravnih dimenzija; oprez: ponašanje je drugačije ako je jedan od operanada skalar!

- `A.*B` – množenje element po element, `A(r,s)*B(r,s)`, matrice `A` i `B` su istih dimenzija

- `A'` – kompleksno konjugirana i transponirana matrica
- `A.'` – transponirana matrica (bez konjugiranja)

Aritmetički operatori

- `A+B`, `A-B`, `A*B` – zbrajanje/oduzimanje/množenje matrica ispravnih dimenzija; oprez: ponašanje je drugačije ako je jedan od operanada skalar!

- `A.*B` – množenje element po element, `A(r,s)*B(r,s)`, matrice `A` i `B` su istih dimenzija

- `A'` – kompleksno konjugirana i transponirana matrica
- `A.'` – transponirana matrica (bez konjugiranja)

- `A^p` – matrično potenciranje

Aritmetički operatori

- `A+B`, `A-B`, `A*B` – zbrajanje/oduzimanje/množenje matrica ispravnih dimenzija; oprez: ponašanje je drugačije ako je jedan od operanada skalar!

- `A.*B` – množenje element po element, `A(r,s)*B(r,s)`, matrice `A` i `B` su istih dimenzija

- `A'` – kompleksno konjugirana i transponirana matrica
- `A.'` – transponirana matrica (bez konjugiranja)

- `A^p` – matrično potenciranje
- `A.^p` – potenciranje element po element, `A(r,s)^p`, $p \in \mathbb{R}$

Aritmetički operatori

- `A+B`, `A-B`, `A*B` – zbrajanje/oduzimanje/množenje matrica ispravnih dimenzija; oprez: ponašanje je drugačije ako je jedan od operanada skalar!

- `A.*B` – množenje element po element, `A(r,s)*B(r,s)`, matrice `A` i `B` su istih dimenzija

- `A'` – kompleksno konjugirana i transponirana matrica
- `A.'` – transponirana matrica (bez konjugiranja)

- `A^p` – matrično potenciranje
- `A.^p` – potenciranje element po element, `A(r,s)^p`, $p \in \mathbb{R}$

- `A.^B` – potenciranje element po element, `A(r,s)^B(r,s)`, matrice `A` i `B` su istih dimenzija

Aritmetički operatori

- `A\b` – matrično lijevo dijeljenje.
  - `A` regularna i kvadratna
    $\rightsquigarrow$ `x=A\b` je rješenje sustava $Ax = b$.

  - `A` nije regularna ili pravokutna
    $\rightsquigarrow$ `x=A\b` je rješenje problema najmanjih kvadrata $\min_x \|Ax - b\|$.

Ovo radi i za matrice: `A\B` računa $A^{-1}B$.

Aritmetički operatori

- `A\b` – matrično lijevo dijeljenje.
  - `A` regularna i kvadratna
    $\rightsquigarrow$ `x=A\b` je rješenje sustava $Ax = b$.
  - `A` nije regularna ili pravokutna
    $\rightsquigarrow$ `x=A\b` je rješenje problema najmanjih kvadrata $\min_x \|Ax - b\|$.

  Ovo radi i za matrice: `A\B` računa $A^{-1}B$.

- `A.\B` – *lijevo* dijeljenje po elementima: `B(i,j)/A(i,j)`

Aritmetički operatori

- `A\b` – matrično lijevo dijeljenje.
  - `A` regularna i kvadratna
    $\rightsquigarrow$ `x=A\b` je rješenje sustava $Ax = b$.

  - `A` nije regularna ili pravokutna
    $\rightsquigarrow$ `x=A\b` je rješenje problema najmanjih kvadrata $\min_x \|Ax - b\|$.

  Ovo radi i za matrice: `A\B` računa $A^{-1}B$.

- `A.\B` – *lijevo* dijeljenje po elementima: `B(i,j)/A(i,j)`
- `A/B` – matrično *desno* dijeljenje, isto kao `(B'\A')'`; računa AB^{-1}

Aritmetički operatori

- `A\b` – matrično lijevo dijeljenje.
  - `A` regularna i kvadratna
    $\rightsquigarrow$ `x=A\b` je rješenje sustava $Ax = b$.
  - `A` nije regularna ili pravokutna
    $\rightsquigarrow$ `x=A\b` je rješenje problema najmanjih kvadrata $\min_x \|Ax - b\|$.

  Ovo radi i za matrice: `A\B` računa $A^{-1}B$.

- `A.\B` – *lijevo* dijeljenje po elementima: `B(i,j)/A(i,j)`
- `A/B` – matrično *desno* dijeljenje, isto kao `(B'\A')'`; računa AB^{-1}
- `A./B` – *desno* dijeljenje po elementima: `A(i,j)/B(i,j)`

Aritmetički operatori

- `A\b` – matrično lijevo dijeljenje.
  - `A` regularna i kvadratna
    $\rightsquigarrow$ `x=A\b` je rješenje sustava $Ax = b$.
  - `A` nije regularna ili pravokutna
    $\rightsquigarrow$ `x=A\b` je rješenje problema najmanjih kvadrata $\min_x \|Ax - b\|$.

  Ovo radi i za matrice: `A\B` računa $A^{-1}B$.

- `A.\B` – *lijevo* dijeljenje po elementima: `B(i,j)/A(i,j)`
- `A/B` – matrično *desno* dijeljenje, isto kao `(B'\A')'`; računa AB^{-1}
- `A./B` – *desno* dijeljenje po elementima: `A(i,j)/B(i,j)`

Dakle:

- `X=A\B` $\rightsquigarrow X = A^{-1}B$, odnosno $AX = B$
- `X=A/B` $\rightsquigarrow X = AB^{-1}$, odnosno $XB = A$

Logički i relacijski operatori, funkcije

Svi logički i relacijski operatori i navedene funkcije se izvrednjavaju po elementima.

- `~A` (logičko NE), `A&B` (logički I), `A|B` (logički ILI)
- `A<B`, `A<=B`, `A>B`, `A>=B`, `A==B`, `A~=B`
- `sin`, `cos`, `asin`, `acos`, `sinh`, `cosh`, `asinh`, `acosh`, `exp`, `log`, `log10`, `sqrt`, `abs`, `round`

Na skalarnim (1×1) podacima postoje i bitovni operatori `&&`, `||` koji imaju *lijeno* izvrednjavanje; korisno kod ▸ if-ova.

Specijalne matrične funkcije

Neka je

```
A = [1 2 3 4; 5 6 7 8; 9 10 11 12; ...
     13 14 15 16; 17 18 19 20]
```

```
A =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
    17    18    19    20
```

Na sljedećim stranicama ispisujemo što razne funkcije vraćaju za ovu matricu.

Specijalne matrične funkcije

`size( A )` – vektor-redak sa dimenzijama matrice

```
5 4
```

`length( A )` – najveća od dimenzija matrice (korisno za vektore)

```
5
```

`min( A )` – najmanji elementi u svakom stupcu. Ako je `A` vektor-redak, onda vraća najmanji element. Može vratiti i index.

```
1 2 3 4
```

`max( A )` – najveći elementi u svakom stupcu. Ako je `A` vektor-redak, onda vraća najveći element. Može vratiti i index.

```
17 18 19 20
```

Specijalne matrične funkcije

`diag( A )` – vektor-stupac s dijagonalom matrice

```
1
6
11
16
```

`diag(diag(A))` – ako je input za `diag` vektor, vraća dijagonalnu matricu

```
1     0     0     0
0     6     0     0
0     0    11     0
0     0     0    16
```

`sort([7 3 8 11])` – Sortira uzlazno vektor-redak. Sortira uzlazno svaki stupac matrice. Može vratiti i permutaciju.

```
3 7 8 11
```

Specijalne matrične funkcije

`triu( A )` – gornje-trokutasti dio od `A`

```
1      2      3      4
0      6      7      8
0      0     11     12
0      0      0     16
0      0      0      0
```

`tril( A )` – donje-trokutasti dio od `A`

```
 1      0      0      0
 5      6      0      0
 9     10     11      0
13     14     15     16
17     18     19     20
```

- `norm( A, 1 )`, `norm( A )`, `norm( A, inf )`, `norm( A, 'fro' )`
  – 1 / 2 / ∞ / Frobeniusova-norma matrice `A`

Funkcije numeričke linearne algebre

- `norm( A, 1 )`, `norm( A )`, `norm( A, inf )`, `norm( A, 'fro' )` – 1 / 2 / ∞ / Frobeniusova-norma matrice `A`
- `[L, U, P] = lu( A )` – LU-faktorizacija matrice `A` s parcijalnim pivotiranjem: `P*A=L*U`

Funkcije numeričke linearne algebre

- `norm( A, 1 )`, `norm( A )`, `norm( A, inf )`, `norm( A, 'fro' )` – 1 / 2 / ∞ / Frobeniusova-norma matrice `A`

- `[L, U, P] = lu( A )` – LU-faktorizacija matrice `A` s parcijalnim pivotiranjem: `P*A=L*U`
- `R = chol( A )` – Cholesky faktorizacija matrice `A = R'*R`

Funkcije numeričke linearne algebre

- `norm( A, 1 )`, `norm( A )`, `norm( A, inf )`, `norm( A, 'fro' )` – 1 / 2 / ∞ / Frobeniusova-norma matrice `A`

- `[L, U, P] = lu( A )` – LU-faktorizacija matrice `A` s parcijalnim pivotiranjem: `P*A=L*U`
- `R = chol( A )` – Cholesky faktorizacija matrice `A = R'*R`

- `[Q, R] = qr( A )` – QR-faktorizacija matrice `A`

Funkcije numeričke linearne algebre

- `norm( A, 1 )`, `norm( A )`, `norm( A, inf )`, `norm( A, 'fro' )` – 1 / 2 / ∞ / Frobeniusova-norma matrice `A`

- `[L, U, P] = lu( A )` – LU-faktorizacija matrice `A` s parcijalnim pivotiranjem: `P*A=L*U`
- `R = chol( A )` – Cholesky faktorizacija matrice `A = R'*R`

- `[Q, R] = qr( A )` – QR-faktorizacija matrice `A`
- `[Q, R] = qr( A, 0 )` – *kompaktna* QR-faktorizacija

Funkcije numeričke linearne algebre

- `norm( A, 1 )`, `norm( A )`, `norm( A, inf )`, `norm( A, 'fro' )` – 1 / 2 / ∞ / Frobeniusova-norma matrice `A`
- `[L, U, P] = lu( A )` – LU-faktorizacija matrice `A` s parcijalnim pivotiranjem: `P*A=L*U`
- `R = chol( A )` – Cholesky faktorizacija matrice `A = R'*R`
- `[Q, R] = qr( A )` – QR-faktorizacija matrice `A`
- `[Q, R] = qr( A, 0 )` – *kompaktna* QR-faktorizacija
- `eig( A )` – svojstvene vrijednosti (i vektori) matrice `A`

Funkcije numeričke linearne algebre

- `norm( A, 1 )`, `norm( A )`, `norm( A, inf )`, `norm( A, 'fro' )` – 1 / 2 / ∞ / Frobeniusova-norma matrice `A`

- `[L, U, P] = lu( A )` – LU-faktorizacija matrice `A` s parcijalnim pivotiranjem: `P*A=L*U`
- `R = chol( A )` – Cholesky faktorizacija matrice `A = R'*R`

- `[Q, R] = qr( A )` – QR-faktorizacija matrice `A`
- `[Q, R] = qr( A, 0 )` – *kompaktna* QR-faktorizacija

- `eig( A )` – svojstvene vrijednosti (i vektori) matrice `A`
- `svd( A )` – singularne vrijednosti (i vektori) matrice `A`

STRINGOVI, STRUKTURE I CELL-ARRAY

String je vektor-redak varijabli tipa `char`:

```
s1 = 'pero'; s2 = 'mirko';
s1( 2:4 )
[s1 s2]   % konkatenacija -- kao kod vektora!
length( s2 )
size( s1 )
```

```
ero
peromirko
5
1 4
```

Postoje i specijalizirane funkcije poput `strcmp` – vidi Help.

Strukture mogu sadržavati elemente raznih tipova:

```
s.ime = 'Mirko';
s.ocjena = 5;
s.matrica = [1 2 3; 4 5 6];
s
```

```
ime: 'Mirko'
ocjena: 5
matrica: [2x3 double]
```

Moguće je doznati je li nešto element strukture:

```
isfield( s, 'ime' )
isfield( s, 'prezime' )
```

```
1
0
```

Cell-array je *matrično* organizirana struktura – elementi mogu biti raznih tipova, ali nemaju imena već su posloženi u matricu.

```
c{1,2} = 'Pero';
c{3,1} = [1 2; 3 4];
c{2,2}.ime = 'Mirko';
c
```

```
[]                      'Pero'
[]                [1x1 struct]
[2x2 double]                []
```

Za cell-array su dostupne neke matrične funkcije poput `size`.

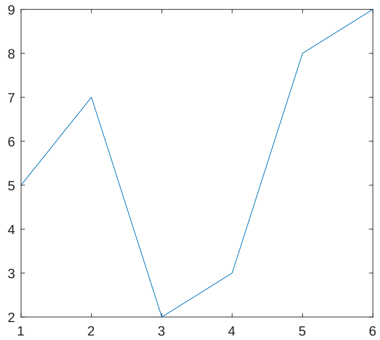
VIZUALIZACIJA REZULTATA

U Matblabu su dostupne brojni alati za crtanje 2D i 3D grafova funkcija, kontura i slično. Ovdje ćemo kroz nekoliko primjera vidjeti samo osnovnu upotrebu funkcije `plot` za crtanje 2D grafova, te `semilogy`.

Grafove je moguće dodatno uređivati pomoću editora koji se otvori prilikom prikaza grafa.

plot

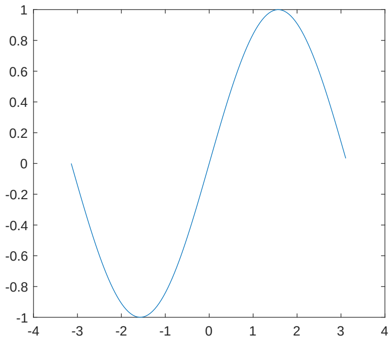
```
y = [5 7 2 3 8 9];
plot( y );
```



Na x-osi: `1` `2` `3` `4` `5` `6`. Točke spojene linijom.

plot

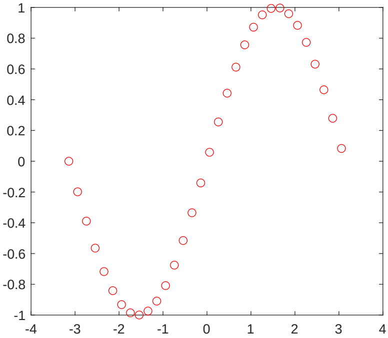
```
x = -pi:0.05:pi; y = sin( x );
plot( x, y );
```



Točke spojene linijom.

plot

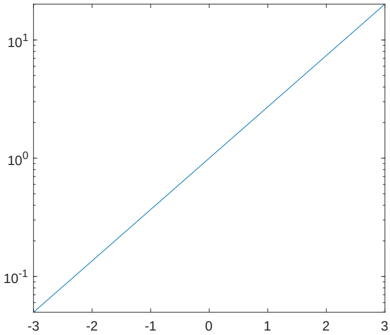
```
x = -pi:0.2:pi; y = sin( x );
plot( x, y, 'ro' );
```



`'ro'` = crveni kružići; `'gx'` = zeleni x; `'y*'` = žute *; `'b+-'` = plavi + spojeni linijom; `'k.:'` = crne točkice spojene isprekidanom linijom

semilogy

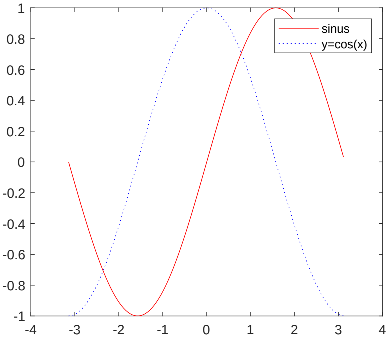
```
x = -3:0.1:3; y = exp( x );
semilogy( x, y );
```



Na y-osi je logaritamska skala. Slično, `semilogx`, `loglog`.

Više grafova na jednoj slici + legenda

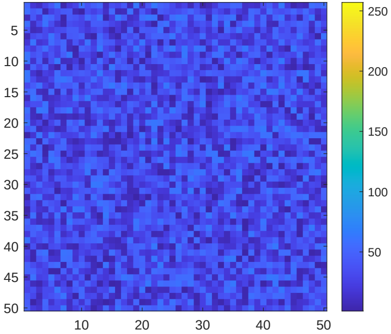
```
x = -pi:0.05:pi; y = sin( x ); z = cos( x );
plot( x, y, 'r-' ); hold on;
plot( x, z, 'b:' ); hold off;
legend( 'sinus', 'y=cos(x)' );
```



`hold on` – svi naredni grafovi idu na trenutnu sliku.

image

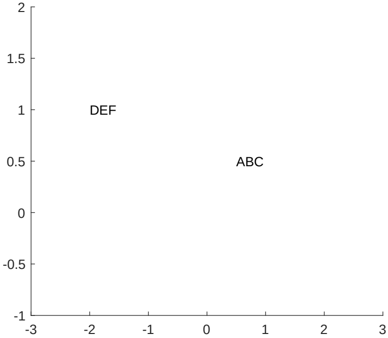
```
a = 64 * rand( 50 );
image( a ); colorbar;
```



Kvadratić (i, j) obojan je bojom $a(i, j) \in \{1, \ldots, 64\}$. Za druge palete boja vidi `doc colormap`.

text

```
text( 0.5, 0.5, 'ABC' ); text( -2, 1, 'DEF' );
set( gca, 'XLim', [-3, 3], 'YLim', [-1, 2] );
```



SKRIPTE I FUNKCIJE

Skripte

Niz naredbi kojeg želimo kasnije ponovno pozvati možemo napisati u bilo kojem editoru teksta (npr. VS Code) i spremiti u .m-datoteku.

Skripte

Niz naredbi kojeg želimo kasnije ponovno pozvati možemo napisati u bilo kojem editoru teksta (npr. VS Code) i spremiti u .m-datoteku.

Matlab također ima svoj editor: *Window->Editor*.

Skripte

Niz naredbi kojeg želimo kasnije ponovno pozvati možemo napisati u bilo kojem editoru teksta (npr. VS Code) i spremiti u .m-datoteku.

Matlab također ima svoj editor: *Window->Editor*.

Ako niz naredbi spremimo u datoteku `skripta.m`, onda ih možemo redom sve izvršiti tako da u interaktivnoj konzoli napišemo naredbu `skripta`.

Skripte

Niz naredbi kojeg želimo kasnije ponovno pozvati možemo napisati u bilo kojem editoru teksta (npr. VS Code) i spremiti u .m-datoteku.

Matlab također ima svoj editor: *Window->Editor*.

Ako niz naredbi spremimo u datoteku `skripta.m`, onda ih možemo redom sve izvršiti tako da u interaktivnoj konzoli napišemo naredbu `skripta`.

Važno je da datoteka `skripta.m` bude

- ili u trenutnom direktoriju (*Current Folder*)
- ili u putanji (*path*). Bilo koji direktorij u putanju možemo dodati preko *File->Set Path...*

Skripte

Unutar skripti dostupne su nam standardne naredbe za grananje, te petlje:

```
if uvjet_1
   naredba_1;
   naredba_2;
elseif uvjet_2
   naredba_3;
else
   naredba_4;
end
```

Primjer:

```
if abs( x - y ) < 1e-8
   'Brojevi x i y su preblizu'
else
   z = 1 / ( x - y );
end
```

Skripte

Petlje:

```
for brojač = vektor-redak
    naredba_1;
    naredba_2;
end

while uvjet
    naredba_1;
    naredba_2;
end
```

Primjer:

```
for i = 3:2:10
    suma = suma + i;
end
```

Dostupne su i standardne naredbe **break** i **continue**.

Funkcije

Funkcije također možemo spremiti u .m-datoteku.
Unutar jedne m-datoteke može biti više funkcija, ali samo prvu (koja mora imati ime kao i datoteka) će biti moguće pozvati izvana. Ostale funkcije unutar m-datoteke su pomoćne.
Unutar skripte nije moguće definirati funkcije.

Definicija funkcije s 2 parametra i 3 povratne vrijednosti:

```
function [x_1, x_2, x_3] = ime_fje( p_1, p_2 )
    % kod funkcije...
    % treba nešto spremiti u varijable x_1, x_2 i x_3.
end
```

Funkciju treba spremiti u datoteku `ime_fje.m`.

Funkcija se iz konzole poziva sa:

```
[var_1, var_2, var_3] = ime_fje( arg_1, arg_2 );
```

Primjer funkcije

U editoru napišemo i spremimo pod imenom `suma_kvadrata.m`:

```
function s = suma_kvadrata( v )
% Funkcija računa sumu kvadrata svih elemenata
% vektora-retka v
    if( size( v, 1 ) ~= 1 )
        error( 'Funkciji nije poslan vektor-redak!' );
    end

    s = 0;
    for i = 1 : length( v )
        s = s + kvadrat( v( i ) );
    end
end

function k = kvadrat( x )
    k = x .^ 2;
end
```

Primjer funkcije

U konzoli pokrenemo:

```
v = [2 5 1];
x = suma_kvadrata( v );
x
```

```
30
```

Funkcija `kvadrat` nije dostupna iz konzole!

Matlab prosljeđuje parametre po vrijednosti – dakle, funkcija ne može promijeniti varijablu koja joj je poslana (stvara se lokalna kopija unutar funkcije).

Zadatak 1: QR faktorizacija

1. Napišite funkciju `[Q, R]=cgs(A)` koja izračunava QR-faktorizaciju matrice `A` koristeći klasični Gram-Schmidtov algoritam:

$$\tilde{q}_k = a_k - \sum_{j=1}^{k-1} \underbrace{(q_j^\mathsf{T} a_k)}_{R(j,k)} q_j$$

2. Napišite funkciju `[Q, R]=mgs(A)` koja izračunava QR-faktorizaciju matrice `A` koristeći modificirani Gram-Schmidtov algoritam: u gornjoj sumi (petlji) treba zamijeniti a_k sa $\tilde{q}_k$.

3. Usporedite "ortogonalnost" faktora `Q` dobivenog pomoću funkcija `cgs` i `mgs`, te Matlabove funkcije `qr`.