



vazmfb.com/

Projektovanje primenom računara

Inženjersko programiranje

2022.

Katedra za vazduhoplovstvo

Mašinski fakultet

Univerziteta u Beogradu

Miloš D. Petrašinić 

Mihailo G. Petrović 

Nikola G. Raičević 



Sadržaj

- Programski jezici i okruženja
- Uvod u GNU Octave
- Radno okruženje
- Tipovi i strukture podataka
- Dodatni paketi
- Skripte i funkcije
- Naredbe uslovnog granjanja i petlje
- Ulaz i izlaz podataka
- Grafički prikaz podataka
- Simbolička izračunavanja



Programski jezici i okruženja

- Veoma je bitno izabrati adekvatan, efikasan i aktuelan programski jezik.
- Programski jezik inženjeru treba da omogući najbržu moguću implementaciju zamišljenog proračuna, brzo izvršavanje programa, jednostavnu analizu rezultata i interoperabilnost u cilju razmene podataka sa drugim inženjerima i njihovim radnim okruženjima.
- Inženjer treba da bude minimalno opterećen sintaksom i vođenjem računa o za njega trivijalnim stvarima i pritom da ima maksimalnu slobodu.



Programski jezici i okruženja

- Aktuelni programski jezik je onaj sa velikim brojem ljudi i kompanija koje učestvuju u njegovom razvoju ili ga koriste što je garant napredovanja programskog jezika kroz vreme.
- Programski jezici Matlab i GNU Octave su specijalizovani za numeričke proračune, dok programski jezici JavaScript, Python i Julia opšti programski jezici ali daleko aktuelniji.
- JavaScript je programski jezik sa najvećim brojem korisnika (V8 JavaScript engine razvija kompanija Google za projekat Chromium i Node.js). Većina svih kompanija na svetu koje se bave programiranjem ili učestvuju u razvoju ili koriste JavaScript u svojim proizvodima.



Programski jezici i okruženja

- Aktuelni programski jezik je onaj sa velikim brojem ljudi i kompanija koje učestvuju u njegovom razvoju ili ga koriste što je garant napredovanja programskog jezika kroz vreme.
- Programski jezici Matlab i GNU Octave su specijalizovani za numeričke proračune, dok programski jezici JavaScript, Python i Julia opšti programski jezici ali daleko aktuelniji.
- JavaScript je programski jezik sa najvećim brojem korisnika (V8 JavaScript engine razvija kompanija Google za projekat Chromium i Node.js). Većina svih kompanija na svetu koje se bave programiranjem ili učestvuju u razvoju ili koriste JavaScript u svojim proizvodima.



Programski jezici i okruženja

- Mogućnost korišćenja programskih jezika visokog nivoa zajedno sa programskim jezicima C/C++ ili FORTRAN može biti velika prednost.
- Količina već napisanog koda i biblioteka koje je moguće integrisati mogu biti jedan od glavnih aduta pri izboru adekvatnog programskog jezika.
- Izuzetno je bitno koliko je dobra dokumentacija za programski jezik, koliko ima zvaničnih primera za funkcionalnosti i da li je on otvorenog koda.



Projekti otvorenog koda

- Korisnik direktno može ispravljati greške, dodavati nove funkcionalnosti i to ne samo za sebe već za celu zajednicu!
- Samim korišćenjem i promovisanjem programa, korisnik je deo zajednice.
- Preko nekoliko dostupnih kanala (zvanični forum, društvene mreže ili izmenom izvornog koda i slanjem izmena) korisnik može direktno da komunicira sa osnivačima projekta i aktivnim programerima.
- Nove funkcionalnosti, ispravke grešaka i druga unapređenja su dostupna na dnevnom nivou.



Projekti otvorenog koda

- U odnosu na komercijalne programske pakete korisnik ima svu slobodu (*"Free as in Freedom"*).
- **Nije potrebno materijalno ulaganja od strane korisnika na početku poslovanja!**
- Mogućnost slobodnog sticanja i širenja znanja.
- Korisniku nije moguće onemogućiti da primeni svoje znanje kao što je to moguće prostim oduzimanjem licence za korišćenje u slučaju komercijalnih paketa.
- Umeća i znanja vezana za korišćenje komercijalnih paketa su korisna samo ukoliko i kada su ti paketi dostupni što je izuzetno ograničavajuć faktor.



Literatura

- John W. Eaton, et al.,
Free Your Numbers
<https://octave.org/octave.pdf>
- Dokumentacija:
<https://octave.org/doc/interpreter/>
- GNU Octave Wiki:
https://wiki.octave.org/GNU_Octave_Wiki
- Octave Forge paketi:
<https://octave.sourceforge.io/>
- Octave Forge dokumentacija:
<https://octave.sourceforge.io/docs.php>



GNU Octave je programski paket koji sadrži programski jezik visokog nivoa, prvenstveno namenjen numeričkim proračunima. Program pomaže prilikom numeričkog rešavanja linearnih i nelinearnih problema. Pored ovoga, jezik je predviđen za matrična izračunavanja, rešavanje sistema linearnih jednačina, iscrtavanje grafika funkcija itd. Mogućnosti programa se proširuju programima koji su pisani u programskom jeziku **GNU Octave** koji mogu da se implementiraju zajedno sa drugim programskim jezicima kao što su C/C++, Fortran i Java. Pored numeričkih izračunavanja postoji paket **symbolic** za simbolička izračunavanja (implementiran na osnovu `sympy` biblioteke programskog jezika Python).

Preporučuje se korišćenje **verzije 7.2.0** ili novije.



The screenshot shows the GNU Octave 6.0.90 interface. The window title is "Octave". The menu bar includes "File", "Edit", "Debug", "Window", "Help", and "News". The "Current Directory" is set to "C:\Users\petra".

The interface is divided into several panes:

- File Browser:** Shows a tree view of the file system under "C:\Users\petra". The "Downloads" folder is highlighted. Annotation **C** points to this pane.
- Workspace:** A table with columns "Name", "Class", "Dimension", and "Value". It is currently empty. Annotation **B** points to this pane.
- Command Window:** Displays the GNU Octave version (6.0.90), copyright information (© 2020 The Octave Project Developers), and a disclaimer: "This is free software; see the source code for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'." It also shows configuration details and links for more information. Annotation **A** points to this pane.
- Current Directory:** A dropdown menu showing "C:\Users\petra". Annotation **D** points to this dropdown.
- Navigation:** Green and blue arrows point to the "Up" and "Down" directory navigation buttons. Annotation **E** points to these buttons.
- Bottom Panel:** Contains tabs for "Command Window", "Documentation", and "Editor". Annotation **F** points to the "Editor" tab.



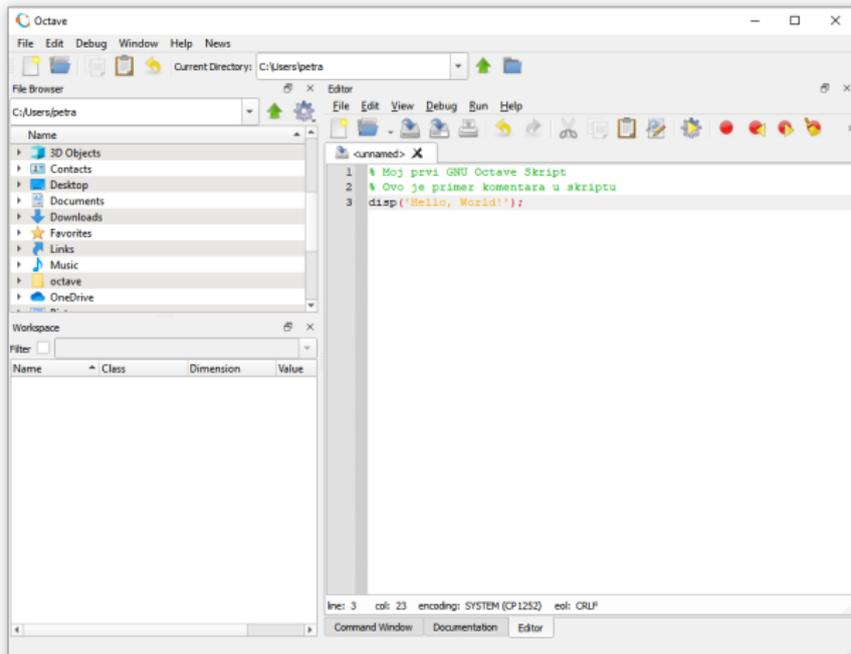
Delovi radnog prostora

- **A (Command Window)** - Komandni prozor za unos komandi i ispisivanje rezultata,
- **B (Workspace)** - Prozor sa prikazom dostupnih promenljivih,
- **C (Current Folder)** - Prozor sa prikazom dostupnih datoteka u okviru trenutno otvorenog foldera,
- **D** - Apsolutna putanja trenutno otvorenog foldera,
- **E** - Dugme `Browse directories` za izmenu foldera.
- **F** - Dugme za otvaranje Editor prozora.

Napomena: Ovaj izgled korisničkog interfejsa se omogućava sa `Edit/Preferences.../General/Interface/Style` → Fusion



Prozor Editor i skript





Tipovi podataka

- Celobrojni tipovi podataka (`int8`, `uint8`, `int16`, `uint16`, `int32`, `uint32`, `int64` i `uint64`),
- Tipovi podataka u pokretnom zarezu (`single`, `double`),
- Logički tip (`true` i `false`),
- Znakovni tip (`char`),
- Funkcije (`function handle`),
- Klase (`class`),
- Čelije (`cell`),
- Specijalne vrednosti (`pi`, `i`, `j`, `inf` i `NaN`).



Strukture podataka

- Matrice ($A = [1, 2; 3, 4]$),
- Višedimenzionalni nizovi ($B(:, :, 2) = [5, 6; 7, 8]$),
- Nizovi karaktera - **string** ($C = \text{'Hello'}$),
- Nizovi ćelija ($D = \{ 'a', 1, [1, 2] \}$).



Aritmetičke operacije

- $+$ - sabiranje
- $-$ - oduzimanje
- $.*$ - množenje
- $./$ - desno deljenje
- $.\backslash$ - levo deljenje
- $.^$ - stepenovanje
- $.**$ - stepenovanje
- $.'$ - transponovanje
- $'$ - konjugovano kompleksno transponovanje
- $*$ - matrično množenje
- $/$ - matrično desno deljenje
- \backslash - matrično levo deljenje
- $^$ - matrično stepenovanje
- $**$ - matrično stepenovanje



Relacioni operatori

- $<$ - manje
- $>$ - veće
- $<=$ - manje ili jednako
- $>=$ - veće ili jednako
- $==$ - jednako
- \neq ili $!=$ - različito



Inkrementalni operatori

- $x++$ - ekvivalentno sa $x = x+1$;
- $x--$ - ekvivalentno sa $x = x-1$;
- $++x$ - ekvivalentno sa $ans = x, x = x+1$;
- $--x$ - ekvivalentno sa $ans = x, x = x-1$;



Logičke operacije

- `&` ili `&&` ili `and` - konjunkcija
- `|` ili `||` ili `or` - disjunkcija
- `~` ili `!` ili `not` - negacija
- `xor` - ekskluzivna disjunkcija
- `all` - konjunkcija svih elemenata vektora
- `any` - disjunkcija svih elemenata vektora



- `pkg list` - prikaz dostupnih dodatnih paketa,
- `pkg load`, `pkg unload` - aktiviranje i deaktiviranje dodatnog paketa,
- `pkg install`, `pkg uninstall` - instalacija i uklanjanje dodatnog paketa,
- `pkg updatel` - instalacija novije verzije dodatnog paketa,
- `pkg build`, `pkg rebuild` - kompajliranje dodatnog paketa,
- `compare_versions` - upoređivanje verzija.



- Uobičajeno je da fajlovi u kojima su pisani programi za **GNU Octave** imaju ekstenziju **m**.
- **Skript** fajlovi ne prihvataju argumente i ne vraćaju vrednost, koriste promenljive iz radnog prostora. Koriste se kada je neophodno izvršiti neki niz komandi koji može da se sačuva.
- **Funkcijski** fajlovi prihvataju ulazne argumente i vraćaju vrednost, koriste lokalni radni prostor i lokalne promenljive. Koriste se kada je potrebno napraviti nove komande.
- Prilikom pokretanja **skripte (Script)** ili funkcije (**Function**) neophodno je da bude sačuvana u okviru foldera koji je trenutno aktivan (ili funkcijom `addpath` dodati taj folder medju foldere koje **GNU Octave** pretražuje).
- Komandom `clear` se brišu promenljive iz radnog prostora, komandom `clc` se briše sve iz komandnog prozora i sa `close all` se zatvaraju grafici. Program je moguće prekinuti sa **CTRL+C**.



Primer skripte - primer1.m

```
1 %% Naslov u skripti
2 % Komentar u skripti
3 clear, clc, close all % brisanje i zatvaranje
4 A = 1 % prikazuje se u komandnom prozoru
5 B = 1; % ne prikazuje se u komandnom prozoru sa ';'
6 C = A+B
7 a = [1,2] % vektor vrsta
8 b = [3;4] % vektor kolona
9 c = [5,6,7;8,9,10] % matrica 2x3
10 d = size(c) % d = [2,3]
11
12 abe = a.*b % mnozenje po elementima
13 abm = a*b % matricno mnozenje
14 b2 = b(2) % drugi element vektora
15 c12 = c(1,2) % element u prvoj vrsti i drugoj koloni
16
17 i1 = a(1) == d(1) % i1 = logical 0
18 i2 = a(1) ~= d(1) % i2 = logical 1
```



Primer funkcije - podeli.m

```
1 function [ceoBroj,ostatak] = podeli(prviBroj,drugiBroj)
2 % Funkcija koja deli dva broja i vraca ceo broj i ostatak
3 % podeli(11.3,3) = [3, 2.3]
4
5 ceoBroj = floor(prviBroj./drugiBroj);
6 ostatak = mod(prviBroj,drugiBroj);
7 end
```

- **ceoBroj** i **ostatak** - izlazni argumenti,
- **podeli** - naziv funkcije,
- **prviBroj** i **drugiBroj** - izlazni argument,
- linije 2 i 3 - objašnjenje funkcije (vidi »**help podeli**),
- linije 5 i 6 - naredbe u funkciji.



Implementirane funkcije - tip promenljive i prikaz

- `whos` - osnovne informacije o promenljivim,
- `isempty`, `islogical`, `ischar`, `isstring`, `isnumeric`, `isfloat`, `isinteger`, `isscalar`, `isreal`, `isnan`, `isinf`, `isfinite`, `issparse`, `isvector`, `isobject`, `ismatrix`, `isrow`, `iscolumn`, `iscell` - provera promenljive,
- `disp` - prikaz promenljive u komandnom prozoru,
- `num2str` - konverzija numeričkog tipa u string zadatog formata i preciznosti.



Implementirane funkcije - elementarna matematika

- `sqrt` - kvadratni koren,
- `exp` - eksponencijalna funkcija,
- `log`, `log10`, `log2` - logaritam za različite osnove,
- `abs`, `sign` - apsolutna vrednost i znak,
- `round`, `floor` i `ceil` - zaokruživanje brojeva,
- `sin`, `cos`, `tan`, `cot`, `asin`, `acos`, `atan`, `atan2`, `acot` - trigonometrijske funkcije (argumenti u radijanima, za argumente u stepenima dodati `d` na kraj, npr. `sind`),
- `sinh`, `cosh`, `tanh`, `coth`, `asinh`, `acosh`, `atanh`, `acoth` - hiperboličke funkcije (argumenti u radijanima).



Implementirane funkcije - matrice

- `linspace` - formiranje vektora sa ekvidistantnim tačkama,
- `meshgrid` - formiranje dve dvodimenzione matrice na osnovu dva vektora (pravougaona mreža),
- `ones`, `zeros`, `eye`, `diag` - formiranje matrica,
- `size`, `ndims`, `length`, `numel` - dimenzije i broj elemenata,
- `reshape`, `fliplr`, `fliud`, `repmat` - menjanje oblika matrica,
- `sum`, `prod`, `cross`, `norm`, `det`, `rank`, `min`, `max`, `mean`, `sort`, `unique` - operacije nad matricama i vektorima i osobine.



Primer skripte - primer2.m

```
1 A1 = 2:5 % formiranje niza
2 A2 = 2.1:5
3 B1 = 3.1:2.2:20
4 C = linspace(1,10,3)
5 whos A1 % informacije o promenljivoj
6 isnumeric(A1(2)) % provera tipa
7 disp(['A1=' ... % nastavljajanje naredbe u sledecem redu
8     num2str(A1(2))]); % prikaz broja u stringu
9
10 Blr = round(B1(1)) % zaokruzivanje brojeva
11 sB1 = sum(B1) % suma elemenata vektora
12 mB1 = mean(B1) % srednja vrednost elemenata vektora
13 pA = prod(A1) % proizvod elemenata vektora
14 flipA1 = fliplr(A1) % okretanje vektora
15
16 % Sistem linearnih jednačina Ax=B
17 % 2x+3y = 5, 3x+5y = 7
18 A = [2,3;3,5], B = [5;7], x = A\B
```



Primer skripte - primer3.m

```
1 A = rand(6,5) % matrica 6x5
2 % rand - uniformno rasporedjeni pseudoslucajni brojevi
3
4 Av2 = A(2,:) % druga vrsta
5 Ak3 = A(:,3) % treca kolona
6 Ave = A(end,:) % poslednja vrsta
7 Av224 = A(2,2:4) % elementi u vrsti 2 i kolonama 2, 3 i 4
8 Avn = A(1,1:2:end) % vrsta 1 i neparne kolone
9
10 B1 = zeros(4,3) % matrica 4x3 puna nula
11 B2 = ones(3,5) % matrica 3x5 puna jedinica
12
13 C = [2,7,9,1]
14 Cm = diag(C) % matrica sa vektorom C na dijagonalii
```



Implementirane funkcije - numerika

- `interp1`, `interp2`, `interp2d` - interpolacija,
- `fzero`, `roots` - nula funkcije ili polinoma,
- `integral`, `trapz`, `quad` - numeričko izračunavanje integrala,
- `diff` - razlika i numeričko diferenciranje,
- `polyfit`, `splinefit` - aproksimacija podataka,
- `lsode`, `ode15i`, `ode15s`, `ode23`, `ode23s`, `ode45` - diferencijalne jednačine.



Naredbe if i switch

Naredbe `if` i `switch` su naredbe uslovnog granjanja, na osnovu izaraza se određuje koji će blok naredbi biti izvršen.

```
1 a = 1; b = 2;
2 if a > b
3     disp('a>b');
4 elseif a < b
5     disp('a<b');
6 else
7     disp('a=b');
8 end
```

```
1 a = 2;
2 switch a
3     case 1
4         disp('a = 1');
5     case 2
6         disp('a = 2');
7     otherwise
8         disp('a ~= [1,2]');
9 end
```



Naredba try

Naredba `try` se koristi kada je potrebno probati izvršenje nekog bloka naredbi i ukoliko dodji do greške moguće je registrovati (u okviru naredbe `catch`) i nastaviti sa izvršavanjem programa.

```
1  A = [1,2];
2  try
3      B = A(3);
4  catch err
5      disp('Doslo je do greske');
6      disp(err.identifier);
7      disp(err.message);
8      disp(err.stack);
9      B = 0;
10 end
11 A,B
```



Petlje for i while

Petlja **for** izvršava blok naredbi unapred zadati broj puta dok petlja **while** izvršava blok naredbi sve dok je ispunjen uslov.

```
1  for i = 1:3
2      is = num2str(i);
3      disp(['ciklus ' is]);
4  end

1  i = 1;
2  while i < 4
3      is = num2str(i);
4      disp(['ciklus ' is]);
5      i = i+1;
6  end
```



Naredbe return, break, continue i pause

- **return** - prekida izvršenje funkcije,
- **break** - prekida izvršenje petlje,
- **continue** - omogućava prelazak na sledeći ciklus u petlji bez završetka prethodnog,
- **pause** - prekida izvršenje programa na određeno vreme.



Anonimne funkcije

Definiše se tip podataka `function_handle`, to je funkcija definisana jednom linijom koda bez snimanja u `m` fajl (linija 3 u sledećem listingu). Prikazano je izračunavanje određenog integrala za ovu funkciju i nalaženje nule.

```
1 x0 = 0;  
2 x1 = 1;  
3 p = @(x) x.^2-1 % anonimna funkcija  
4 a = p(2) % a = 4  
5 % Izracunavanje integrala x^2 od 0 do 1  
6 I = integral(p,x0,x1)  
7 nula = fzero(p,x0) % x0 je pretpostavka
```



Implementirane funkcije - osnovne

- `load` i `save` - učitavanje i snimanje promenljivih,
- `fopen` i `fclose` - otvaranje i zatvaranje fajlova,
- `fwrite`, `fprintf`, `sprintf` - pisanje i formatiranje podataka,
- `fread`, `fgetl`, `fscanf`, `fgets`, `fgetl`, `sscanf`, `fileread`, `textscan` - čitanje podataka,
- `feof`, `ferror`, `frewind`, `fseek`, `ftell` - pomoćne funkcije prilikom čitanja i pisanja.



Implementirane funkcije - dodatne

- `imread`, `imwrite` - učitavanje i snimanje slika,
- `serial`, `seriallist` - serijska komunikacija (paket `instrument-control`),
- `web`, `webread`, `webwrite`, `ftp` - internet stranice i protokoli,
- `jsondecode`, `jsonencode` - JSON objekti (od verzije 6.1.0).



Formatiranje podataka

Na sledećoj slici je prikazana struktura operatora za formatiranje podataka.



Dokumentacija: <https://octave.org/doc/v5.2.0/Table-of-Input-Conversions.html>

[//octave.org/doc/v5.2.0/Table-of-Input-Conversions.html](https://octave.org/doc/v5.2.0/Table-of-Input-Conversions.html)



Primer skripte - primer4.m

```
1 fid = fopen('fajl.txt','w') % otvaranje fajla za pisanje
2 fprintf(fid,'Hello, World!\n') % upisivanje teksta u fajl
3
4 % pravljenje stringa od broja
5 spi = sprintf('%.8f',pi); % pi sa 8 decimala
6 fprintf(fid,['pi = ' spi '\n'])
7 fprintf(fid,'pi = %d\n',2)
8 fprintf(fid,'slovo %c\n','A')
9 fclose(fid) % zatvaranje fajla
10
11 fid = fopen('fajl.txt','r') % otvaranje fajla za citanje
12 i = 1; t = {};
13 while ~feof(fid)
14     t{i} = fgetl(fid); % citanje teksta iz fajla
15     t{i}
16     i = i+1;
17 end
18 fclose(fid)
```



Jednostavan grafički prikaz podataka u okviru **GNU Octave**-a je jedna od najznačajnijih njegovih karakteristika za inženjere. Grafički ili vizuelni prikaz podataka se može podeliti na **dvodimenzionalne i trodimenzionalne grafike**.

- `figure` - prozor grafika,
- `subplot` - više grafika u prozoru,
- `gca` - aktivan grafik,
- `cla` - brisanje podataka sa grafika,
- `hold`, `grid`, `box` - zadržavanje grafika, prikaz mreže i kontura grafika,
- `drawnow` - osvežavanje podataka,
- `get(gca, 'ColorOrder')` - trenutna paleta boja za prikaz.



Podšavanje grafika

- `title` - naslov grafika,
- `axis` - podesavanje osa,
- `xlim`, `zlim`, `zlim` - granice osa,
- `xlabel`, `ylabel`, `zlabel` - naziv osa,
- `view`, `rotate`, `rotate3d`, `zoom`, `pan`, `colormap`, `colorbar`, `shading` - podesavanje prikaza,
- `text` - ispisivanje teksta,
- `legend` - legenda grafika.



Dvodimenzionalni grafici

- `plot`, `area`, `loglog`, `semilogx`, `semilogy` - crtanje linija,
- `fplot` - crtanje funkcija,
- `fill` - crtanje poligona,
- `hist`, `pie`, `scatter` - prikaz raspodele podataka,
- `bar`, `barh`, `stem`, `stairs` - prikaz diskretnih podataka,
- `polar` - polarni grafici,
- `contour`, `contourf`, `contourc` - konturni grafici,
- `feather`, `quiver` - vektorski grafici.



Trodimenzionalni grafici

- `plot3` - crtanje linija,
- `pie3`, `scatter3` - prikaz raspodele podataka,
- `bar3`, `barh3`, `stem3` - prikaz diskretnih podataka,
- `contour3` - konturni grafici,
- `quiver3` - vektorski grafici.
- `surf`, `mesh`, `trisurf`, `trimesh`, `waterfall`, `ribbon`, `patch` - crtanje površi i poligona.



Izgled linija

Izgled linije na grafiku je definisan sa tri komponente i to **tip linije, markeri i boja**. Svaka od ovih komponenata se podešava na odredjeni način.

Dokumentacija:

<https://octave.org/doc/v5.2.0/Line-Styles.html>

<https://octave.org/doc/v5.2.0/Colors.html>

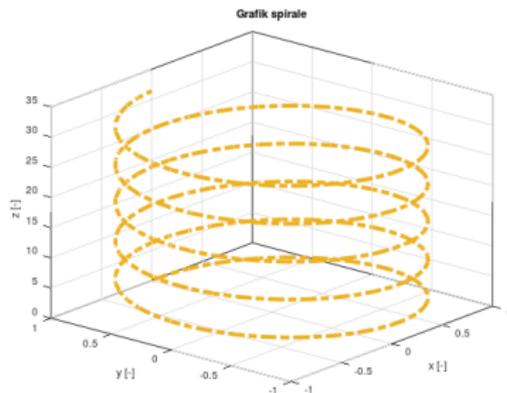
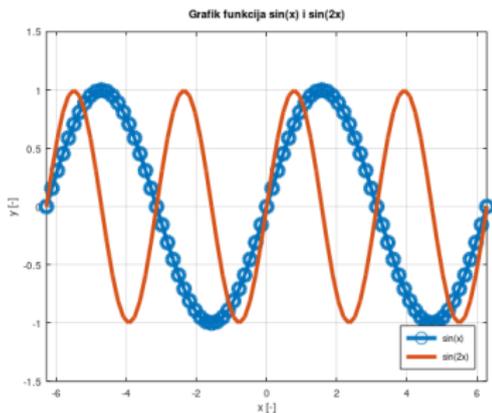


Primer skripte - primer5.m

```
1 close all % zatvaranje svih grafika
2 figure(1) % prozor prvog grafika
3 hold on, grid on, box on
4 x = -2*pi:pi/20:2*pi;
5 plot(x, sin(x), '-o', 'LineWidth', 2); % crtanje linije
6 plot(x, sin(2*x), '-', 'LineWidth', 2);
7 title(['Grafik funkcija sin(x) i sin(2x)']) % naslov
8 xlabel('x [-]'), ylabel('y [-]') % naziv osa
9 legend('sin(x)', 'sin(2x)', 'Location', 'SouthEast')
10 axis([-2*pi, 2*pi, -1.5, 1.5]); % granice osa
11
12 figure(2) % prozor drugog grafika
13 hold on, grid on, box on, boje = get(gca, 'colororder');
14 z = 0:pi/20:10*pi; x = sin(z); y = cos(z);
15 plot3(x, y, z, '-.', 'LineWidth', 2, 'Color', boje(3, :));
16 title('Grafik spirale')
17 xlabel('x [-]'), ylabel('y [-]'), zlabel('z [-]')
18 view(-50, 25) % podesavanje pogleda
```



Prikaz grafika





Sva prethodna izračunavanja su bila prikazana za numerički tip podataka, pored ovog tipa moguće je u okviru **GNU Octave**-a raditi i sa **simboličkim promenljivim** zahvaljujući **Octave Symbolic Package**-u.

Ovaj paket je moguće instalirati sledećom komandom (internet je neophodan zbog preuzimanja paketa):

```
1 pkg("install", ["https://github.com/cbm755/octsympy/"...
2   "releases/download/v2.9.0/"...
3   "symbolic-win-py-bundle-2.9.0.tar.gz"]);
```

Sada je za korišćenje paketa dovoljno samo:
»pkg load symbolic



Implementirane funkcije

- **sym**, **syms**, **assume** - formiranje simboličke promenljive ili izraza,
- **symvar** - simboličke promenljive iz izraza,
- **function_handle** - pretvaranje izraza u anonimnu funkciju,
- **subs** - simbolička supstitucija,
- **vpa** - numerička vrednost izraza,
- **digits** - broj cifara za numeričke vrednosti.



Implementirane funkcije

- `solve`, `dsolve`, `vpasolve` - rešavanje jednačina,
- `simplify`, `coeffs`, `expand`, `numden` - pojednostavljivanje i preuredjenje izraza,
- `diff`, `jacobian` - diferenciranje,
- `int` - integracija,
- `limit` - granična vrednost funkcije,
- `symsum` - sumiranje.



Primer skripte - primer6.m

```
1 pkg("load","symbolic");
2
3 syms x y % formiranje simbolicke promenljive
4
5 simplify(cos(x)^2-sin(x)^2) % pojednostavljenje izraza
6 f = x^3+2*x^2 % simbolička funkcija
7 diff(f) % diferenciranje
8 int(f) % integracija
9 limit((1+1/x)^x,x,inf) % granicna vrednost
10
11 f_h = function_handle(f,'vars',x) % anonimna funkcija
12 fx = f_h(1.5) % numericka vrednost za x = 1.5
13 f2 = subs(f,x,sym(15)/10) % supstitucija
14 f2n = vpa(f2) % numericka vrednost
15
16 e1 = 2*x+3*y == 5; % prva jednacina
17 e2 = 3*x+5*y == 7; % druga jednacina
18 [x1,y1] = solve(e1,e2,x,y) % resenje sistema jednacina
```



Hvala na pažnji!