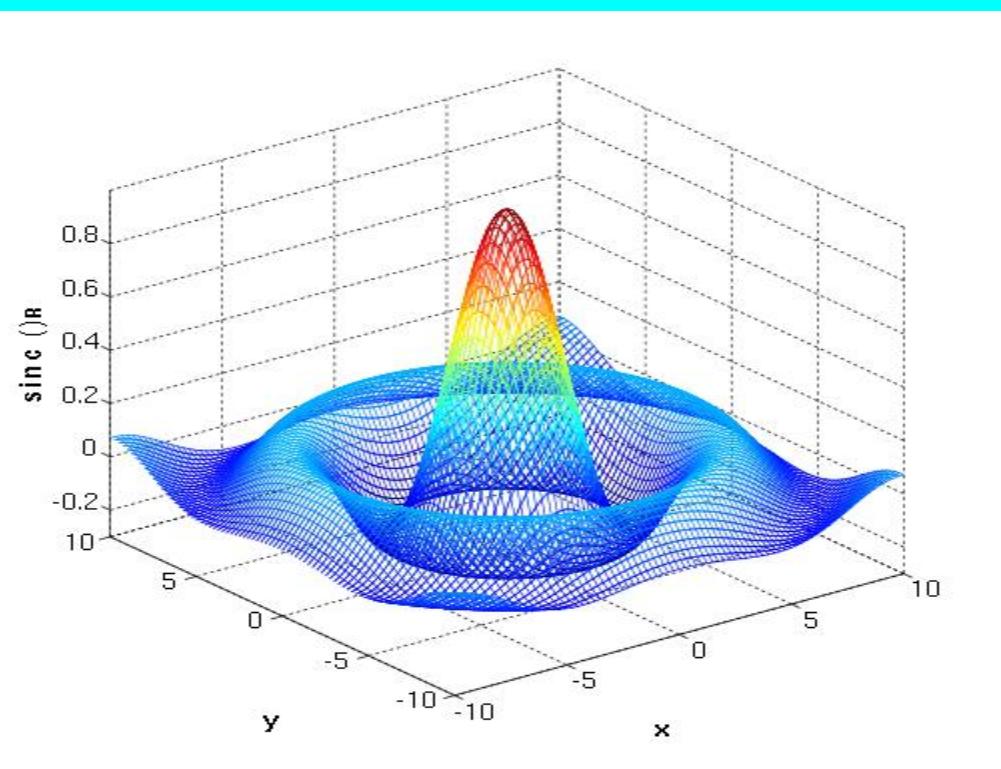
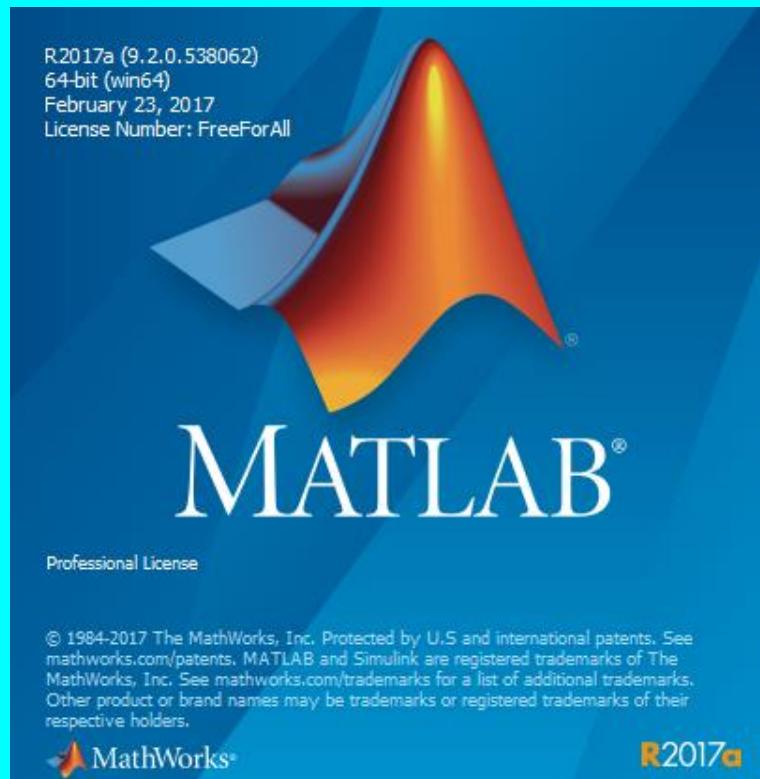




Računarski alati



Prof. dr Milošević Borivoje

MatLab - Matrix Laboratory.

Pored klasične instalacije MATLAB, koja "teži" oko 10GB, možemo je pokrenuti sa on line Google ili Azure Cloud platforme sa linka:
<https://www.mathworks.com/products/matlab-online.html>.
Preporučujem da se prvo ulogujete na MATLAB (e-mail i pasvord).



The screenshot shows a web browser displaying the MathWorks MATLAB Online page. The URL in the address bar is <https://www.mathworks.com/products/matlab-online.html>. The page features the MathWorks logo and navigation links for Products, Solutions, Academia, Support, Community, and Events. A search bar is visible. The main content area is titled "MATLAB Online" and includes links for Overview, Specifications and Limitations, and System Requirements. Below this is a large image of a person's hands working on a laptop keyboard. Text on the page reads "MATLAB Online" and "Use MATLAB and Simulink through your web browser". A blue button at the bottom left says "Start using MATLAB Online" with a blue arrow pointing towards it. At the bottom, a note states "MATLAB Online is available with select licenses. [Check your eligibility](#)".

MatLab - Matrix Laboratory.

MATLAB

Search Help Center

Get Help

- Help Center
- Answers
- File Exchange
- Videos

You do not have an active license linked with your MathWorks Account
Choose an option below to get started

 Link a License

Link your account to a license to access MATLAB

[Link](#)

 Try 1 Month Free

Test-drive MATLAB Products with a free trial

[Try Now](#)

 Buy Now

Purchase products or services

[Buy Now](#)

Learn

Online Training

 MATLAB Onramp

Unlimited Access

[Start](#)

 Machine Learning Onramp

Unlimited Access

[Start](#)

 Deep Learning Onramp

Unlimited Access

[Start](#)

[View more ▾](#)

[View all courses](#)

Možemo izabrati opciju iz Online Training - MATLAB Onramp i startovati je. Prikazaće se veliki izbor alata za koje dobijamo neograničeni pristup i koje možemo koristiti :

MatLab - Matrix Laboratory.

MATLAB Onramp

First time here?

Course Overview

Familiarize yourself with the course.

[Course Overview](#)

Commands

Enter commands in MATLAB to perform calculations and create variables.

[Entering Commands](#)

[Naming Variables](#)

[Saving and Loading Variables](#)

[Using Built-in Functions and Constants](#)

MATLAB Desktop and Editor

Write and save your own MATLAB programs.

 [MATLAB Desktop and Editor](#)

[The MATLAB Editor](#)

[Running Scripts](#) ←

[Debugging MATLAB Code](#)

Vectors and Matrices

Create MATLAB variables that contain multiple elements.

[Manually Entering Arrays](#)

[Creating Evenly-Spaced Vectors](#)

[Array Creation Functions](#)

Pokrenuti opciju Running Scripts. NAPOMENA: učitavanje traje malo duže !!!

MatLab - Matrix Laboratory.

The screenshot shows the MATLAB Onramp interface. At the top, it says "MATLAB Onramp (3% complete)" and "Borivoje Milosevic". The menu bar includes "HOME", "LIVE EDITOR" (selected), and "VIEW". Below the menu is a toolbar with icons for Text, Code, Control, Task, Refactor, Section Break, Run Section, Run and Advance, Run to End, Run, Step, and Stop. A live script named "runscript mlx" is open in the editor. The script contains the following code:

```
r = 3  
x = pi*r^2
```

The output pane shows the results:

```
r = 3  
x = 28.2743
```

The left panel, titled "Task 1", contains the following instructions:

TASK
Run the script to populate the output pane.

Then modify the command for **Task 1** so that `r` has a value of `0.5`.

At the bottom of the task panel, there are "Hint | See Solution | Reset" buttons and a "Submit" button.

Dobili smo uslužni editor MATLAB-a sa svim funkcijama. U regionu Task unosimo podatke, u Calculate Area pišemo npr. Funkciju i sa opcijom Run ili Submit je izvršavamo.

MatLab - Matrix Laboratory.

MATLAB Desktop and Editor > Running Scripts

← PREVIOUS | NEXT →

Task 1

This live script contains formatted text, code, and section breaks. In this course, scripts will include Task headers to show where you should enter your code. MATLAB will display any output in the output pane to the right of your code.

You can test your code before submitting by running the script. To execute the entire script, click the Run button.

TASK
Run the script to populate the output pane.

Then modify the command for Task 1 so that `r` has a value of `0.5`.

Hint | See Solution | Reset Submit

LIVE EDITOR

HOME LIVE EDITOR VIEW

Text Normal Code Control Task Refactor Run Section Section Break Run and Advance Run to End Run Step Stop

runscript mlx * +

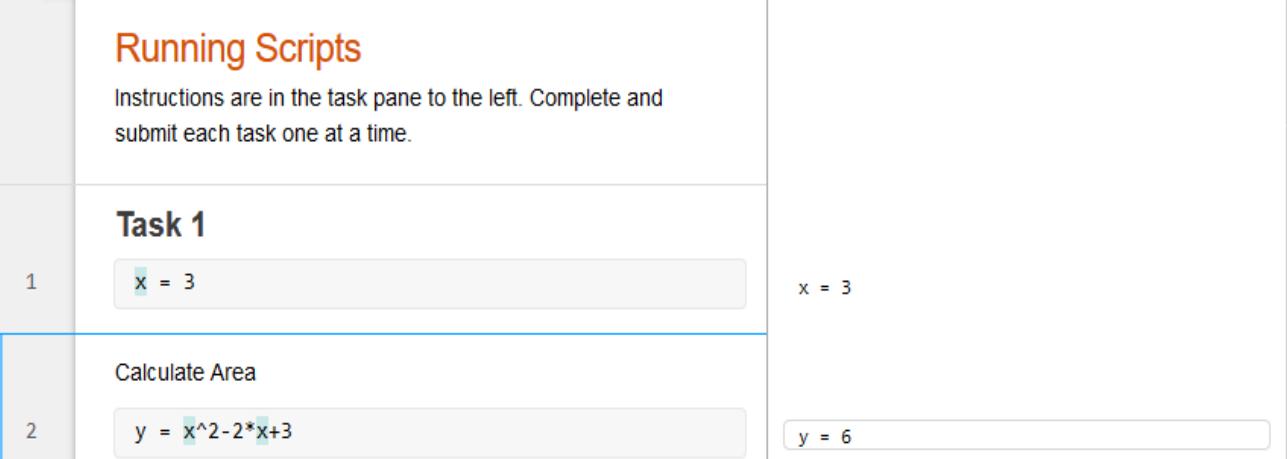
Running Scripts

Instructions are in the task pane to the left. Complete and submit each task one at a time.

Task 1

1 `x = 3` `x = 3`

2 Calculate Area `y = x^2-2*x+3` `y = 6`



MatLab - Matrix Laboratory.

Task 1

This live script contains formatted text, code, and section breaks. In this course, scripts will include Task headers to show where you should enter your code. MATLAB will display any output in the output pane to the right of your code.

You can test your code before submitting by running the script. To execute the entire script, click the  Run button.

TASK

Run the script to populate the output pane.

Then modify the command for **Task 1** so that **r** has a value of **0.5**.

[Hint](#) | [See Solution](#) | [Reset](#)

[Submit](#)

[Further Practice](#)

LIVE EDITOR

HOME

VIEW

Text

B I U M

Code

Control

Task

Refactor

Section Break

Run Section

Run and Advance

Run to End

SECTION

Run

Step

Stop

runscript.mlx * +

Running Scripts

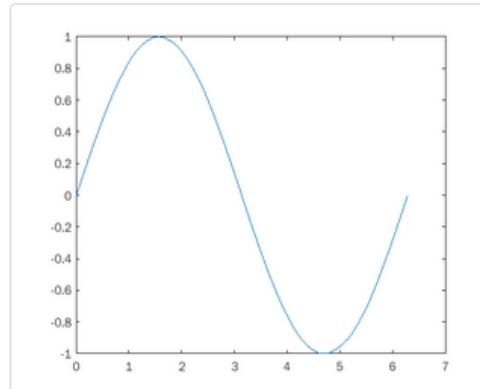
Instructions are in the task pane to the left. Complete and submit each task one at a time.

Task 1

```
x = 0:pi/100:2*pi;
y = sin(x);
```

Calculate Area

```
figure % opens new figure window
plot(x,y)
```



MatLab - Matrix Laboratory.

- Cilj ovog predmeta je da prikaže fundamentalne mogućnosti i funkcije MatLab paketa. Kako ovaj paket pruža ogroman broj mogućnosti za rad na mnogim naučnim poljima, rad sa njim se oslanja na korišćenje primarnih struktura a zatim dopunjavanje i izučavanje sekvenci.
- Biće dat osnovni pregled programa - izgled radne površine, podaci o mogućnostima i značaju prozora koji su dostupni, kao i vrste skripti koje su korišćene za prikaz rada raznih funkcija u narednim poglavljima. Takođe je dat pregled sintakse i pravila korišćenja i skladištenja varijabli, izraza, konstanti i numeričkih i nenumeričkih ugrađenih funkcija.
- Kako je MatLab prvobitno služio za rad sa matricama, i najnovija verzija je zadržala konvenciju da se forma podataka u MatLabu posmatra kroz niz koji ne zahteva dimenzionisanje. Rad sa numeričkim, znakovnim i logičkim nizovima i funkcijama koje olakšavaju rad sa konkretnim zadacima, kako sa jednodimenzionalnim, tako i sa višedimenzionalnim nizovima.
- Takođe fundamentalnu, prilično široku oblast čine grafikoni. Oni se mogu kreirati i uređivati na više načina. U sledećem poglavlju kroz konkretnе primere biće prikazane njihove vrste, mogućnosti u funkcionalnosti ali i u dizajnu, kao i upotreba na konkretnim rezultatima programa.
- Poslednji segment ovog predmeta, biće posvećeno grafičkom programskom okruženju zasnovanom na MatLab paketu koje služi za modeliranje, simulaciju i analizu višedomenskih dinamičkih sistema.

MatLab - Matrix Laboratory.

Osnovne osobine

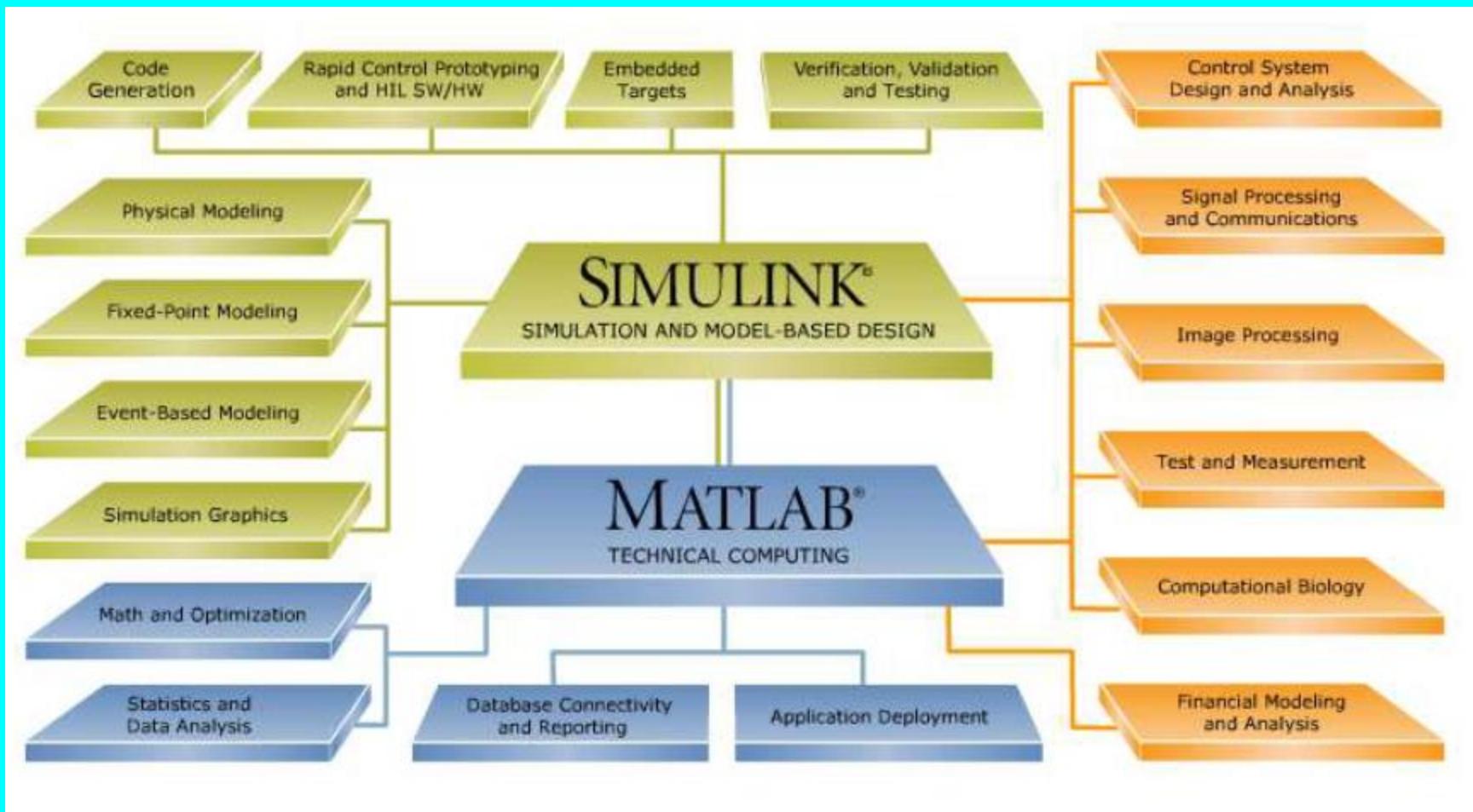
- **Jezik visokog nivoa** za tehničko računanje
- **Razvojno okruženje** za upravljanje programskim kodom, datotekama i podacima
- **Interaktivni alati** za iterativno istraživanje, dizajn i rešavanje problema
- **Biblioteka matematičkih funkcija** iz linearne algebre, statistike, Fourierove analize, filtriranja, optimizacije, numeričke integracije
- **2D i 3D grafičke funkcije** za prikaz podataka
- **Alati** za kreiranje korisničkog interfejsa
- **Funkcije** za integraciju MATLAB algoritama sa spoljašnjim aplikacijama i programskim jezicima: Fortran, C, C++, Java, COM, i Microsoft Excel

MatLab - Matrix Laboratory.

Primena

- na Univerzitetima je alat za obuku
- u industriji se upotrebljava za istraživanje i rešavanje praktičnih inženjerskih i matematičkih problema
- tipična upotreba pokriva:
 - Matematiku i numeriku
 - Razvoj algoritama
 - Modeliranje, simulaciju i razvoj prototipova
 - Analizu podataka, istraživanja i vizuelizaciju
 - Naučnu i inženjersku grafiku
 - Razvoj aplikacija, uključujući grafički korisnički interfejs
- poseduje kolekcije metoda (alata, rešenja) specifičnih za odredjene oblasti -Toolbox-ovi

MATLAB familija proizvoda



MatLab - Matrix Laboratory.

- MatLab je izvanredan proizvod kompanije MathWokrs Inc. i prevashodno je namenjen kao program za pomoć inženjerima pri njihovom radu.
- Startovanjem MatLab programa pokreće se radno okruženje. Radno okruženje svaki korisnik može podešiti prema svojim potrebama, ali podrazumevano radno okruženje sastoji se od komandnog prozora koji zauzima najveći deo samog prostora. Zatim, obično, sa leve strane se u okrivu istog prozora nalaze prikaz radnog direktorijuma i radni prostor koji prikazuje sve definisane promenljive u MatLabu.

Pregled MatLab-a

- MatLab je matematički i grafički softverski paket sa mnogobrojnim numeričkim, grafičkim i programskim mogućnostima koji ima ugrađene funkcije za obavljanje raznovrsnih operacija. Takođe postoje i alati koji se mogu dodati radi proširenja ovih funkcija (npr. za obradu signala). Postoje verzije dostupne za različite hardverske platforme, kako u profesionalnim tako i u studentskim izdanjima. U ovom radu je korišćena verzija najnovijeg MatLaba iz 2017. godine.

Pokretanje MatLab-a i njegove komande

- Podrazumevani izgled radne površine ove verzije prikazan je na slici. Pojavljuju se tri prozora - komandni prozor, prozor trenutnih direktorijuma i prozor radnog prozora. Moguće je otvoriti i dodatne prozore koji znatno olakšavaju praćenje rada u MatLab-u, poput Command History prozora koji prikazuje istoriju instrukcija koje su se izvršile.

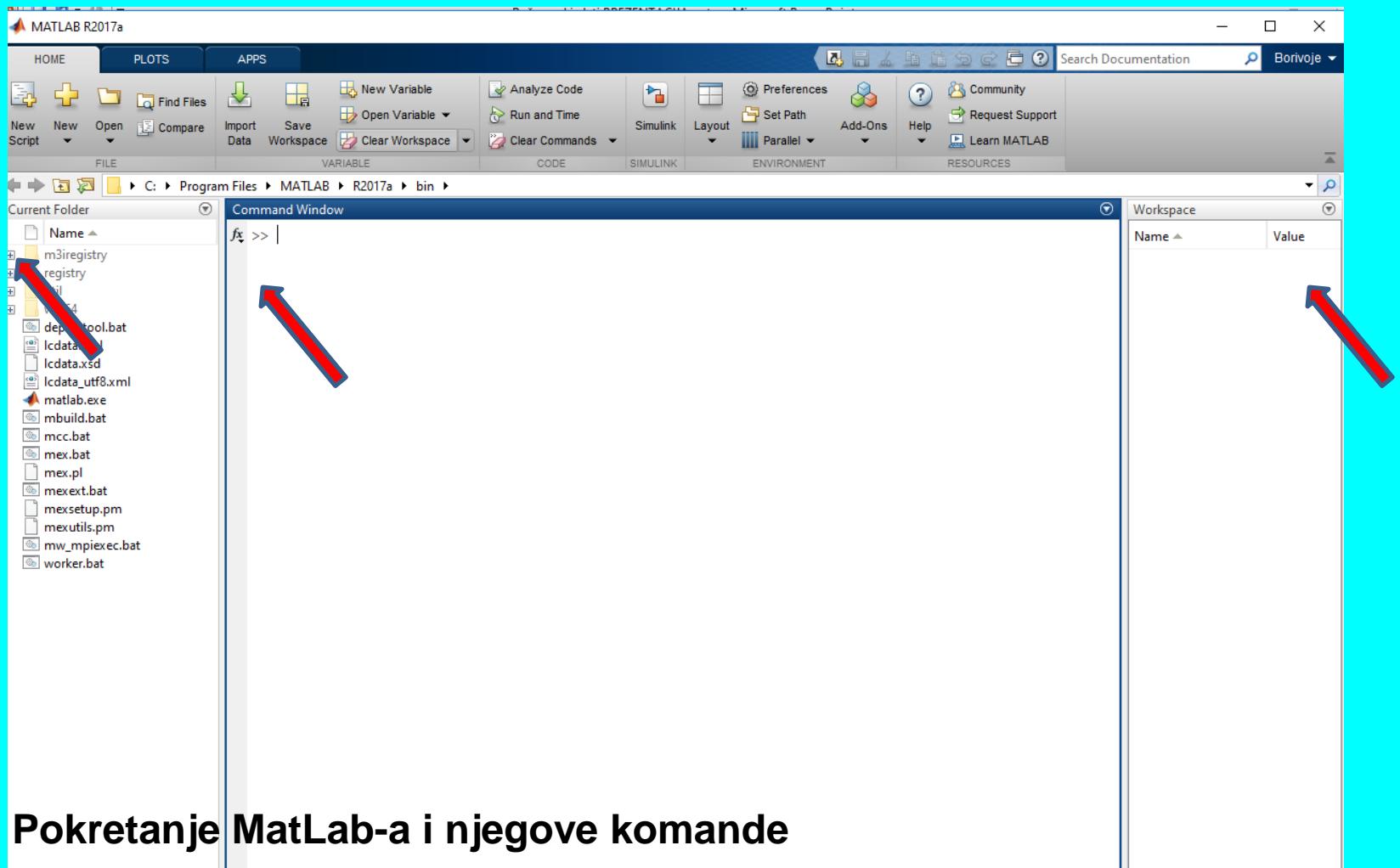
MatLab - Matrix Laboratory.

- Program MATLAB služi za rešavanje različitih matematičkih problema, i čitav niz izračunavanja i simulacija vezanih uz obradu signala, upravljanje, regulaciju i identifikaciju sistema. Prva verzija MATLAB-a, jednostavni matrični laboratorijum (*Matrix Laboratory*), napisana je krajem 1970. godine na *University of New Mexico* i *Stanford University* sa ciljem primene u matričnoj teoriji, linearnoj algebri i numeričkoj analizi. Korićten je Fortran i delovi biblioteka LINPACK i EISPACK. Početkom 80-tih se prelazi na C programske jezik uz dodavanje novih mogućnosti, i to prvenstveno u područjima obradbe signala i automatskog upravljanja. Od 1984. MATLAB je dostupan kao komercijalni proizvod kompanije MathWorks.
- Danas svojstva MATLAB-a daleko prelaze originalni “matrični laboratorijum”. Radi se o interaktivnom sistemu i programskom jeziku za opšta tehnička i naučna izračunavanja. Osim osnovnog sistema postoje i brojni programski paketi koji ga proširuju i pokrivaju gotovo sva područja inženjerske delatnosti: obradu signala i slike, 2D i 3D grafičke prikaze, automatsko upravljanje, identifikaciju sistema, statističke obrade, analizu u vremenskom i frekvencijskom domenu, simboličku matematiku i brojne druge. Jedan od važnijih paketa je SIMULINK—vizualni alat koji omogućuje simulaciju kontinualnih i diskretnih sistema pomoću funkcijskih blok dijagrama ne zahtevajući od korisnika detaljno poznavanje sintakse nekog programskog jezika.
- MATLAB je takođe zamišljen kao sistem u kome korisnik na jednostavan način može graditi svoje vlastite alate i biblioteke i modifikovati postojeće. U tu svrhu se koristi jednostavni programski jezik. Također je moguće koristiti C, Fortran, Adu ili Javu.

Osnovne mogućnosti MATLAB-a

- Svi podaci u MATLAB-u tretiraju se kao matrice čije dimenzije nije potrebno čuvati kao posebne varijable. Čak se i skalarne veličine predstavljaju kao matrice sa dimenzijom 1×1 . Svi su podaci interno zapisani u *double float* obliku (pokretni zarez dvostrukе preciznosti – 64 bita) što osigurava vrlo veliki dinamički raspon i točnost za brojne primene. Pored realnih brojeva i matrica, podržane su i kompleksni brojevi te kompleksne matrice.
- Po svojoj formi MATLAB je interaktivni jezik—interpreter, namenjen prvenstveno matričnim izračunavanjima. Po svojoj formi blizak je načinu na koji i inače zapisujemo matematičke formule, pa jedan red u MATLAB-u može zameniti stotine reda napisanih u nekom programskom jeziku opšte namene (C, C++, PASCAL, BASIC i sl.).

MatLab - Matrix Laboratory.



Pokretanje MatLab-a i njegove komande

- Podrazumevani izgled radne površine ove verzije prikazan je na slici. Pojavljuju se tri prozora - komandni prozor, prozor trenutnih direktorijuma i prozor radnog prozora. Moguće je otvoriti i dodatne prozore koji znatno olakšavaju praćenje rada u MatLab-u, poput Command History prozora koji prikazuje istoriju instrukcija koje su se izvršile.

MatLab - Matrix Laboratory.

- Umesto uobičajenog menija i traka sa alatkama, MatLab Desktop sada ima samo traku sa alatkama gde se podrazumevano prikazuju tri kartice („HOME“, „PLOTS“ i „APPS“), mada se mogu dodati i druge, uključujući „SHORTCUTS“.
- Na kartici „HOME“ nalaze se mnoge korisne funkcije koje su podeljene u funkcionalne odeljke „FILE“, „VARIABLE“, „CODE“, „SIMULINK“, „ENVIRONMENT“ i „RESOURCES“. Na primer, „ENVIRONMENT“ omogućava prilagođavanje prozora u okruženju radne površine. U svakom odeljku postoji čitav niz opcija koje korisniku pružaju mnogobrojne mogućnosti.
- **Komandni prozor – Command Window** je osnovni prozor za komunikaciju sa MatLab programom, služi za ukucavanje uputstava različitih vrsta: komandi, funkcija, izjava, matlab skripta. U komandnom prozoru MatLab se može koristiti interaktivno. Posle upita **>>** (koji se pojavljuje nakon pritiska na dugme *Enter*) se može uneti bilo koja MatLab naredba ili izraz i program će odmah odgovoriti rezultatom. Ukoliko želimo samo da nešto definišemo, bez direktnog izračunavanja rezultata, nakon naredbe je potrebno staviti simbol ; Kada se na početak reda upiše znak procenta (%), time se red označi kao komentar. Tako označeni redovi se ne izvršavaju.
- **Zaustavljanje izvršenja komandi** – Pre nego što počnete, trebali bi te znati da kada ste spremni napustiti MATLAB, možete upisati **quit** u taj prozor, ovako,
>> quit
i pritisnite Enter. MATLAB će tada zatvoriti sve otvorene prozore i odustati od početog posla. Za zaustavljanje izvršavanja MATLAB® skripta, pritisnите **Ctrl+C** ili **Ctrl+Break**. Ctrl+C ne zaustavlja uvek izvršavanje za datoteke koje se pokreću dugo vremena.

MatLab - Matrix Laboratory.

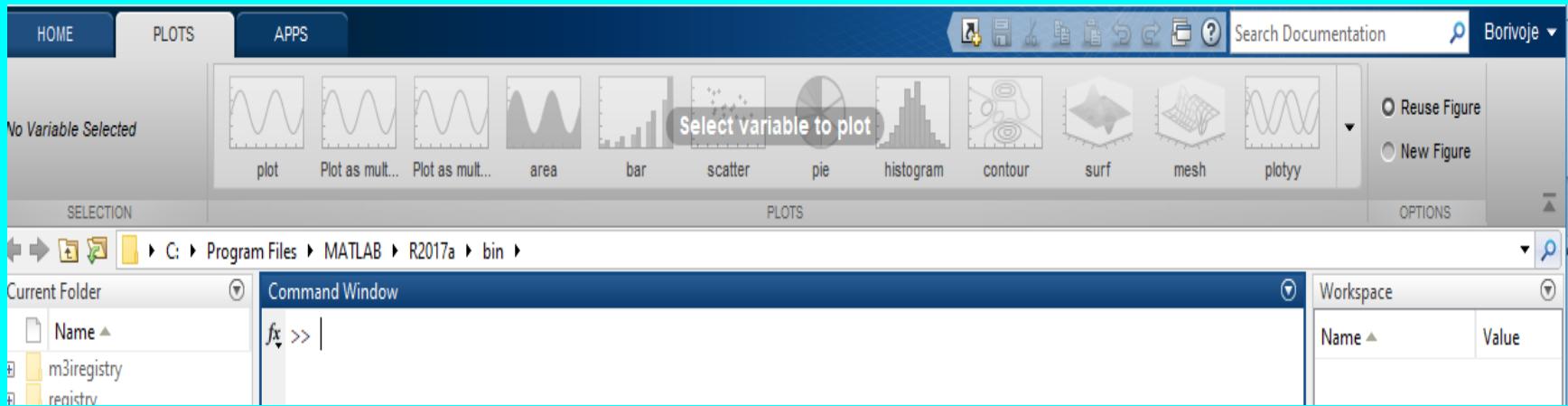
- **Prozor sa prethodnim komandama** - Command History prikazuje komande prethodno unete u komandnom prozoru. Prikazuju se i komande iz prethodnih sesija MatLab-a. Svaka komanda iz prozora sa prethodnim komandama može biti ponovo upotrebljena u komandnom prozoru. Osim toga, komande se mogu odvući u komandni prozor, tamo se po potrebi izmeniti i zatim izvršiti. Spisak komandi u prozoru s prethodnim komandama se može skraćivati - opcijom *Delete Section*.
- **Prozor direktorijuma** vodi računa o kreiranju i čuvanju trenutnih datoteka i funkcija programa. MatLab koristi nekoliko vrsta datoteka koje omogućavaju da čuvanje programa, podataka i rezultate sesija. Datoteke MatLab funkcija i programske datoteke čuvaju se sa ekstenzijom .m i nazivaju se m-datotekama. Mat-datoteke imaju ekstenziju .mat i koriste se za čuvanje imena i vrednosti promenljivih stvorenih tokom MatLab sesije. Budući da se radi o ASCII datotekama, M-datoteke se mogu kreirati koristeći gotovo bilo koji procesor teksta. MAT-datoteke su binarne datoteke koje čita samo softver koji ih je stvorio. MAT-datoteke sadrže mašinski potpis koji im omogućava prenos između tipova različitih mašina. Treća vrsta datoteke koja se često koristiti jeste ASCII datoteka podataka. MatLab se koristi za analizu podataka skladištenih u takvoj datoteci koju je kreirao npr. program za proračunske tablice, program za obradu teksta ili laboratorijski sistem za prikupljanje podataka.

MATLAB-ovi prozori

Prozor	Namena
Komandni prozor (Command Window)	Glavni prozor za unošenje promenljivih i izvršavanje programa.
Grafički prozor (Figure)	Sadrži rezultate grafičkih komandi.
Prozor za pisanje programa (Editor)	Za pisanje skript datoteka i funkcijskih datoteka, te za otkrivanje i otklanjanje grešaka u njima.
Prozor sistema za pomoć (Help)	Sadrži pomoćne informacije.
Prozor za pokretanje (Launch Pad Window)	Omogućava pristupanje alatkama, demonstracijama i dokumentaciji.
Prozor s prethodnim komandama (Command History)	Sadrži komande unesene u komandnom prozoru.
Prozor radnog prostora (Workspace Window)	Sadrži podatke o svim upotrebljenim promenljivama.
Prozor tekućeg direktorijuma (Current Directory)	Prikazuje datoteke u tekućem direktorijumu.

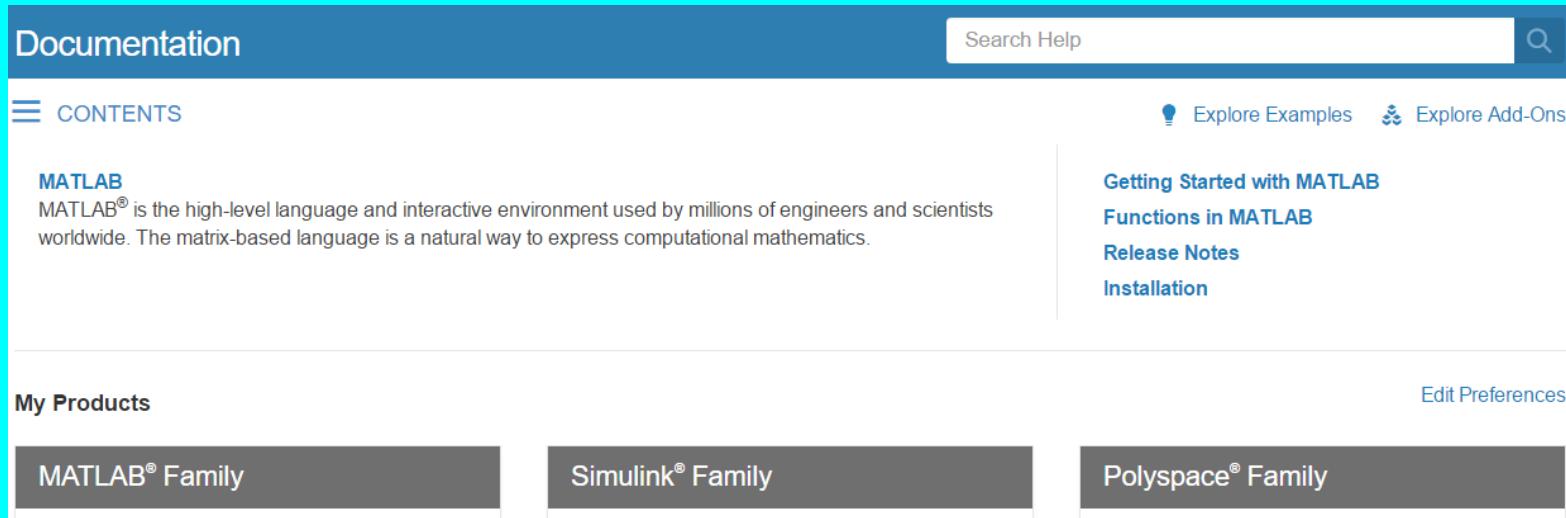
MatLab - Matrix Laboratory.

- **Grafički prozor** (Figure) je prozor koji se automatski otvara kada se izvršavaju grafičke komande i sadrži grafike nacrtane pomoću tih komandi. Biće detaljnije objašnjen u sekciji posvećenoj funkcijama za rad sa figurama.

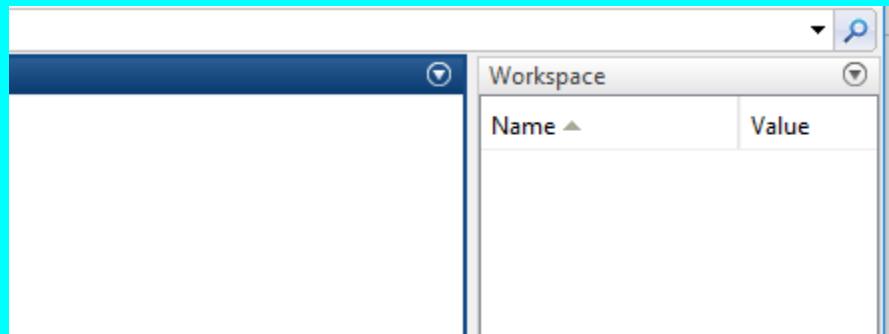


MatLab - Matrix Laboratory.

- **Prozor sistema za pomoć** - Help sadrži ugrađenu pomoć za rad sa MatLab programima. Interaktivan je i služi za dobijanje pomoćnih informacija o bilo kojoj komponenti ili svojstvu MatLab-a. Opcije su poređane po oblastima, tako da je olakšano pronalaženje rešenja problema.



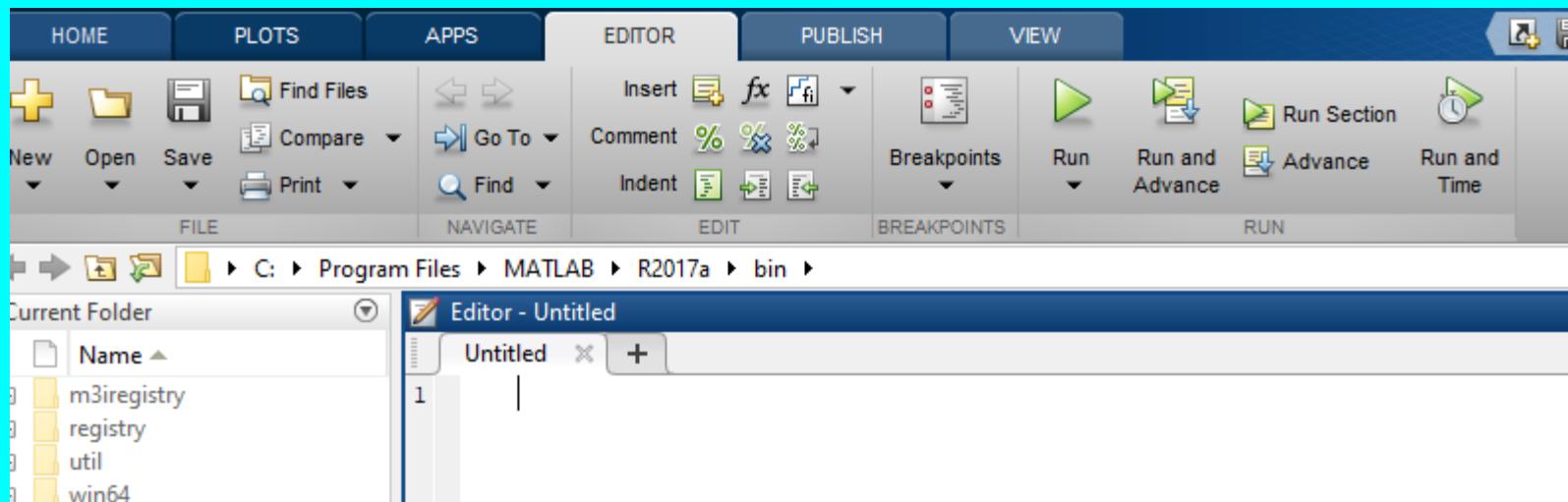
- **Prozor radnog okruženja** - Workspace sadrži podatke o svim upotrebljenim varijablama. Osim imena samih varijabli i njihovih vrednosti moguće je videti njihovu veličinu i tip, odnosno klasu.



MatLab - Matrix Laboratory.

Skripte - Editor

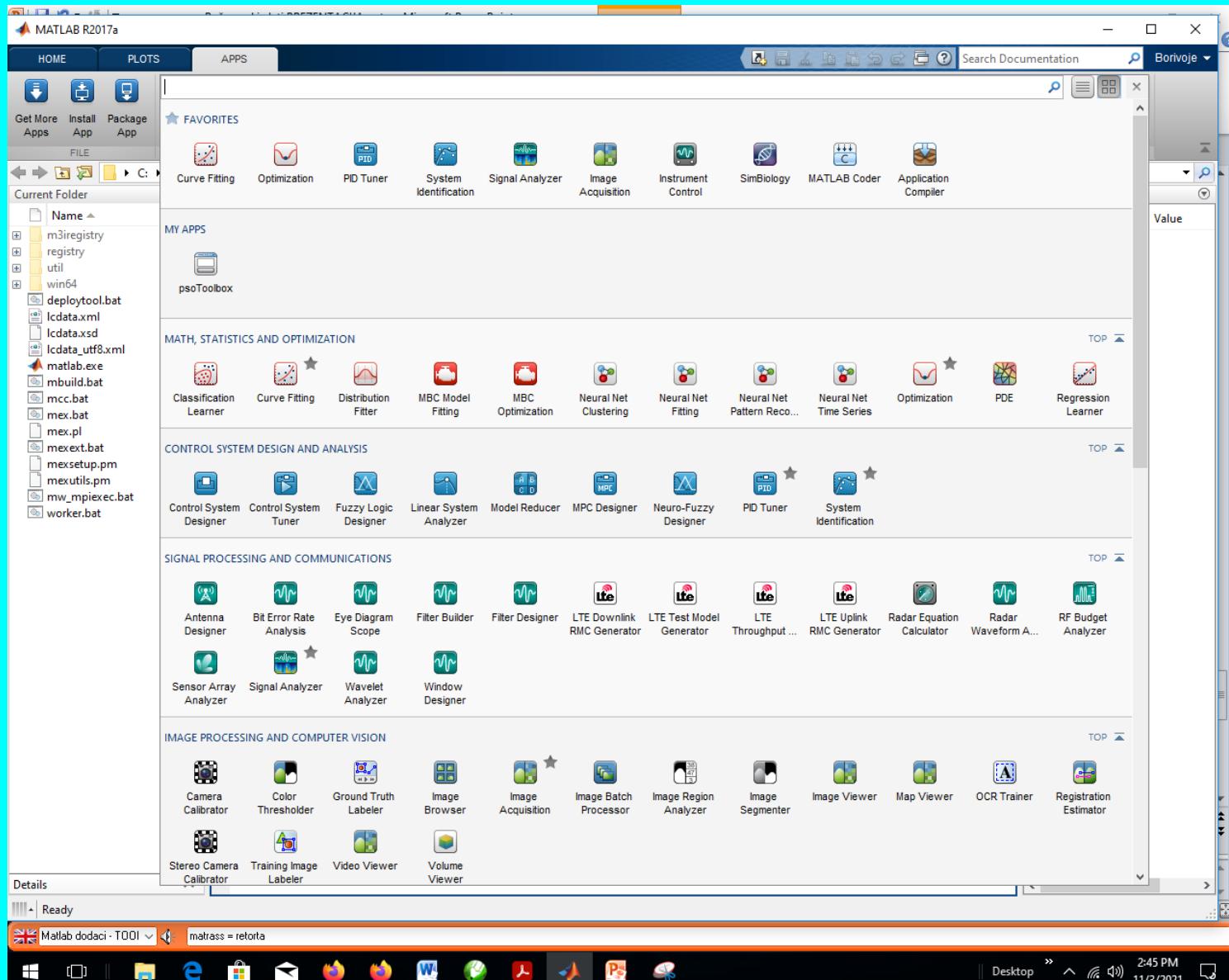
- U MatLab-u se mogu kreirati skripte pokretanjem funkcije *New Script*, i one, kao što je već pomenuto, imaju ekstenziju **.m**. Ukoliko skript datoteka ima komandu koja generiše izlaz (npr. dodeljuje vrednost promenljivoj u iskazu na čijem kraju ne стоји тачка и зarez), izlaz se prikazuje u komandnom prozoru. Pokreću se opcijom *Run*.
- Nakon kreiranja programa, takođe je moguće kreirati skriptu uz pomoć funkcije *Create Live Script*. Pokretanjem ove funkcije otvara se editor sa skriptom koja sadrži linije koda koje se prethodno obeležavaju (npr. putem Command History prozora). Tako kreiranu skriptu možemo editovati, deliti u sekcije, dodavati tekst, naslove, slike, jednačine itd.



- Skripte su ASCII datoteke koje sadrže blok naredbi MATLAB jezika.

MatLab - Matrix Laboratory.

Matlab dodaci - TOOLBOX



MatLab - Matrix Laboratory.

Matlab dodaci - TOOLBOX

Aerospace Toolbox

Bioinformatics Toolbox

Communications Toolbox

Control System Toolbox

Curve Fitting Toolbox

Data Acquisition Toolbox

Genetic Algorithm and Direct Search
Toolbox

Image Processing Toolbox

Instrument Control Toolbox

Model-Based Calibration Toolbox

Model Predictive Control Toolbox

Neural Network Toolbox

Optimization Toolbox

Partial Differential Equation (PDE)
Toolbox

Robust Control Toolbox

Signal Processing Toolbox

Statistics Toolbox

Symbolic Math Toolbox

System Identification Toolbox

Matlab-Numerički tipovi podataka

Tip	Opseg	M-funkcija
Celobrojan		
Označen 8-bit integer	- 2^7 to 2^7-1	int8
Označen 16-bit integer	- 2^{15} to $2^{15}-1$	int16
Označen 32-bit integer	- 2^{31} to $2^{31}-1$	int32
Označen 64-bit integer	- 2^{63} to $2^{63}-1$	int64
Neoznačen 8-bit integer	0 to 2^8-1	uint8
Neoznačen 16-bit integer	0 to $2^{16}-1$	uint16
Neoznačen 32-bit integer	0 to $2^{32}-1$	uint32
Neoznačen 64-bit integer	0 to $2^{64}-1$	uint64
U pokretnom zarezu		
Jedostruke preciznosti 32-bit	- 3.4×10^{-38} to 3.4×10^{38}	single
Dvostruke preciznosti 64-bit	IEEE Standard 754	double

```
» x = int16(325);
```

```
» x = 325.5;
```

```
» int16(fix(x))
```

```
ans =
```

```
325
```

```
» int16(325) * 4.39
```

```
ans =
```

```
1427
```

```
» uint8('Danas je ...')
```

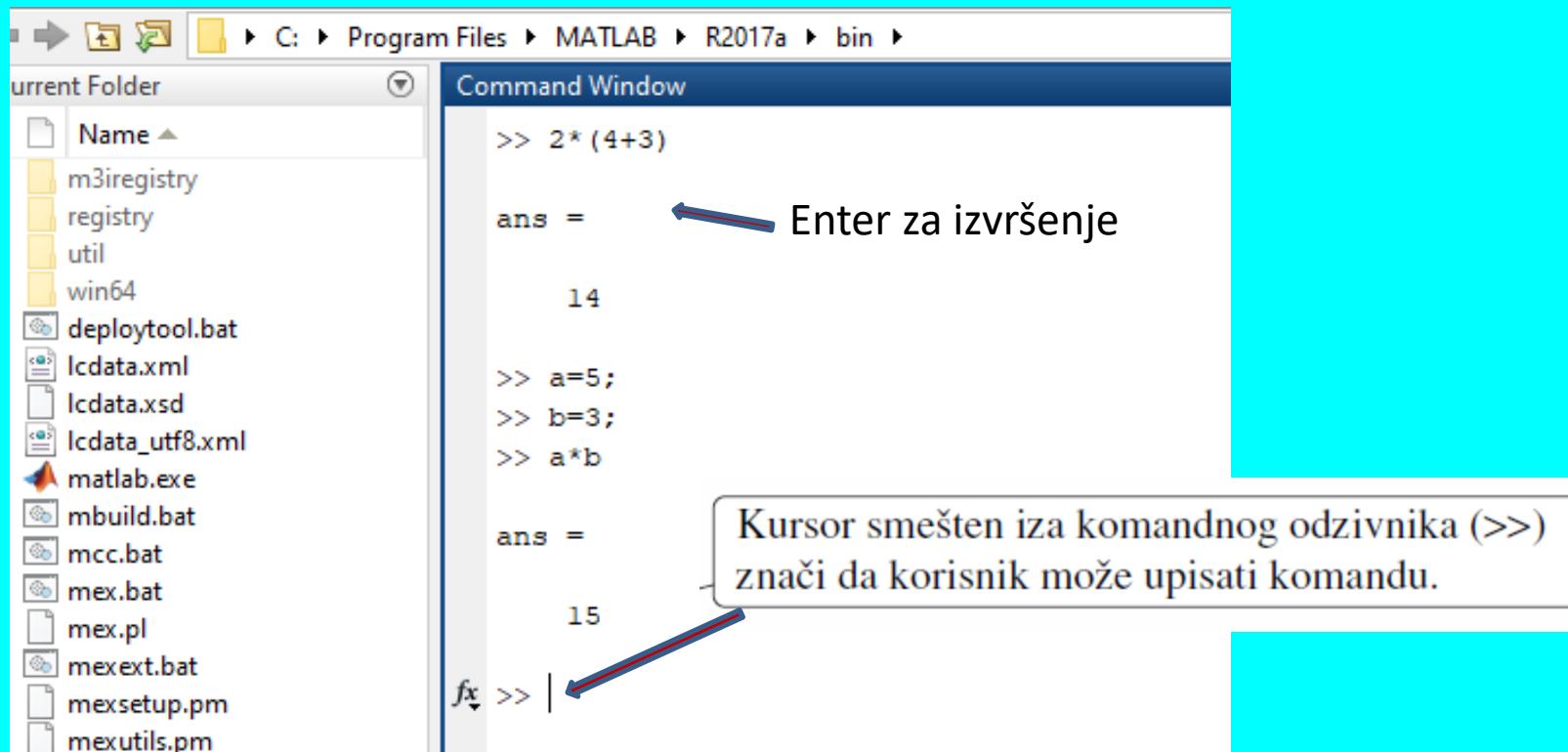
```
ans =
```

```
68 97 110 97 115 32 ...
106 101 32 46 46 46
```

MatLab - Matrix Laboratory.

RAD U KOMANDNOM PROZORU

- Komandni (Command) prozor je glavni MATLAB-ov prozor i služi za izvršavanje komandi, otvaranje prozora, pokretanje programa koje je napisao korisnik i upravljaće MATLAB-om. Primer komandnog prozora, sa nekoliko jednostavnih komandi koje ćemo objasniti :



The screenshot shows the MATLAB interface with the Command Window active. The window title is "Command Window". Inside, the following commands and their results are displayed:

```
>> 2*(4+3)
ans =
    14

>> a=5;
>> b=3;
>> a*b

ans =
    15
```

A red arrow points from the text "Enter za izvršenje" to the carriage return after the first command. A callout bubble with a red border contains the text: "Kursor smešten iza komandnog odzivnika (>>) znači da korisnik može upisati komandu."

MatLab - Matrix Laboratory.

Korišćenje MATLAB-a kao kalkulatora

- MATLAB je najjednostavnije koristiti kao kalkulator. Tako ga koristite kada u komandni prozor upisujete matematičke izraze i pritiskate **Enter**. MATLAB će izračunati izraz, napisati `ans =` i prikazati numerički rezultat u sledećem redu.

```
>>7+8/2           ← Upisujemo i pritiskamo Enter.  
ans =  
    11  
>>(7+8)/2         ← Upisujemo i pritiskamo Enter.  
ans =  
    7.5000  
>>4+5/3+2  
ans =  
    7.6667           ^ Na stepen  
>>27^(1/3)+32^0.2  
ans =  
    5  
>>27^1/3+32^0.2  
ans =  
    11  
>>0.7854-(0.7854)^3/(1*2*3)+0.785^5/(1*2*3*4*5*...  
-(0.785)^7/(1*2*3*4*5*6*7)           ← Upisujemo tri tačke ... (i pritiskamo Enter) da bismo izraz nastavili u sledećem redu.  
ans =  
    0.7071  
>>           Poslednji izraz sadrži prva četiri člana Tejlorovog reda za  $\sin(\pi/4)$ .
```

Prvo se izračunava $8/2$.

Prvo se izračunava $7+8$.

Prvo se izračunava $5/3$.

Prvo se izračunava $1/3$, zatim $27^{(1/3)}$ i $32^0.2$; poslednje je $+$.

Prvo se izračunava 27^1 i $32^0.2$; zatim $/3$; poslednje je $+$.

MatLab - Matrix Laboratory.

Napomene o radu u komandnom prozoru:

- Da biste upisali komandu, kurzor mora biti neposredno iza komandnog odzivnika (>>).
- Upisana komanda biće izvršena kada pritisnete taster **Enter**. Međutim, izvršava se samo poslednja upisana komanda. Sve prethodno izvršene komande ostaju nepromenjene.
- U isti red možete upisati više komandi ako ih razdvojite zarezom. Kada pritisnete **Enter** komande će biti izvršene redom, sa leva na desno.
- Ne možete se vratiti u prethodni red unutar komandnog prozora, izmeniti nešto i zatim ponovo izvršiti tako izmenjenu komandu.
- Pritisnite taster sa strelicom nagore (↑) da biste iza komandnog odzivnika (>>) prikazali prethodnu komandu. Tako napisanu komandu možete izmeniti (ako treba) i izvršiti. Strelicom nadole (↓) redom prikazujete sve prethodno upisane komande.
- Ako je komanda preduga da bi stala u jedan red, upišite tri tačke (...) i pritisnite **Enter**. Nastavak komande pišite u sledećem redu. Komanda se može protezati na više redova i sadržati do 4096 znakova.

MatLab - Matrix Laboratory.

Napomene o radu u komandnom prozoru:

- **Tačka i zarez (;)**: Komanda se izvršava kada je upišete u komandni prozor i pritisnute taster **Enter**. i rezultat komande se prikazuje u komandnom prozoru. Ukoliko na kraj komande upišete tačku i zarez (;), čen rezultat se neće prikazati. To je podesno kada je rezultat očigledan ili poznat, a i kada je veoma velik. Ukoliko u isti red upišete više komandi, njihovi rezultati se neće prikazati ako ih razdvojite znakovima tačka i zarez umesto zarezima.
- **Znak procenta (%)**: Kada na početak reda upišete znak procenta (%), time ste red označili kao komentar. Tako označeni redovi se ne izvršavaju. Ukoliko znak % i odgovarajući tekst (komentar) upišete iza komande (u isti red), to neće uticati na izvršavače komande. Obično nema potrebe da se u komandnom prozoru pišu komentari. Sa druge strane, opisi i objašnjenja se često dodaju programima u vidu komentara.
- **Komanda clc**: Komanda clc (upišite clc i pritisnite **Enter**) briše sadržaj komandnog prozora. Posle dužeg rada u komandnom prozoru, n jegov sadržaj može postati predugačak. Komanda clc briše sadržaj komandnog prozora i ne čini ništa više. Na primer, sve prethodno definisane promenljive i dalje postoje i mogu se upotrebljavati. Prethodno upisane komande i dalje možete prikazati kad pritisnete taster sa strelicom nagore (↑).

MatLab - Matrix Laboratory.

➤ Ulaz sa tastature

Neki MATLAB programi zahtevaju unos od korisnika putem tastature. U tipičnoj situaciji, korisnik daje ulaz kao odgovor na poruku koju ispisuje program tražeći unos od korisnika. Kako smo saznali, ova poruka se zove prompt. Već smo videli da MATLAB-ov komandni prozor koristi vrlo jednostavan prompt: >>. U programima koje čete napisati, prompt može biti bilo koji niz znakova koji vam se svidja. Može se kretati od jednog ili dva znaka do izraza ili cele rečenice, ali bez obzira na oblik upita, korisnik mora odgovoriti na njega upisivanjem nečega i pritiskom na tipku Enter. Sve se to postiže pomoću funkcije unosa. Na primer, pretpostavimo da M-datoteka uključuje naredbu:

```
fave = input('Type your favorite vector:');
```

- Kada se komanda izvrši, program će štampati sledeće u komandnom prozoru,

Type your favorite vector: |

MatLab - Matrix Laboratory.

Formatirani izlaz

- Rad programa nikome ne koristi ako ne izloži svoje rezultate u spoljni svet. Do sada smo iskoristili činjenicu da je komanda nakon koje ne sledi tačka-zarez automatski ispisuje bilo koju vrednost računa u komandni prozor. Takodje smo videli da je moguće promeniti način na koji se izračunati rezultati prikazuju koriššenjem formata naredba.
- Ova metoda komunikacije je izvediva za jednostavne programe, ali MATLAB pruža daleko sofisticiraniji način za ispis informacija sredstava funkcije, pozajmljene iz jezika C i C++ (što smo nadam se već naučili), tzv **fprintf**. Njegov naziv otprilike znači "formatirani ispis u datoteku,,.

```
fprintf('%d pomnoženo sa %.2f\nTotal Z = %5.2f\n',x,y,z)
```

ans:

3 pomnoženo sa 2.71

Total Z = 8.13

MatLab - Matrix Laboratory.

FORMAT SPEC	DESCRIPTION
c	single character
d	decimal notation (but no decimal if integral)
e	exponential notation
E	exponential notation with an upper case E
f	fixed-point notation
g	shorter of e or f
G	shorter of e or f but with an upper case E
o	unsigned octal notation
s	string
u	unsigned decimal notation
x	hexadecimal notation
X	hexadecimal notation with uppercase

Rad sa promenljivama

Za rad sa promenljivama iz radnog prostora na raspolaganju su nam komande prikazane u tabeli :

Komande za rad sa promenljivama radnog prostora

Komanda	Rezultat
<i>who</i>	Prikazuje listu promenljivih iz memorije
<i>whos</i>	Prikazuje detaljnu listu promenljivih iz memorije
<i>clear x y</i>	Brise promenljive <i>x</i> i <i>y</i> iz memorije
<i>clear</i>	Brise sve promenljive iz memorije radnog prostora

R.b.	Pravilo
1.	Ime promenljive mora poceti slovom
2.	Ime promenljive moze sadrzati slova, brojeve i donju crtu (<i>_</i>). Ukupan broj znakova u imenu zavisi od verzije Matlab-a (U verziji R2009b Matlab koristi prva 63 znaka u imenu promenljive).
3.	U Matlabu postoji razlika izmedju malih i velikih slova. Promenljiva sa imenom <i>x</i> se razlikuje od promenljive sa imenom <i>X</i> . Slicno, promenljiva <i>sila</i> nije isto sto i promenljiva <i>Sila</i> .
4.	Rezervisane reci (npr. <i>break</i> , <i>case</i> , ...) se ne mogu koristiti za imena promenljivih. Lista rezervisanih reci Matlab-a se moze dobiti komandom <i>iskeyword</i> .
5.	Pri davanju imena promenljivama treba izbegavati imena ugradjenih funkcija i imena unapred definisanih promenljivih.

Aritmeticki izrazi

- Pri pisanju aritmetickih izraza pored brojeva i promenljivih koristimo i aritmeticke operatore. U tabeli data je lista aritmetickih operatora.

Operacija	Operator	Primer
Sabiranje	+	$x+7$
Oduzimanje	-	$11-y$
Mnozenje	*	$x*y$
Deljenje zdesna	/	$8/3$
Deljenje sleva	\	$a\b = b/a$ (rezultat deljenja sleva je reciprocna vrednost deljenja sdesna)
Stepenovanje	^	2^3 (isto je sto i 2^3)

Neke od ugradjenih matematičkih funkcija u Matlabu

Funkcija	Opis	Primer
<code>sin(x)</code>	Sinus argumenta x datog u radijanima.	<code>sin(pi/3) :</code> <code>ans=0.8660</code>
<code>sind(x)</code>	Sinus argumenta x datog u stepenima.	<code>sin(60) :</code> <code>ans=0.8660</code>
<code>asin(x)</code>	Inverzna funkcija od <code>sin(x)</code> . Rezultat je u radijanima.	<code>asin(0.866) :</code> <code>ans=1.0471</code>
<code>asind(x)</code>	Inverzna funkcija od <code>sin(x)</code> . Rezultat je u stepenima.	<code>asind(0.866) :</code> <code>ans=59.9971</code>
<code>exp(x)</code>	Vrednost izraza e^x . e je osnova prirodnog algoritma.	<code>exp(1) :</code> <code>ans= 2.7183</code>
<code>log(x)</code>	Prirodni logaritam broja x .	<code>log(2.7183)</code> <code>ans=1.0000</code>
<code>log10(x)</code>	Logaritam broja x za osnovu 10.	<code>log10(10) :</code> <code>ans=1</code>
<code>sqrt(x)</code>	Kvadratni koren broja x .	<code>sqrt(9) :</code> <code>ans=3</code>
<code>abs(x)</code>	Apsolutna vrednost broja x .	<code>abs(-3) :</code> <code>ans=3</code>
<code>round(x)</code>	Zaokružuje vrednost broja x na najbliži ceo broj.	<code>round(7.8) :</code> <code>ans=8</code>
<code>fix(x)</code>	Zaokružuje vrednost broja x na ceo broj blizi nuli.	<code>fix(7.8) :</code> <code>ans=7</code>
<code>rem(x,y)</code>	Ostatak celobrojnog deljenja x sa y .	<code>rem(10,4) :</code> <code>ans=2</code>
<code>sign(x)</code>	Funkcija znaka. Ako je $x>0$ vrednost funkcije je 1. Kada je	<code>sign(2.5) :</code>

MatLab - Matrix Laboratory.

TIPOVI PODATAKA

- Numerički
- Karakteri i stringovi
- Nizovi ćelija sa stringovima
- Logički
- Simbolički
- Kategorijski nizovi
- Datum i vreme
- Tabele
- Strukture
- Vremenske serije
- Tip podataka za identifikaciju

MatLab - Matrix Laboratory.

NUMERIČKI PODACI

Celi brojevi:

357, -82475

Realni brojevi:

3, -99, 0.0001, -9.639825,

Brojevi sa pokretnim zarezom

1.62e-20, -10.34e200

opseg brojeva:

$[-1.7977 \cdot 10^{308}, -2.2251 \cdot 10^{-308}] \cup [2.2251 \cdot 10^{-308}, 1.7977 \cdot 10^{308}]$

Kompleksni brojevi:

-4+7i 8i -9j -9*j 1.23-12e-4j

MatLab - Matrix Laboratory.

KARAKTERI I STRINGOVI

Kreiranje stringa spajanjem više karaktera

'Zdravo narode'

Kreiranje pravougaonog niza karaktera (fiksna dužina)

['Mitrović M. Petar '; ...

'Hemijsko inzenjerstvo'; ...

'3 god. ']

NIZOVI ĆELIJA SA STRINGOVIMA (dužina je promenljiva)

{'Mitrović M. Petar'; ...

'Hemijsko inzenjerstvo'; ...

'3 god.'}

Znakovna promenljiva

Znakovna promenljiva se formira pomocu naredbe dodeljivanja:

- >> ulica='dositejeva 19'
- Znakovni niz se formira koriscenjem jednostrukih navodnika

```
>> Adresa=['Dositejeva 19';
'Kraljevo 36000']
```

```
ulica =
dositejeva 19
• Elementima niza pristupamo pomocu indeksa:
```

- >> ulica(1)

```
ans =
d
```

ili

- >> ulica(2:6)

```
ans =
osite
```

- Naredbom dodeljivanja menjamo vrednost clanova niza:
- >> ulica(1)='D'

```
ulica =
Dositejeva 19
```

Izlaz:
Adresa =

2×14 char array

'Dositejeva 19 '
'Kraljevo 36000'

>>

MatLab - Matrix Laboratory.

LOGIČKI PODACI

1 (true)

0 (false)

SIMBOLIČKI PODACI

`sym(1/3)` ---> umesto 0.333... daje simboličku vrednost [1/3]

`sym(sqrt(2))` ---> umest 1.41... daje simboličku vrednost [2^(1/2)]

MatLab - Matrix Laboratory.

PROMENLJIVE

Tretiraju se kao matrice odgovarajućih dimenzija:

- skalarne promenljive,
- vektorske promenljive (jednodimenzione),
- matrične promenljive (dvodimenzione),
- trodimenzione promenljive

Svakoj promenljivoj se mora dodeliti odgovarajuća vrednost.

a=3

A=5;

s=1415.2381;

d=45, E=31

MatLab - Matrix Laboratory.

PODELA PROMENLJIVIH

Prema sadržaju podataka koje primaju:

Numeričke

- Realne a=23.619
- Kompleksne z=64 + j*13

Stringovi

Naziv_institucije='Tehnoloski fakultet'

Simboličke

syms x1 x2

Nizovi ćelija sa string. x={'Mihajla Pupina', '24', 11000, 'Beograd'}

Prema vidljivosti:

- Lokalne promenljive u funkc. se brišu nakon završetka funkc.
- Globalne promenljive ostaju važeće i nakon izlaska iz funkcije

Prema izvoru nastanka:

- Interne generiše ih Matlab
- Eksterne generiše ih korisnik ili program

MatLab - Matrix Laboratory.

Interne promenljive:

Naziv	Značenje
pi	3.1415...
i ili j	imaginarna jedinica
Inf	beskonačna vrednost, ∞ , delenje nulom ne prekida program već nastaje vrednost Inf
NaN	neodređena vrednost (<i>not a number</i>) NaN je rezultat neodređenih izraza Inf / Inf ili 0/0
eps	podrazumevana tačnost
ans	“privremena” promenljiva
...	...

MatLab - Matrix Laboratory.

Eksterne promenljive:

- naziv promenljive čine slova i brojevi
- mora početi slovom
- prvih 31 znakova se uzima u obzir
- razlikuju se mala i velika slova (npr. a i A su dve promenljive)

Primer.

temperatura

Temperatura

Tmp1

T_1

MatLab - Matrix Laboratory.

Dodeljivanje vrednosti promenljivoj

Primer. a=23, b=157.491, T='tekst'

Brisanje promenljivih naredbom clear

Naziv komande	Značenje
<code>clear</code>	briše sve promenljive iz workspace-a
<code>clear a, b, c</code>	briše promenljive a, b i c iz workspace-a
<code>clear global</code>	briše globalne promenljive iz workspace-a
<code>clear d*</code>	briše sve promenljive iz workspace-a čije ime počinje na slovo "d"

Brisanje promenljivih u prozoru workspace-a

MatLab - Matrix Laboratory.

Snimanje promenljivih

`save ime_datoteke`

`save ime_datoteke var1,var2,var3`

`save ime_datoteke var1,var2,var3 –ascii`

Učitavanje promenljivih

`load ime_datoteke`

MatLab - Matrix Laboratory.

OPERACIJE MATLABA

- Aritmetički operatori
- Relacijski operatori
- Logički operatori
- Skripte i funkcije
-
- Kontrola toka programa

MatLab - Matrix Laboratory.

ARITMETIČKI OPERATORI

prio-ritet	opera-tor	opis
1	()	Grupisanje
2	'	Transponovanje, konjugacija
	:	Transponovanje bez konjugacije
3	^	Stepenovanje
	.^	Pojedinačno stepenovanje elemenata
4	*	Matrično množenje
	.*	Množenje između elemenata matrica
	/	Desno deljenje ($X/Y = X*Y^{-1}$)
	\	Levo deljenje ($X\backslash Y = X^{-1}*Y$)
	.	Deljenje među elementima matrica
5	+, -	Sabiranje, oduzimanje

Primeri.

a=3+2, A=5*13;

s=1415.238/2.12,

d=45^11, E=31*d,

w=(a+A)/(1+2*d)

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

B=[1 4 8];

C=A*B',

D=A.^2,

F=A\B'

MatLab - Matrix Laboratory.

RELACIJSKI OPERATORI		
Operator	Opis	Primer
<	Manje	$a < b$
\leq	Manje ili jednako	$a \leq b$
>	Veće	$a > b$
\geq	Veće ili jednako	$a \geq b$
$=$	Jednako	$a == b$
$\sim=$	Različito	$a \sim= b$

A=2;
B=3;
C=A<=B

C =
1

C=A \sim =B
C =
0

Daju vrednost „1“ ukoliko je relacija tačna, odnosno „0“ ukoliko je relacija netačna.

MatLab - Matrix Laboratory.

LOGIČKI OPERATORI

Opearator	Opis	Primer
&	Logičko I – Ako A i B imaju vrednost true (1), rezultat je true , u suprotnom rezultat je false (0).	A&B
	Logičko ILI – Ako jedan ili oba operanda imaju vrednost true (1), rezultat je true (1), u suprotnom rezultat je false (0)	A B
~	Logički komplement – daje suprotnu vrednost operanda	~A
xor	Logičko ekskluzivno ILI	xor(A,B)

Kod logičkih operatora, numerička vrednost „0“ se smatra za logičku nulu, dok se ostale numeričke vrednosti smatraju logičkom jedinicom.

```
-3 & 0  ---> 1 & 0      4< -1 | 5> 0  ---> 0 | 1      ~4 & 0+8*~(4|0)  ---> 0 & 0 + 8*0  
ans =                                ans =                                ans =  
0                                     1                                     0
```

MatLab - Matrix Laboratory.

SKRIPTE I FUNKCIJE

MATLAB ima i mogućnost razvoja programa u vlastitom programskom jeziku.

Programi se skladište u tzv. "m" datoteke.

Razlikujemo dve vrste "m" datoteka:

- Skripte
- Funkcije

Skripta - skup naredbi uskladištenih u „m“ fajl koji se izvršavaju kao da su kucane u komandnom prozoru.

Funkcija - skup naredbi uskladištenih u „m“ fajl sa lokalnim važenjem promenljivih (**function** zaglavlje)

MatLab - Matrix Laboratory.

SKRIPTE

Skripte su ASCII datoteke koje sadrže blok naredbi MATLAB-a.

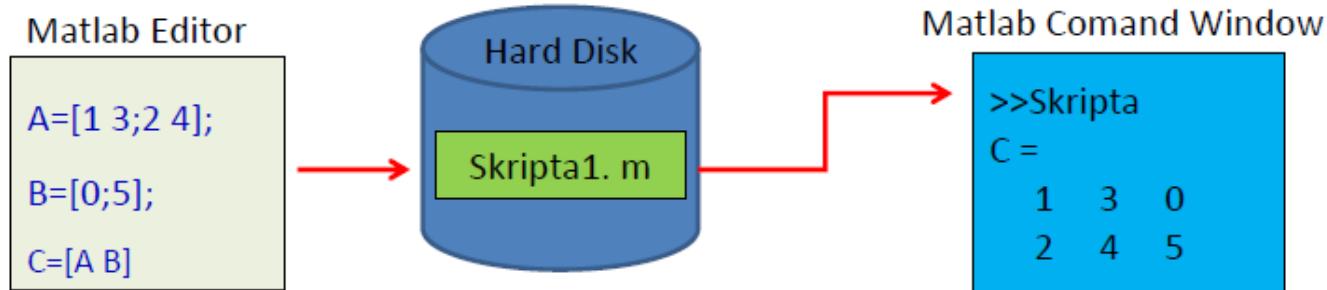
Unose se pomoću posebnog editora i snimaju kao datoteke tipa “.m”.

Pozivaju se sa komandne linije ili iz neke druge skripte navođenjem njihovog imena bez “.m”

Skripte nemaju parametre.

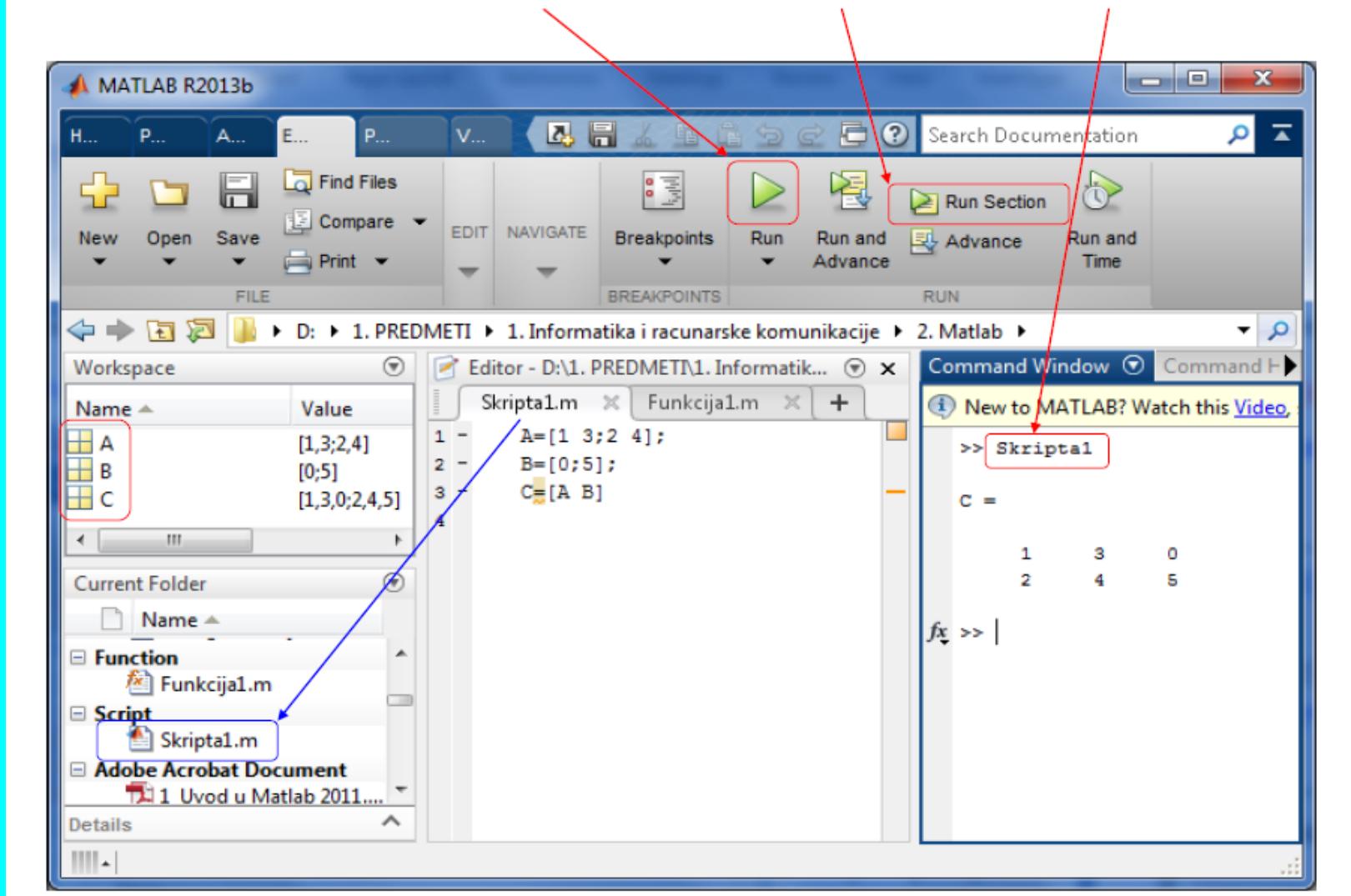
Skripte preuzimaju promenljive iz radnog okruženja i skladište ih u random okruženju.

Primer.



MatLab - Matrix Laboratory.

Primer. Napisati skriptu „Skripta1.m“ za rešavanje problema iz prethodnog primera.
Izvršiti skriptu koristeći: a) dugme Run ili b) Run Section, ili sa c) komandne linije.



MatLab - Matrix Laboratory.

FUNKCIJE

To su "m" datoteke oblika:

```
function [ ip1, ip2, ... ipn ] = ime ( up1, up2, ... upm )  
    <niz naredbi>  
end
```

gde su:

function – službena reč koja ukazuje na početak definicije funkcije
ime – naziv funkcije - odgovara imenu „m“ fajla u koji se funkcija snima
up1,up2,...upm - ulazni parametri (argumenti)
ip1,ip2,...ipn - izlazni parametri (argumenti)
niz naredbi – naredbe koje čine funkciju
end – službena reč koja ukazuje na kraj definicije funkcije

Funkcija se može pozvati sa komandne linije, iz skripte ili iz druge funkcije.

Prilikom poziva funkcije ne moraju se navoditi svi parametri; funkcijama se dozvojava da imaju promenljivi broj ulaznih i izlaznih argumenata.

Sve promenljive koje se definišu u funkciji su lokalnog karaktera; izlaskom iz funkcije one se brišu.

MatLab - Matrix Laboratory.

Vrste funkcija

Interne funkcije

Funkcije unutar Toolboxova

Funkcije definisane od strane korisnika

Poziv funkcija

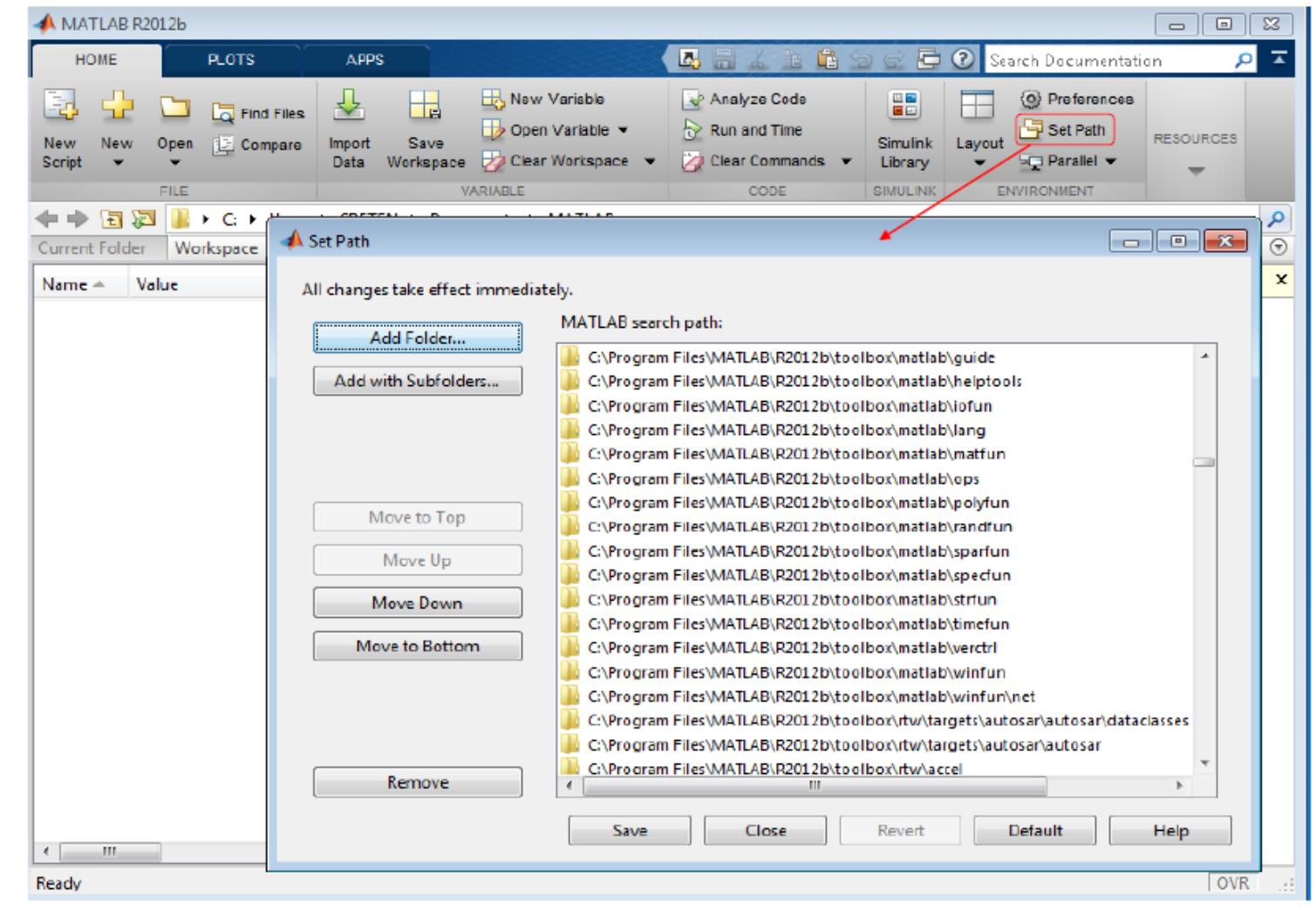
```
[I_arg1,I_arg2,...,I_argn ] = ime_funkcije (U_arg1,U_arg2,...,U_argn);
```

Funkcija mora da se nalazi u tekućem direktoriju ili u nekom od direktorijuma koji su dodati u Path.

Direktoriji u kojima se nalaze interne i toolbox funkcije su automatski dodane u fazi instalacije.

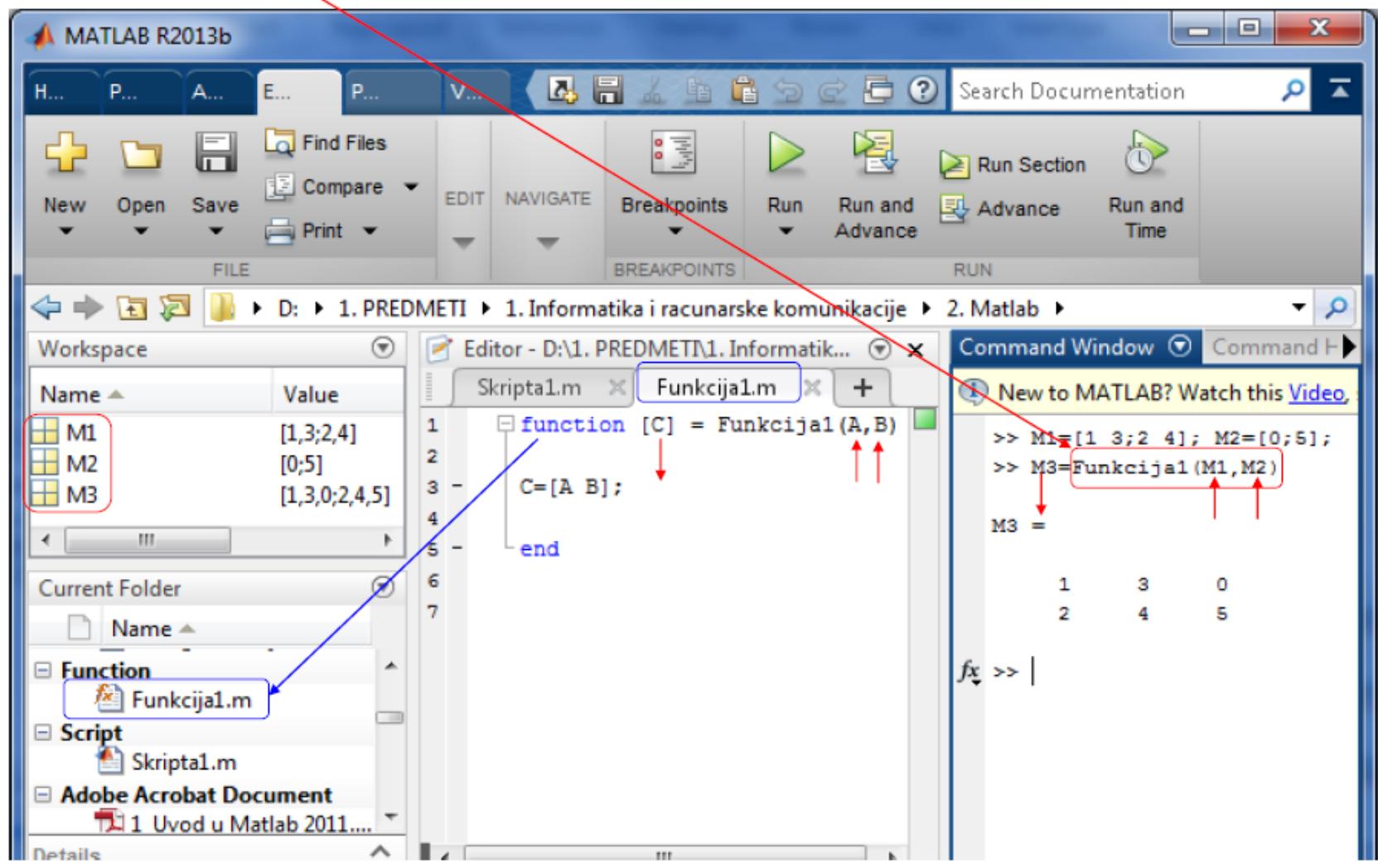
MatLab - Matrix Laboratory.

Direktorijumi sa korisničkim funkcijama dodaju se pomoću naredbe [Set Path](#) sa palete alata.



MatLab - Matrix Laboratory.

Primer. Napisati funkciju „Funkcija1.m“ umesto skripte „Skripta1.m“ i pozvati je sa komandne linije.



MatLab - Matrix Laboratory.

INTERNE FUNKCIJE

Elementarne matematičke funkcije

Trigonometrijske

Logaritamske i hiperbolne funkcije

Ostale funkcije

Funkcije za obradu vektora i matrica

Funkcije za rad s polinomima

MatLab - Matrix Laboratory.

TRIGONOMETRIJSKE FUNKCIJE

Funkcija	Opis
$y=\sin(x)$	Sinus funkcija ugla u radijanima
$y=\cos(x)$	Cosinus funkcija ugla u radijanima
$y=\tan(x)$	Tangens funkcija ugla u radijanima
$y=\text{asin}(x)$	Arcus sinus funkcija $-1 \leq x \leq 1$
$y=\text{acos}(x)$	Arcus cosinus funkcija $-1 \leq x \leq 1$
$y=\text{atan}(x)$	Arcus tangens funkcija
$\phi=\text{atan2}(x,y)$	Arcus tangens funkcija definisana u 4 kvadranta

MatLab - Matrix Laboratory.

LOGARITAMSKE I HIPERBOLIČKE FUNKCIJE

Funkcija	Opis
$y=\exp(x)$	Eksponencijalna funkcija sa osnovom e
$y=\log(x)$	Logaritamska funkcija s osnovom e
$y=\log10(x)$	Logaritamska funkcija s osnovom 10
$y=\sinh(x)$	Funkcija sinus hiperbolični
$y=\cosh(x)$	Funkcija cosinus hiperbolični
$y=\tanh(x)$	Funkcija tangens hiperbolični
$y=\text{asinh}(x)$	Funkcija arcus sinus hiperbolični
$y=\text{acosh}(x)$	Funkcija arcus cosinus hiperbolični
$y=\text{atanh}(x)$	Funkcija arcus tangens hiperbolični

MatLab - Matrix Laboratory.

OSTALE ARITMETIČKE FUNKCIJE

Funkcija	Opis
$y=abs(x)$	Apsolutna vrednost broja x
$y=sqrt(x)$	Kvadratni koren broja x
$y=sign(x)$	Funkcija predznaka (signum funkcija)
$y=angle(x)$	Ugao kompleksnog broja (u radijanima)
$y=real(x)$	Realni deo kompleksnog broja
$y=imag(x)$	Imaginarni deo kompleksnog broja
$y=conj(x)$	Konjugovano-kompleksna vrednost broja x
$y=round(x)$	Zaokruživanje prema najbližem celom broju
$y=ceil(x)$	Zaokruživanje prema najbližem celom broju u smeru $+\infty$
$y=floor(x)$	Zaokruživanje prema najbližem celom broju u smeru $-\infty$

MatLab - Matrix Laboratory.

IZRAZI

Opšti oblik:

promenljiva = izraz

izraz: konstante, imena promenljivih ili funkcija povezana operatorima.

- Znak “;” na kraju izraza sprečava da se ispiše vrednost promenljive.
- Ako se izostavi promenljiva sa leve strane znaka “=”, privremena promenljiva **ans** prima vrednost izraza
- Kod dugačkih izraza, kada je potrebno unos nastaviti u narednom redu, prethodni red završiti sa “...” (tri tačke)

```
a=23
```

```
a=
```

```
23
```

```
B=2*a^2+3*a+4;
```

```
c=(a+B)/B
```

```
c =
```

```
1.0203
```

```
(a+B)/B % izostav. prom.
```

```
ans =
```

```
1.0203
```

```
d=a+B+c*(a-10) ...
```

```
+b/a
```

```
d =
```

```
1.2164e+03
```

MatLab - Matrix Laboratory.

Generator slučajnih brojeva

- U mnogim naučnim i inženjerskim problemima slučajni brojevi igraju važnu ulogu. MATLAB, kao i većina programskih jezika, ima ugradjenu podršku za generisanje slučajnih brojeva. Da budemo precizni, brojevi koje daje generator slučajnih brojeva nisu istinski slučajni jer ih izračunava deterministički algoritam. Stoga se oni nazivaju "pseudo slučajnim" brojevima. Ali za većinu praktičnih problema oni su dovoljni. Primetite da su svi brojevi koje generiše rand iznad između 0 i 1. To nije slučajno. Funkcija vraća brojeve striktno veće od 0 i manje od 1 koji su jednoliko rasporedjeni. To znači da bilo koji broj između 0 i 1 ima potpuno istu verovatnoću da se pojavi kao izlaz rand funkcije. Da to demonstriramo, pokušajmo ovo:

```
>> rand  
ans =  
0.8147
```

```
>> rand(3)  
ans =  
0.9058 0.6324 0.5469  
0.1270 0.0975 0.9575  
0.9134 0.2785 0.9649
```

```
>> rand(3,5)  
ans =  
0.1576 0.4854 0.4218 0.9595 0.8491  
0.9706 0.8003 0.9157 0.6557 0.9340  
0.9572 0.1419 0.7922 0.0357 0.6787
```

Funkcije za rad sa nizovima

Funkcija	Opis
size (A)	Vraca broj elemenata duz svake dimenzije u nizu. Kod vektora i matrica rezultat ce biti vektor ciji prvi element predstavlja broj vrsta a drugi element broj kolona u nizu.
length (a)	Vraca broj elemenata u vektoru.
ndims (A)	Vraca dimenziju niza.
numel (A)	Vraca broj elemenata u nizu.
isempty (A)	Vraca logicko 1 (True) ako je niz A prazan. Niz je prazan ako mu nisu definisani elementi.
isequal (A, B)	Vraca logicko 1 (True) ako su nizovi A i B numericki ekvivalentni. To znaci ako imaju istu dimenziju i jednake vrednosti odgovarajucih elemenata.
reshape (A, m, n)	Od postojeceg niza X dimenzije pxq formira niz dimenzije mxn. Preuredjivanje se vrsti po kolonama. Pri tome mora da vazi $pxq = mxn$.
diag (a)	Ako je a vektor formira kvadratnu matricu sa elementima na glavnoj dijagonali jednakim vektoru a.
diag (A)	Ako je A matrica formira vektor od elemenata njene glavne dijagonale.
find (A)	Vraca indekse elemenata niza A koji su razliciti od nule.
end	Koristi se kao oznaka poslednjeg elementa u nizu. Na primer: a (3 : end) vraca sve elemente niza a osim prva dva.
isscalar (a)	Vraca 1 ukoliko je a skalar, u suprotnom vraca 0.
isvector (a)	Vraca 1 ukoliko je a vektor, u suprotnom vraca 0.

Ugradjene funkcije za rad sa nizovima

Funkcija	Opis
<code>inv (A)</code>	Odredjuje inverznu matricu kvadratne matrice A
<code>det (A)</code>	Odredjuje determinantu kvadratne matrice A
<code>sum (A)</code>	Ako je A vektor vraca zbir njegovih elemenata. Ako je A matrica vraca vektor vrstu ciji elementi predstavljaju zbirove elemenata kolona matrice
<code>mean (A)</code>	Ako je A vektor vraca srednju vrednost njegovih elemenata. Ako je A matrica vraca vektor vrstu sa srednjim vrednostima kolona matrice.
<code>sort (A)</code>	Ako je A vektor sortira elemente vektora u rastucem redosledu. Ako je A matrica sortiranje primenjuje na svaku kolonu matrice pojedinačno.
<code>e=max (A)</code>	Ako je A vektor, e je najveci element vektora. Ako je A matrica e je vektor vrsta sa najvecim elementima kolona matrice A
<code>[e n]=max (A)</code>	Ako je A vektor, e je najveci element vektora a n je redni broj elementa. Ako je A matrica e je vektor vrsta sa najvecim elementima kolona matrice A a n je vektor vrsta sa rednim brojevima najvecih elemenata po kolonama.
<code>e=min (A)</code>	Kao i za <code>e=max (A)</code> ali za najmanji element
<code>[e n]=min (A)</code>	Kao i za <code>[e n]=max (A)</code> ali za najmanji element
<code>dot (a, b)</code>	Izracunava skalarni proizvod vektora a i b
<code>cross (a, b)</code>	Izracunava vektorski proizvod vektora a i b. Vektori moraju da imaju po tri elementa (koordinate).

MatLab - Matrix Laboratory.

NIZOVI I MATRICE I OPERACIJE NAD NJIMA

Vektor vrsta

```
a=[1 3 6 2 4 1 4 90]
```

```
a =
```

```
1     3     6     2     4     1     4    90
```

Vektor kolona

```
b=[3; 7; 1; 9; 12]
```

```
b =
```

```
3
```

```
7
```

```
1
```

```
9
```

```
12
```

format long	format short
<pre>>> X = [1 2; 3.4 pi] X = 1.0000 2.0000 3.4000 3.1416 >> format long >> X X = 1.0000000000000000 2.0000000000000000 3.4000000000000000 3.141592653589793 >></pre>	<pre>>> X = [1 2; 3.4 pi] X = 1.0000 2.0000 3.4000 3.1416 format short >> X X = 1.0000 2.0000 3.4000 3.1416 >></pre>

MatLab - Matrix Laboratory.

Vektor sa konstantnim korakom između susednih elemenata

```
c=1:10 % podrazumevani korak je 1
```

```
c =  
1 2 3 4 5 6 7 8 9
```

```
c=2:-2:-9 % korak je -2
```

```
c =  
2 0 -2 -4 -6 -8
```

Znači, koristi se takozvani “**Colon Operator**”:

Neka su elementi vektora $x = [1 \ 4 \ 7]$ pravilno rasporedjeni i povećavaju se naprimer za korak po 3. MATLAB pruža novi prikladan način za definisanje ovog vektora:

$x = 1:3:7$, što znači: "Formirati vektor x čiji elementi počinju sa 1, uvećajte svaki za 3 i ne idite više od 7.,,"

```
>> x = 1:3:7
```

```
x =
```

```
1 4 7
```

MatLab - Matrix Laboratory.

Vektor sa konstantnim korakom između susednih elemenata

```
e=linspace(1,200,5) % 5 elemenata od 1 do 20
```

```
e =  
1.0000    50.7500   100.5000   150.2500   200.0000
```

Vektor sa logaritamskim korakom između susednih elemenata

```
f=logspace(-1,2,5) % 5 elementa od  $10^{-1}$  do  $10^2$ 
```

```
f =  
0.1000    0.5623    3.1623    17.7828   100.0000
```

MatLab - Matrix Laboratory.

Sabiranje i oduzimanje vektora (iste dimenzije)

```
a=[1 3 6 2 4 1 4 90];  
b=[3 7 12 3 8 9 2 1];  
c=a+b
```

```
c =  
4 10 18 5 12 10 6 91
```

Skalarni proizvod vektora (usaglašene dimenzije)

```
a=[1 3 6 2 4 1 4 90];  
b=[3 7 12 3 8 9 2 1]';  
c=a*b
```

```
c =  
241
```

MatLab - Matrix Laboratory.

Definisanje i matrica

```
A=[1 2 3; 6 5 4; 9 8 7; 2 7 1] % 3x3 matrica
```

A =

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

Dimenziije matrica i vektora

```
[n m]=size(A)
```

n =

```
4
```

m =

```
3
```

```
a=[1 3 6 2 4 1 4 90];  
[n m]=size(a)
```

n =

```
1
```

m =

```
8
```

```
dim=length(a)
```

dim =

```
8
```

Rad sa matricama

Na primer za matrice	Na primer za matrice	Na primer za matrice
$A = [2 \ 5 \ 4; 3 \ 9 \ 8]$ i $B = [1 \ 2 \ 4; 3 \ 2 \ 4]$	$A = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9; 10 \ 11 \ 12]$ $A =$ 1 2 3 4 5 6 7 8 9 10 11 12	$A = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9; 10 \ 11 \ 12]$ $A =$ 1 2 3 4 5 6 7 8 9 10 11 12
rezultat naredbe $>> A./B$ je ans = 2.0000 2.5000 1.0000 1.0000 4.5000 2.0000	rezultat naredbe $>> A.^2$ je ans = 1 4 9 16 25 36 49 64 81 100 121 144	rezultat naredbe $>> A(2,3)$ je ans = 6 Član matrice iz drugog reda i treće kolone !!!

MatLab - Matrix Laboratory.

Izdvajanje kolona (redova) iz matrice

Primer. Izdvojiti sve elemente iz 2. i 4. kolone

$$A = \begin{bmatrix} 1 & 2 & 3 & 2 & 5 & 9 \\ 4 & 5 & 6 & 4 & 1 & 2 \\ 7 & 8 & 19 & 22 & 11 & 1 \\ 2 & 4 & 8 & 6 & 4 & 0 \\ 0 & 8 & 5 & 9 & 4 & 1 \end{bmatrix}$$

```
A=[1 2 3 4 5 9; 4 5 6 4 1 2; 7 8 19 22 11 1; 2 4 8 6 4 0; 0 8 5 9 4 1]  
B = A([2 4],:)
```

A =

1	2	3	4	5	9
4	5	6	4	1	2
7	8	19	22	11	1
2	4	8	6	4	0
0	8	5	9	4	1

B =

4	5	6	4	1	2
2	4	8	6	4	0

MatLab - Matrix Laboratory.

Primer. Izdvojiti elemente iz poslednje 3 kolone matrice A

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 9 \\ 4 & 5 & 6 & 4 & 1 & 2 \\ 7 & 8 & 19 & 22 & 11 & 1 \\ 2 & 4 & 8 & 6 & 4 & 0 \\ 0 & 8 & 5 & 9 & 4 & 1 \end{bmatrix}$$

```
A=[1 2 3 4 5 9; 4 5 6 4 1 2; 7 8 19 22 11 1; 2 4 8 6 4 0; 0 8 5 9 4 1];  
B = A(:,end-3:end)
```

```
B =  
3 4 5 9  
6 4 1 2  
19 22 11 1  
8 6 4 0  
5 9 4 1
```

MatLab - Matrix Laboratory.

Primer. Izdvojiti sve elemente neparnih vrsta koji pripadaju istovremeno drugoj i trećoj koloni matrice A.

$$A = \begin{bmatrix} 1 & 2 & 3 & 2 & 5 & 9 \\ 4 & 5 & 6 & 4 & 1 & 2 \\ 7 & 8 & 19 & 22 & 11 & 1 \\ 2 & 4 & 8 & 6 & 4 & 0 \\ 0 & 8 & 5 & 9 & 4 & 1 \end{bmatrix}$$

B = A(1:2:end,[2 3])

B =

2 3
8 19
8 5

MatLab - Matrix Laboratory.

- Aritmetika i matrična aritmetika
- Sabiranje i oduzimanje
- Možemo sabirati i oduzimati matrice (i vektore, koji su samo posebni slučajevi matrica) koje imaju istu veličinu (tj. isti broj redova i kolona), stvarajući rezultirajuću matricu iste veličine. Tako, na primer, ako je Y iste veličine kao X, tada je $Z = X + Y$ legalna operacija. Značenje dodavanja, koje takodje nazivaju i "sabiranje niza" i "sabiranje matrica", za dvodimenzionalne nizove sledi:

$Z = X + Y$ znači da za svako $m \text{ i } n$,

$$Z(m,n) = X(m,n) + Y(m,n)$$

Što znači:

```
>> X = [1 5 -2; 3 0 7]
```

```
X =
```

```
1 5 -2  
3 0 7
```

```
>> Y = [6 0 6; 2 2 1]
```

```
Y =
```

```
6 0 6  
2 2 1
```

```
>> Z = X + Y
```

```
Z =
```

```
7 5 4  
5 2 8
```

MatLab - Matrix Laboratory.

- Množenje i deljenje
- Ostali binarni operatori uključuju množenje i deljenje. Postoje dve vrste množenja i deljenja u MATLAB-u — množenje i deljenje niza, koje množi ili deli odgovarajući elementi operanada i množenje i deljenje matrice, što izvodi standardnu operaciju množenja i deljenja koja se koristi u linearnoj algebri. Kao što smo mi već spomenuli, MATLAB, kao i gotovo svi računarski jezici, koristi zvezdicu koja označava množenje i kosu crtu za deljenje. Množenje niza je navedeno na ovaj način, $Z = X.*Y$, a deljenje $Z = X./Y$. Obratite pažnju na tačku ili "tačku" ispred *. Kao i kod dodavanja i oduzimanja, množenje i deljenje niza zahteva da dve matrice budu iste veličine. Definicija množenja i deljenja niza za dvodimenzionalne nizove je kako sledi:

$Z = X.*Y$ znači da za svaki m i n,
 $Z(m,n) = X(m,n)*Y(m,n)$

$Z = X./Y$ znači da za svaki m i n
 $Z(m,n) = X(m,n)/Y(m,n)$

MatLab - Matrix Laboratory.

Spajanje matrica i vektora

Primer. Formirati matricu spajanjem vrsta matrice A i matrice B.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 19 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 45 & 3 & 4 \end{bmatrix}$$

A=[1 2 3; 4 5 6; 7 8 19];

B=[1 2 3; 45 3 4];

C = [A;B]

C =

1 2 3

4 5 6

7 8 19

1 2 3

45 3 4

MatLab - Matrix Laboratory.

Primer. Formirati matricu spajanjem 1. i 3. vrste matrice A i poslednje vrste matrice B.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 19 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 45 & 3 & 4 \end{bmatrix}$$

```
A=[1 2 3; 4 5 6; 7 8 19];
```

```
B=[1 2 3; 45 3 4];
```

```
C = [A([1 3],:);B(end,:)]
```

```
C =
```

```
1      2      3
```

```
7      8      19
```

```
45     3      4
```

MatLab - Matrix Laboratory.

FUNKCIJE ZA KREIRANJE SPECIJALNIH MATRICA

Funkcija	Opis
y=zeros(m) y=zeros(m,n)	Matrica čiji su svi elementi jednaki nula
y=ones(m) y=ones(m,n)	Matrica čiji su svi elementi jednaki jedinici
y=eye(m), y=eye(m,n)	Jedinična matrica
y=rand(m) y=rand(m,n)	Matrica slučajnih vrednosti na [-1,1] s uniformnom raspodelom
y=randn(m) y=randn(m,n)	Matrica slučajnih vrednosti na [-1,1] s normalnom raspodelom

MatLab - Matrix Laboratory.

Specijalne matrice

Jedinična matrica

`I = eye(3,3)`

`I =`

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Nulta matrica

`N = zeros(5,3)`

`N =`

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

Matrica sa svim „1“

`J = ones(3,6)`

`J =`

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

MatLab - Matrix Laboratory.

FUNNCIJE ZA OBRADU VEKTORA I MATRICA

Funkcija	Opis
$y=\min(x)$	Minimum vektora x
$y=\max(x)$	Maksimum vektora x
$y=\text{mean}(x)$	Srednja vrednost vektora x
$y=\text{median}(x)$	Medijana vektora x
$y=\text{std}(x)$	Standardna devijacija vektora x
$y=\text{sum}(x)$	Suma elemenata vektora
$y=\text{cumsum}(x)$	Kumulativna suma elemenata vektora $y=[x_1, x_1+x_2, x_1+x_2+x_3, \dots]$
$y=\text{diff}(x)$	Vektor razlike susednih elemenata $y(i)=x(i+1)-x(i)$
$y=\text{prod}(x)$	Proizvod elemenata vektora
$y=\text{cumprod}(x)$	Kumulativni proizvod elemenata vektora
$y=\text{sort}(x)$	Sortira članove vektora po rastućem redu

MatLab - Matrix Laboratory.

Primer. Odrediti inverznu matricu, determinantu i sopstvene vrednosti matrice

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 19 \end{bmatrix}$$

A=[1 2 3; 4 5 6; 7 8 19];

Ai=inv(A)

Adet=det(A)

Asv=eig(A)

Ai =
-1.5667 0.4667 0.1000
1.1333 0.0667 -0.2000
0.1000 -0.2000 0.1000

Adet =
-30.0000

Asv =

23.1279
-0.5382
2.4102

MatLab - Matrix Laboratory.

Na primer za matrice	Na primer za matrice	Na primer za matrice
<pre>A = [1 2 3; 4 5 6] A = 1 2 3 4 5 6 Transponovaćemo je operatorom</pre>	<pre>A =[1 2 3; 4 5 6; 7 8 9; 10 11 12] A = 1 2 3 4 5 6 7 8 9 10 11 12</pre>	<pre>>> A = randi(89,2,3,4)+10 A(:,:,1) = 42 91 45 50 85 76 A(:,:,2) = 36 37 92 37 13 72</pre> <p>Slučajni broj</p>
<pre>rezultat naredbe >>A_transpose = A' je A_transpose = 1 4 2 5 3 6</pre>	<pre>rezultat naredbe >> A_permute = permute(A,[2,1]) je A_permute = 1 4 7 10 2 5 8 11 3 6 9 12</pre>	<pre>A(:,:,3) = 31 81 30 55 44 75 A(:,:,4) = 79 66 56 72 78 40</pre>

Matrix-building functions

FUNCTION	RETURNS AN N-BY-M MATRIX OF
<code>zeros(N,M)</code>	<code>zeros</code>
<code>ones(N,M)</code>	<code>ones</code>
<code>eye(N,M)</code>	<code>zeros except for the diagonal elements that are ones</code>
<code>rand(N,M)</code>	<code>random numbers uniformly distributed in the range from 0 to 1</code>

Na primer, `zeros(n,m)` vraćaju matricu n-by-m, čiji su svi elementi jednaki nuli. Slično, `ones(n,m)`, vraća n-po-m jedinica. Zanimljiva je funkcija `eye(n,m)`, koja vraća n-by-m matricu koja ima sve nule, osim onih elemenata na dijagonali, koji su jednaki jedan. Dijagonalna matrice je skup elemenata čiji su indeksi medjusobno jednaki. Dakle, u matrici M dijagonalni elementi su M(1,1), M(2,2), Zašto se zove "eye"? Pa, to je zato što `eye(n,n)` proizvodi matricu identiteta, čiji je simbol u matematici I (izgovara se "eye", razumeš?). n-by-n identitet matrica se može pomnožiti sa bilo kojom drugom n-by-n matricom I, a rezultat je I:

<code>I = eye(3,3)</code> <code>I =</code> 1 0 0 0 1 0 0 0 1	<code>I = ones(3,3)</code> <code>I =</code> 1 1 1 1 1 1 1 1 1	<code>I = zeros(3,3)</code> <code>I =</code> 0 0 0 0 0 0 0 0 0	<code>I = rand(3,3)</code> <code>I =</code> 0.6613 0.2797 0.6506 0.0249 0.8107 0.0331 0.9742 0.8740 0.9174

MatLab - Matrix Laboratory.

FUNKCIJE ZA OBRADU STRINGOVA

Funkcija	Opis
s=blanks(n)	Vraća string koji se sastoji od n praznih mesta
s=findstr(s1,s2)	Pronalazi string s2 unutar stringa s1 i vraća indekse gde započinju stringovi s2
s=isstr(s1)	Vraća logičku jedinicu ukoliko je s1 string
s=lower(s1)	Sva slova u stringu s1 pretvara u mala slova
s=upper(s1)	Sva slova u stringu s1 pretvara u velika slova
s=strcat(s1,s2,s3)	Spaja stringove s1,s2 i s3 u jedna string
s=num2str(x)	Pretvara numeričku vrijednost u string

MatLab - Matrix Laboratory.

FUNCTION	RETURNS
<code>fix(x)</code>	Round x towards zero
<code>floor(x)</code>	Round x towards minus infinity
<code>ceil(x)</code>	Round x towards plus infinity
<code>round(x)</code>	Round x towards nearest integer
<code>rem(x,n)</code>	Remainder of x/n (see help for case of noninteger n)
<code>sign(x)</code>	1 if $x > 0$; 0 if x equals 0; -1 if $x < 0$

Funkcije zaokruživanja i ostatka

FUNCTION	RETURNS
<code>length(v)</code>	Number of elements of v
<code>max(v)</code>	Largest element of v
<code>min(v)</code>	Smallest element of v
<code>mean(v)</code>	Mean of v
<code>median(v)</code>	Median element of v
<code>sort(v)</code>	Sorted version of v in ascending order
<code>std(v)</code>	Standard deviation of v
<code>sum(v)</code>	Sum of the elements of v

Opisne funkcije dodeljene vektorima

FUNCTION	RETURNS A ROW VECTOR CONSISTING OF
<code>max(M)</code>	Largest element of each column
<code>min(M)</code>	Smallest element of each column
<code>mean(M)</code>	Mean of each column
<code>median(M)</code>	Median of each column
<code>size(M)</code>	Number of rows, number of columns
<code>sort(M)</code>	Sorted version, in ascending order, of each column
<code>std(M)</code>	Standard deviation of each column
<code>sum(M)</code>	Sum of the elements of each column

Opisne funkcije dodeljene dvodimanzionalnim matricama

MatLab - Matrix Laboratory.

FUNKCIJE ZA RAD SA POLINOMIMA

- Polinomi se u Matlabu predstavljaju pomoću vektora koji sadrže koeficijente uz opadajuće stepene nezavisne promenljive
- Primer

$$P(x) = x^4 - x^2 + 2x \quad \rightarrow \quad P = [1 \ 0 \ -1 \ 2 \ 0]$$

Funkcija	Opis
<code>x=roots(P)</code>	Koreni polinoma P
<code>P=poly(x)</code>	Vraća polinom na osnovu korena polinoma
<code>R=conv(P,Q)</code>	Množenje polinoma P i Q
<code>[R,S]=deconv(P,Q)</code>	Deljenje polinoma P/Q (R je rezultat deljenja a S ostatak)
<code>Q=polyder(P)</code>	Prvi izvod polinoma P po x
<code>P=polyfit(x,y,n)</code>	Metodom najmanjih kvadrata računaju se koeficijenti polinoma P(x) n-tog reda

MatLab - Matrix Laboratory.

Definisanje polinoma

Polinomi se zadaju preko vektora koeficijenata polinoma, počevši od najvećeg stepena polinoma. Na primer:

$$P(s) = 2s^3 + s^2 + 3$$

$$a = [2 \ -1 \ 0 \ 3]$$

Nule polinoma roots()

$$P(s) = 1s^4 + 10s^3 + 35s^2 + 50s + 24 = 0$$

$$s_{1,2,3} = ?$$

```
p=conv([1 3 2],[1 7 12])
```

```
p =  
    1      10      35      50      24
```

```
si=roots(p)
```

```
si =  
-4.0000  
-3.0000  
-2.0000  
-1.0000
```

Kompleksne nule

$$P(s) = 2s^3 + s^2 + 3 = 0$$

$$s_{1,2,3} = ?$$

```
si=roots([2 -1 0 3])
```

```
si =  
0.7500 + 0.9682i  
0.7500 - 0.9682i  
-1.0000 + 0.0000i
```

MatLab - Matrix Laboratory.

Formiranje polinoma od njegovih nula poly()

$$s_1 = -1, s_2 = 2, s_3 = -5$$

$$P(s) = ?$$

$$\begin{aligned} P(s) &= (s - (-1))(s - 2)(s - (-5)) \\ &= 1s^3 + 4s^2 - 7s - 10 \end{aligned}$$

```
P=poly([-1 2 -5])
P =
    1   4   -7  -10
```

Izračunavanje polinoma polyval()

$$P(s) = 1s^3 + 2s^2 - 5$$

$$P(2) = ?$$

$$P(2) = -5$$

```
y = polyval([-1 2 -5],2)
y =
    -5
```

MatLab - Matrix Laboratory.

Množenje polinoma conv()

$$P(s) = 2s^3 + s^2 + 3$$

$$Q(s) = 2s^2 + s - 1$$

$$R(s) = P(s) \cdot Q(s)$$

```
R=conv([2 -1 0 3],[2 1 -1])
```

R =

```
4 0 -3 7 3 -3
```

Delejne polinoma deconv()

$$P(s) = 2s^3 + s^2 + 3$$

$$Q(s) = 2s^2 + s - 1$$

$$\frac{P(s)}{Q(s)} = R(s) + \frac{O(s)}{Q(s)}$$

$$= s - 1 + \frac{2s + 2}{Q(s)}$$

```
[R,O]=deconv([2 -1 0 3],[2 1 -1])
```

R =

```
1 -1
```

O =

```
0 0 2 2
```

MatLab - Matrix Laboratory.

Izvod polinoma polyder()

$$P(s) = 2s^3 + s^2 + 3$$

$$\frac{dP(s)}{ds} = 6s^2 + 2s$$

```
Pp=polyder([2 -1 0 3])
```

```
Pp =  
6 -2 0
```

Integral polinoma polyint()

$$P(s) = 2s^3 - s^2 + 3$$

$$\int P(s)ds = 2\frac{s^4}{4} - \frac{s^3}{3} + 3s$$

$$= \frac{1}{2}s^4 - \frac{1}{3}s^3 + 3s$$

```
Pi=polyint([2 -1 0 3])
```

```
Pi =  
0.5000 -0.3333 0 3.0000 0
```

MatLab - Matrix Laboratory.

Predstavljanje količnika dva polinoma parcijalnim sabircima residue()

a. Slučaj realnih korena u imeniku količnika $b(s)/a(s)$

$$\frac{b_m(s)}{a_n(s)} = \frac{R_1}{s - p_1} + \dots + \frac{R_n}{s - p_n} + K_{m-n}(s)$$

$$b(s) = 2s^3 + s^2 + 3$$

$$a(s) = s^2 + 3s + 2 = (s+1)(s+2)$$

$$\frac{b(s)}{a(s)} = \frac{9}{s+2} + \frac{2}{s+1} + (2s-5)$$

```
b=[2 1 0 3];
a=[1 3 2]
[r, p, k] = residue(b,a)
```

r =

9

2

p =

-2

-1

k =

2 -5

Rešavanje linearog sistema jednačina

- Sistem $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ ima rešenje $\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b}$
 - u Matlab-u: $\mathbf{x} = \mathbf{A} \setminus \mathbf{b}$
- Sistem $\mathbf{x} \cdot \mathbf{A} = \mathbf{b}$ ima rešenje $\mathbf{x} = \mathbf{b} / \mathbf{A}$
 - u Matlab-u: $\mathbf{x} = \mathbf{b} / \mathbf{A}$
- Primer:

$$x_1 + 2x_2 - x_3 = -2.5$$

$$-x_1 + 3x_2 + 2x_3 = 0.5$$

$$-x_1 - x_2 + x_3 = 1.5$$

$$\begin{bmatrix} 1 & 2 & -1 \\ -1 & 3 & 2 \\ -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2.5 \\ 0.5 \\ 1.5 \end{bmatrix}$$

```
» A = [1 2 -1; -1 3 2; -1 -1 1];
```

```
» b = [-2.5; 0.5; 1.5];
```

```
» x = A \ b
```

x =

2.5000

-1.0000

3.0000

```
» A * x
```

ans =

-2.5000

0.5000

1.5000

```
» x = inv(A) * b
```

x =

2.5000

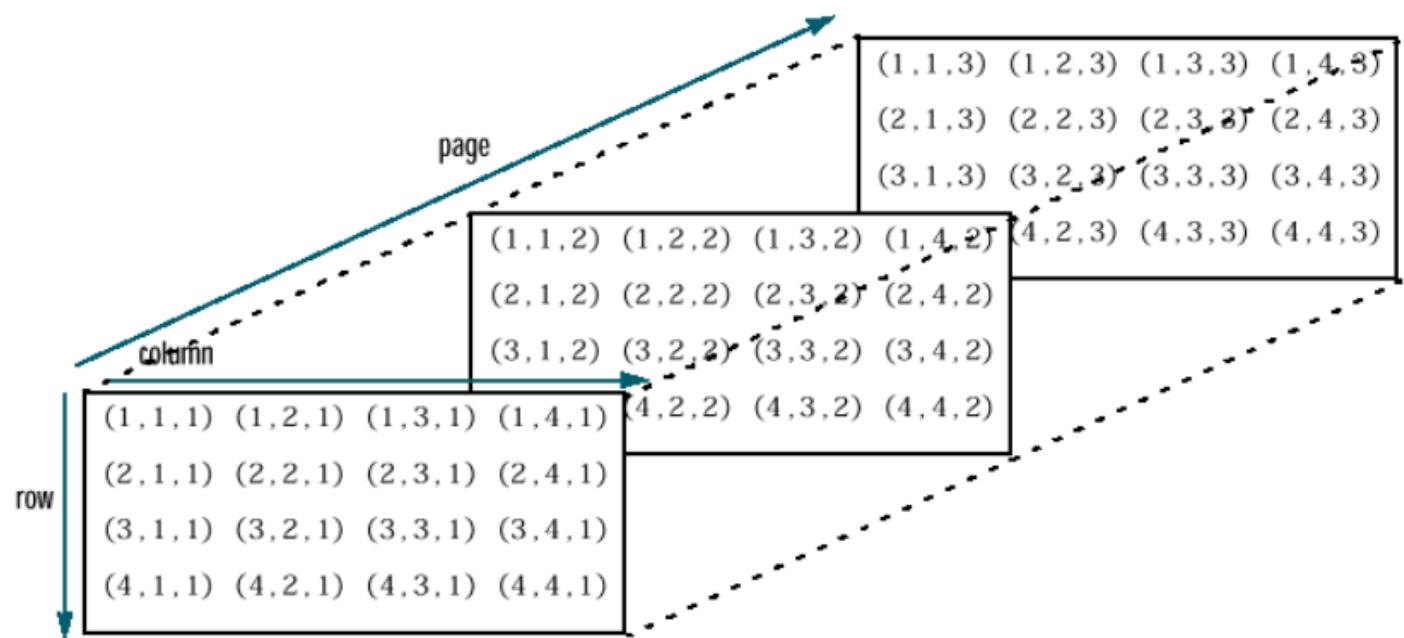
-1.0000

3.0000

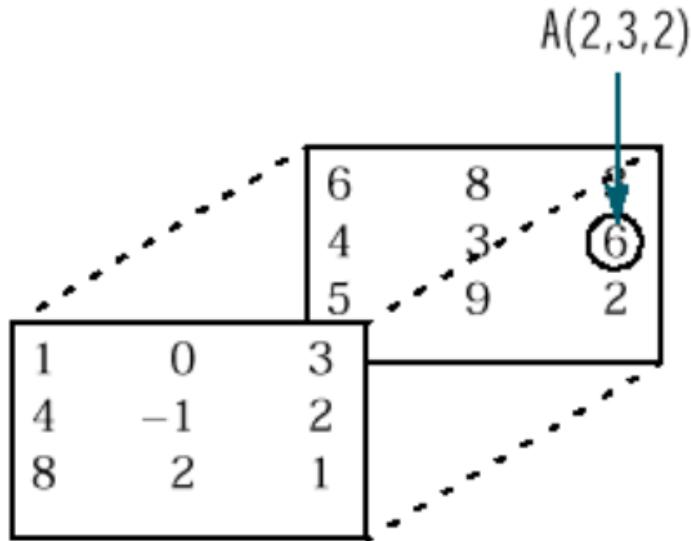
Višedimenzioni nizovi

- dvodimenzioni niz == matrica
 - prvi indeks je redni broj vrste
 - drugi indeks je redni broj kolone
- trodimenzioni niz
 - treći indeks je redni broj strane

column				
row	(1 , 1)	(1 , 2)	(1 , 3)	(1 , 4)
(2 , 1)	(2 , 2)	(2 , 3)	(2 , 4)	
(3 , 1)	(3 , 2)	(3 , 3)	(3 , 4)	
(4 , 1)	(4 , 2)	(4 , 3)	(4 , 4)	



Pristup elementima nizova



» $A(2,3,2)$

ans =

6

» $A(1,2,2)$

ans =

8

» $A(:,2,1)$

ans =

0

-1

2

» $A(:, :, 2)$

ans =

6 8 3

4 3 6

5 9 2

» $A(:,2,:)$

ans(:, :, 1) =

0

-1

2

ans(:, :, 2) =

8

3

9

» $A([1 3], 3, [2 1])$

ans(:, :, 1) =

3

2

ans(:, :, 2) =

3

1

Formiranje višedimenzionih nizova

- na način formiranja matrica, preko indeksa
 - broj elemenata u podnizovima se mora slagati
- primer dodavanja nove dimenzije za A
A je sada 3x3x3x2

```
» A(:,:,1,2) = [1 2 3; 4 5 6; 7 8 9];  
» A(:,:,2,2) = [9 8 7; 6 5 4; 3 2 1];  
» A(:,:,3,2) = [1 0 1; 1 1 0; 0 1 1];
```

```
» A=[1 0 3; 4 -1 2; 8 2 1];  
» A(:,:,2)=[6 8 3; 4 3 6; 5 9 2];  
» A(:,:,3) = 5
```

A(:,:,1) =

1	0	3
4	-1	2
8	2	1

A(:,:,2) =

6	8	3
4	3	6
5	9	2

A(:,:,3) =

5	5	5
5	5	5
5	5	5

Formiranje višedimenzionih nizova

- upotrebljom funkcija: **ones**, **zeros**, **randn**, ...
- umnožavanjem **repmat** osnovne matrice
- upotrebljom **cat** funkcije
B = cat(dim,A1,A2,...)
 - spaja listu nizova duž dimenzijske dimenzije
 - jedinične dimenzijske se automatski umeću

```
» B = cat(4,[2 8; 0 5],[1 3; 7 9])
```

B(:,:,1,1) =

2 8

0 5

B(:,:,1,2) =

1 3

7 9

```
» B = randn(1,1,2)
```

B(:,:,1) =

-0.6918

B(:,:,2) =

0.8580

```
» B = repmat([1 0],[3 4 2])
```

B(:,:,1) =

1 0 1 0 1 0 1 0

1 0 1 0 1 0 1 0

1 0 1 0 1 0 1 0

B(:,:,2) =

1 0 1 0 1 0 1 0

1 0 1 0 1 0 1 0

1 0 1 0 1 0 1 0

MatLab - Matrix Laboratory.

KONTROLA TOKA PROGRAMA

MATLAB pruža četiri oblika kontrole toka programa:

- for petlja
- while petlja
- if-else-end struktura
- switch-case struktura

MatLab - Matrix Laboratory.

for petlja

for index = values statements end

for index = values, statements, end izvršava grupu izraza u petlji odredjeni broj puta. vrednost ima jedan od sledećih oblika:

početna vrednost:krajnja vrednost —

Povećavamo promenljivu indeksa sa početne vrednosti do krajne vrednosti za 1 i ponavljamo izvršavanje naredbi dok indeks ne bude veći od krajne vrednosti.

početna vrednost:korak:krajnja vrednost —

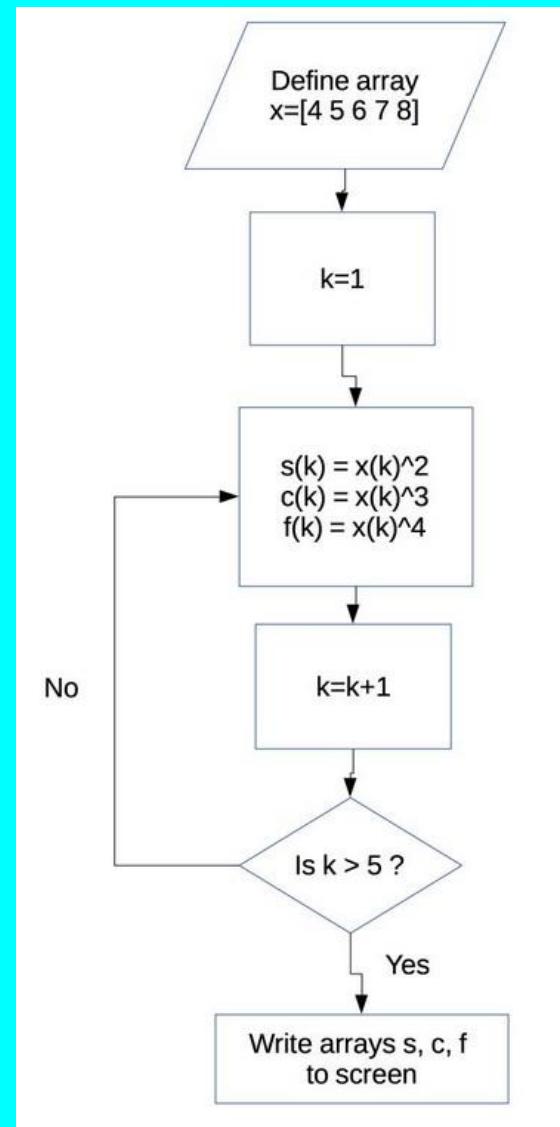
Povećavamo indeks za korak vrednosti na svakoj iteraciji ili smanjite indeks kada je korak negativan.

```
clear all  
x = 4:8; //polje  
for k=1:5 //indeks  
    s(k) = x(k)^2;  
    c(k) = x(k)^3;  
    f(k) = x(k)^4;  
end
```

naredbe
kraj petlje

Rezultat:

```
s=16 25 36 49 64  
c=64 125 216 343 512  
f=256 625 1296 2401 4096
```



MatLab - Matrix Laboratory.

FOR PETLJA

Omogućavaju da se grupa naredbi ponavlja unaprijed određeni broj puta.

Opšti oblik

```
for x = niz  
    naredbe...  
end
```

Naredbe između for i end izvršavaju se jednom za svaku vrednost iz niza.

Primer. (jednostruki FOR)

Izračunati $\sum_{i=1}^{10} \sin(i \frac{\pi}{10})$.

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

Primer. (ugnježdeni FOR) Odrediti matricu

$A_{5 \times 4}$ sa elementima $a_{ij} = \frac{1}{i + j + 1}$

```
for i = 1:5  
    for j = 1:4  
        a(i,j) = 1 / (i+j-1);  
    end  
end
```

MatLab - Matrix Laboratory.

```

s = 10;
H = zeros(s);
for c = 1:s
    for r = 1:s
        H(r,c) = 1/(r+c-1);
    end
end

```

NAPOMENA: Iteracija počinje se sa vodećom petljom pa se ugnježđena izvršava do kraja. Tada se indeks vodeće petlje uvećava za 1 i opet ugnježđena izvršava do kraja. Tako, sve dok uslov petlje važi.

MatLab - Matrix Laboratory.

WHILE PETLJA

Naredbe između while i end izvršavaju se sve dok su svi elementi izraza istiniti:

```
while izraz  
    naredbe...  
end
```

Primer. Izračunati sumu prvih 100 prirodnih brojeva

```
S=0; n=1;  
while n<=100  
    S=S+n;  
    n=n+1;  
end  
S  
>> S=5050
```

Primer. Izračunati $\sum_{i=1}^{\infty} \frac{1}{i^2}$ sa tačnošću 10^{-4} .

```
s = 0; n = 1;  
while 1 / n^2 > 1e-4  
    s = s + (-1)^n / n^2;  
    n = n+1;  
end  
s  
>> s=-0.8225
```

MatLab - Matrix Laboratory.

IF-ELSE-END STRUKTURA

Opšti oblik:

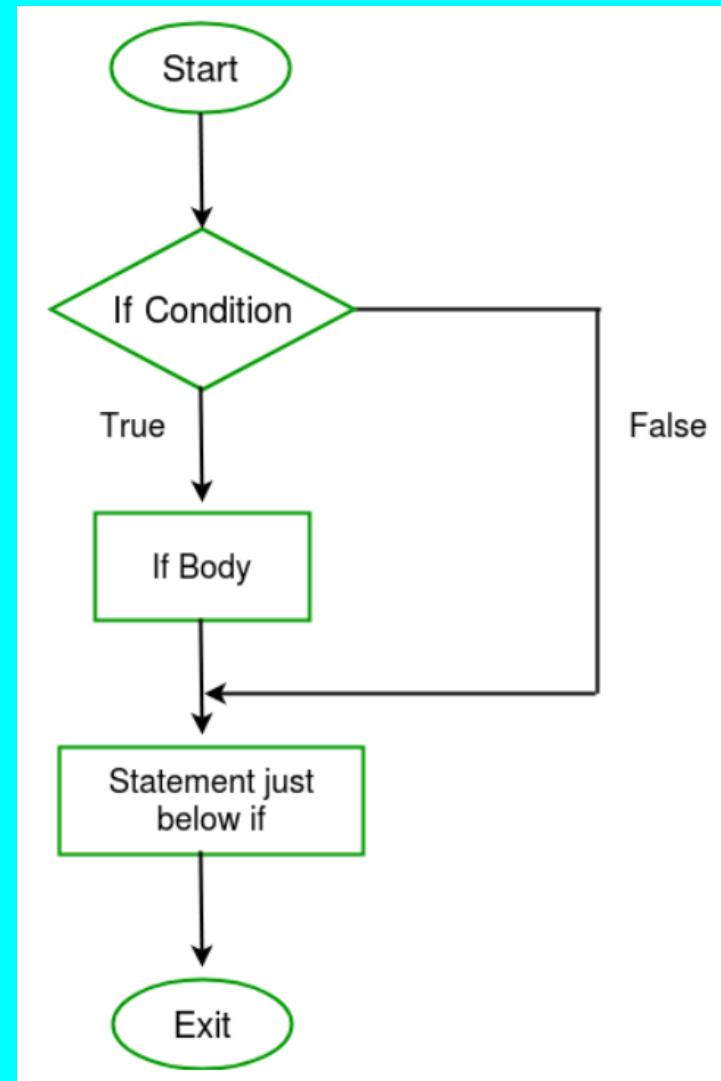
```
if uslov 1  
    blok naredbi 1  
elseif uslov 2  
    blok naredbi 2  
elseif uslov 3  
    ...  
else  
    blok naredbi n  
end
```

Tipični oblik 1:

```
if uslov  
    blok naredbi  
end
```

Tipični oblik 2:

```
if uslov  
    blok naredbi 1  
else  
    blok naredbi 2  
end
```



MatLab - Matrix Laboratory.

Primer. Odrediti novu vrednost parametra a u zavisnosti od njegove tekuće vrednosti i vrednosti parametra b na sledeći način:

$$a = \begin{cases} 0 & a < b \text{ ili } a < 3 \\ a, & \text{inače} \end{cases}$$

```
if a<b | a<3  
    a=0;  
end;
```

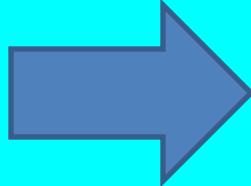
Primer. Odrediti novu vrednost parametra a u zavisnosti od njegove tekuće vrednosti i vrednosti parametara b i c na sledeći način:

$$a = \begin{cases} 0 & a > b, a < 3 \\ 1, & a > c \\ a & \text{inače} \end{cases}$$

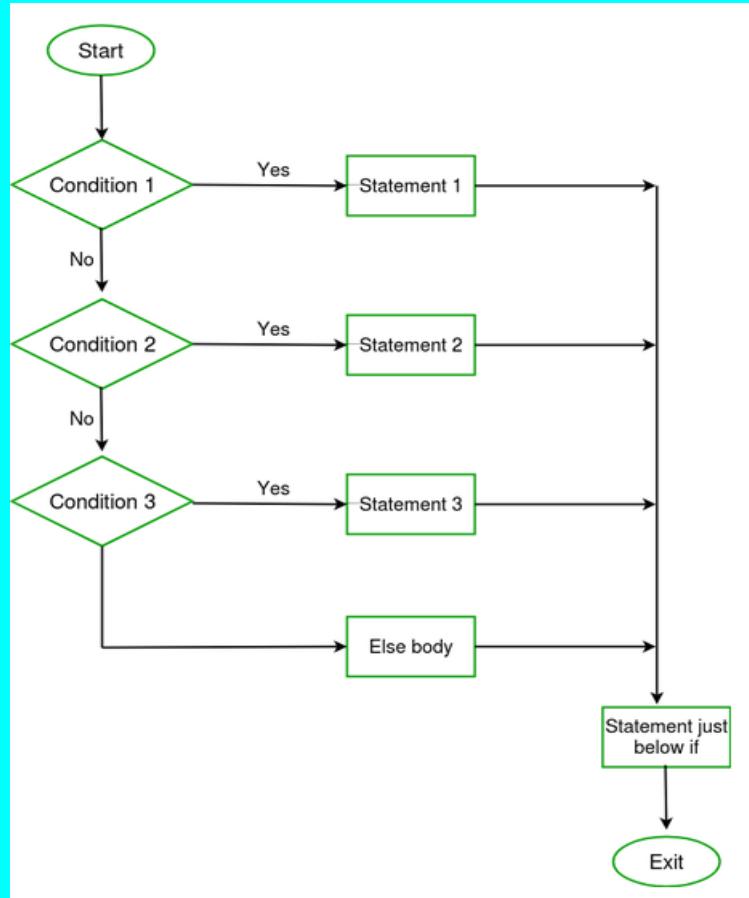
```
if a>b & a<3  
    a=0;  
elseif a>c  
    a=1;  
end;
```

MatLab - Matrix Laboratory.

```
function day_of_week(n)
• if n == 1
•   fprintf('Sunday,');
•   day_type = 2;
• elseif n == 2
•   fprintf('Monday,');
•   day_type = 1;
• elseif n == 3
•   fprintf('Tuesday,');
•   day_type = 1;
• elseif n == 4
•   fprintf('Wednesday,');
•   day_type = 1;
• elseif n == 5
•   fprintf('Thursday,');
•   day_type = 1;
• elseif n == 6
•   fprintf('Friday,');
•   day_type = 1;
• elseif n == 7
•   fprintf('Saturday,');
•   day_type = 2;
• else
•   fprintf('Number must be from 1 to 7.\n');
• return
• end
• if day_type == 1
•   fprintf(' which is a week day\n');
• else
•   fprintf(' which is a weekend day\n');
• end
```



```
>> day_of_week(1)
Sunday, which is a weekend day
>> day_of_week(4)
Wednesday, which is a week day
>> day_of_week(-2)
Number must be from 1 to 7.
```

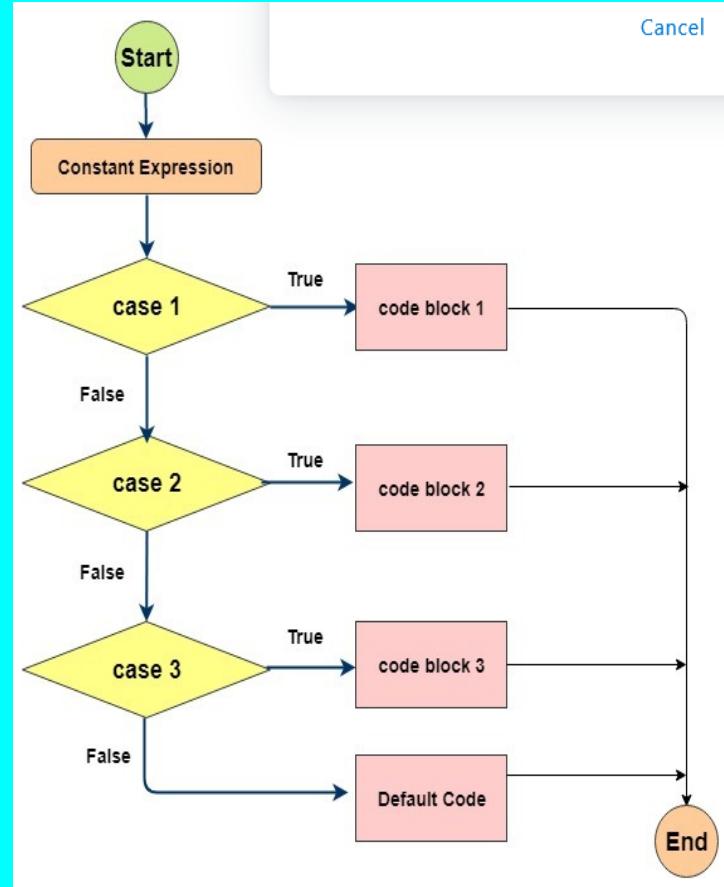


MatLab - Matrix Laboratory.

switch-struktura

- Uz mnoge oblike if-naredbe, MATLAB nudi jedan dodatni izborni konstrukt — switch-naredbu. Uvesćemo switch-naredbu koristeći je za izvodjenje zadatka koji smo gore izveli sa if-izjavama. Vratimo se funkciji iz prethodnog odjeljka: day_of_week, koja imenuje dan u nebelji koji odgovara broju koji mu unesemo i govori nam je li to radni dan ili vikend. Evo verzije switch-naredbe:

```
function day_of_week_switch(n)
switch n
case 1
    fprintf('Sunday,');
    day_type = 2;
case 2
    fprintf('Monday,');
    day_type = 1;
case 3
    fprintf('Tuesday,');
    day_type = 1;
case 4
    fprintf('Wednesday,');
    day_type = 1;
case 5
    fprintf('Thursday,');
    day_type = 1;
case 6
    fprintf('Friday,');
    day_type = 1;
case 7
    fprintf('Saturday,');
    day_type = 2;
otherwise
    fprintf('Number must be from 1 to 7.\n');
return
end
if day_type == 1
    fprintf(' which is a week day\n');
else
    fprintf(' which is a weekend day\n');
end
```



```

1. for i = X % za svaki stupac matrice X
2. disp(i); % izvrši tijelo petlje
3. end

```

Naravno, ukoliko se radi o vektor-retku brojač poprima sve vrijednosti unutar tog retka:

```

1. for i = [1 2 5 7] % za svaki element vektora
2. disp(i); % izvrši tijelo petlje
3. end

```

Obično se koristi operator : koji omogućuje jednostavno postavljanje početne i konačne vrijednosti te koraka:

```

1. a = zeros(1,10); % inicijaliziraj vektor a
2. for i = 1:0.5:10 % za svaki i od 1 do 10 s korakom 0.5
3. a(i) = i*i -3; % na i-to mjesto vektora a ubaci vrijednost i*i-3
4. end % kraj for petlje

```

while petlja se izvršava dok je ispunjen logički uvjet petlje:

```

1. i = 7; % inicijaliziraj varijablu i
2. while (i >= 0) % dok je i veći ili jednak 0 ponavljaj
3. if (a(i) ~= 5) % ako a(i) nije jednako 5
4. a(i) = a(i) - 3; % tada a(i) smanji za tri
5. else % inače
6. a(i) = 127; % u a(i) spremi 127
7. end % kraj if naredbe
8. i = i - 1; % smanji i za 1
9. end % kraj while petlje

```

Naredbu break koristimo za bezuvjetni izlazak samo iz for ili while petlje.

```

1. i = 7;
2. while (i >= 0)
3. if (i > 10)
4. break % ako je i veći od 10
5. end % bezuvjetno prekini petlju
6. end % kraj if naredbe

```

if naredba omogućuje uvjetna izvršavanja koda:

```

1. if a == b
2. c = a + b;
3. elseif abs(a) == b
4. c = a - b;
5. else
6. c = 0;
7. end

```

Za komplikiranija uvjetna izvršavanja obično koristimo naredbu switch:

```

1. a = 10;
2. switch a
3. case a < 9
4. disp('Broj je manji od devet.');
5. case a > 10
6. disp('Broj je veći od deset');
7. otherwise
8. disp('Čestitamo! Osvojili ste Bingo!');
9. end

```

PRIMERI:

MatLab - Prozori figura

- Do sada je izlaz koji je MATLAB proizveo bio jednostavan tekst unutar komandnog prozora. Grafički izlaz je zabavniji i u trenutku može dati mnogo više podataka. Da bi dobili grafički izlaz, jednostavno koristite naredbu koja ga proizvodi, kao što je **plot**. Isprobamo sada nacrtati graf, vratite se na komandni prozor i stvorite dva vektora sa naredbama:

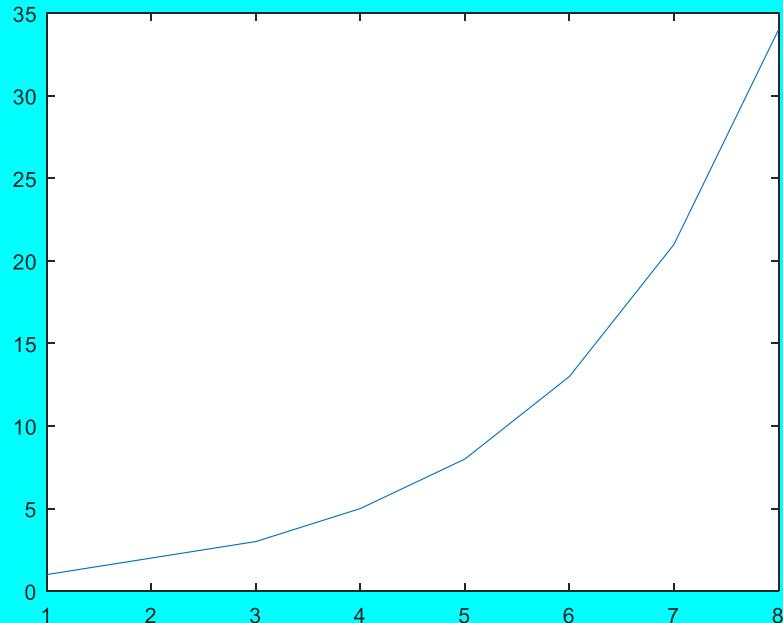
```
>> x = [1 2 3 4 5 6 7 8];
```

```
>> y = [1 2 3 5 8 13 21 34];
```

Dve gornje naredbe postavljaju x i y kao dva vektora sa 8 elemenata. Sada dajemo sledeću naredbu:

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

- Grafik se pojavljuje u prozoru "Figure", koji se automatski pojavljuje kada izvršimo naredbu plot, kao što je prikazano na slici.
- Postoji mnogo mogućih varijacija ovog pojednostavnog crteža. Tačkaste ili isprekidane linije se mogu koristiti, pojedinačne tačke se mogu crtati bez povezivanja, mogu se iscrtati ravne linije, simboli osim tačaka, razne boje korišćene, oznake labela se mogu staviti na ose itd.



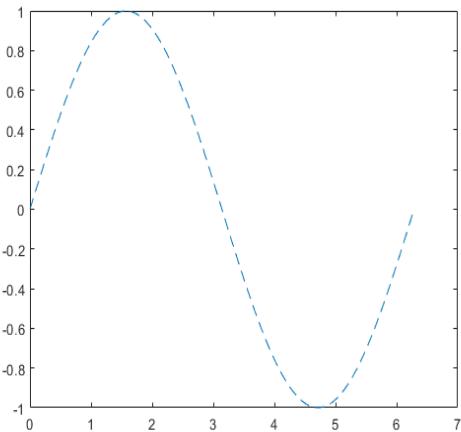
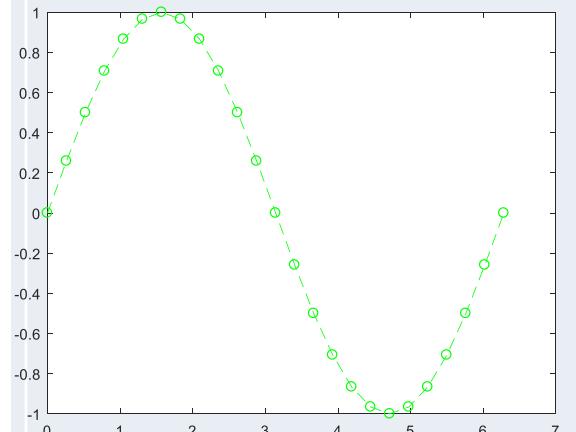
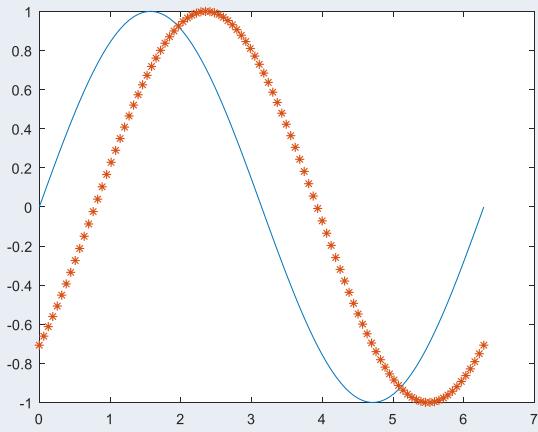
MatLab - Prozori figura

Oznaka boje	Opis	Ekvalent RGB Tripletu
'red' or 'r'	Crvena	[1 0 0]
'green' or 'g'	Zelena	[0 1 0]
'blue' or 'b'	Plava	[0 0 1]
'yellow' or 'y'	Žuta	[1 1 0]
'magenta' or 'm'	Magenta	[1 0 1]
'cyan' or 'c'	Cian	[0 1 1]
'white' or 'w'	Bela	[1 1 1]
'black' or 'k'	Crna	[0 0 0]

Oznaka markera	Opis
'o'	Krug
'+'	Znak plus
'*'	Zvezdica
'.'	Tačka
'-'	Crta
'x'	Puta
'square' or 's'	Kvadrat
'diamond' or 'd'	Dijamant
'^'	Trougao naviše
'v'	Trougao naniže
Trougao udesno	
'<'	Trougao ulevo
'pentagram' or 'p'	pentagram
'hexagram' or 'h'	heksagram
'none'	Bez markera

Boje i markere možemo po izboru dodeljivati u srteže na našoj figuri

MatLab - Prozori figura (2D)

Funkcija	Funkcija	Funkcija
<pre>x = linspace(0,2*pi,100); y = sin(x); figure plot(x,y,'--')</pre> 	<pre>x = linspace(0,2*pi,25); y = sin(x); Figure plot(x,y,'--go')</pre> 	<pre>x = linspace(0,2*pi,100); y1 = sin(x); y2 = sin(x-pi/4); figure plot(x,y1,'-',x,y2,'*')</pre> 

MatLab - Prozori figura (2D)

Funkcija	Funkcija	Funkcija
<pre>x = linspace(0,2*pi,60); a = sin(x); b = cos(x); plot(x,a+b) hold on stem(x,a) stem(x,b) hold off</pre>	<pre>x = linspace(0,2*pi,60); a = sin(x); b = cos(x); plot(x,a+b) hold on stem(x,a) stem(x,b) hold off title('Linearna kombinacija dve funkcije') xlabel('Vreme \musecs') ylabel('Intenzitet') legend('a+b','a = sin(x)','b = cos(x)')</pre>	<pre>x = linspace(0,2*pi,100); y1 = sin(x); y2 = sin(x-pi/4); figure plot(x,y1,'-') hold on plot(x,y2,'*') legend('y1', 'y2') xlabel('vrednosti za x') ylabel('Grafik dve funkcije')</pre>

MatLab - Prozori figura (plot::Bars2d)

plot::Bars2d([[a₁, a₂, ...], [b₁, b₂, ...], ...]) generiše bar čart sa visinama a₁, b₁, ..., a₂, b₂, ... u jednoj boji

plot::Bars2d([a₁, a₂, ...]) generiše bar čart sa visinama a₁, b₁, ...

Sa plot::Bars2d([[a₁, a₂, ...], [b₁, b₂, ...], ...]), bar se crta u poretku a₁, b₁, ..., sa razmakom, a₂, b₂, ...

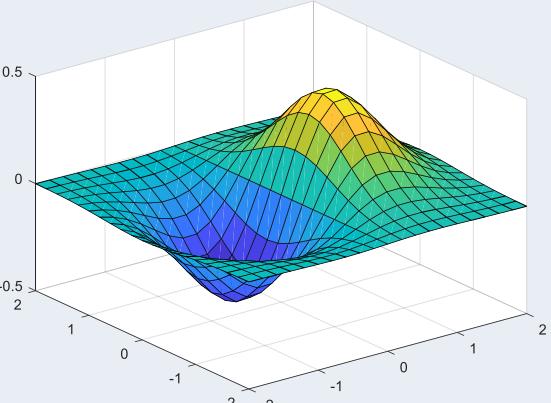
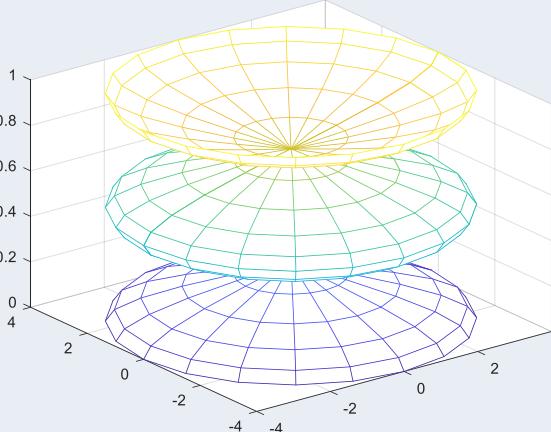
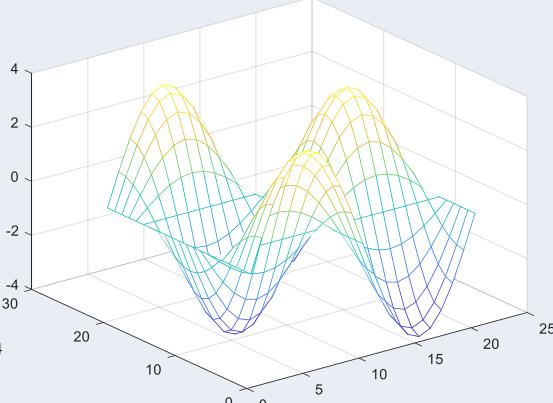
Funkcija	Funkcija	Funkcija
<pre>plot(plot::Bars2d([1, 2, 3, 4, 5]))</pre>	<pre>plot(plot::Bars2d([[5, 10, 24, -3], [6, 5, 2, 18],[19, 45, 12,-10]]))</pre>	<pre>plot(plot::Bars2d([[.5, 1.0, 2.4, -.3], [.6, .5, .2, 1.8],[1.9, 4.5, 1.2, -1.0]], Colors = [RGB::Red, RGB::Green, RGB::Blue], Shadows = TRUE, DrawMode = Horizontal))</pre>

The figure consists of three separate bar charts side-by-side, each generated by a different call to the plot::Bars2d function.

- Left Chart:** A vertical bar chart with a single y-axis labeled 'y' ranging from 0 to 5. It contains four blue bars with heights corresponding to the values in the input vector [1, 2, 3, 5].
- Middle Chart:** A vertical bar chart with a y-axis labeled 'y' ranging from -10 to 40. It contains eight bars in two colors: blue and red. The heights of the bars correspond to the values in the input matrix [[5, 10, 24, -3], [6, 5, 2, 18], [19, 45, 12, -10]]. Note that the negative value -3 is represented by a red bar at y=-3.
- Right Chart:** A horizontal bar chart with a y-axis labeled 'y' ranging from 0 to 14. It contains ten horizontal bars in three colors: blue, green, and red. The lengths of the bars correspond to the values in the input matrix [[.5, 1.0, 2.4, -.3], [.6, .5, .2, 1.8], [1.9, 4.5, 1.2, -1.0]]. The negative value -.3 is represented by a red bar extending below the zero line.

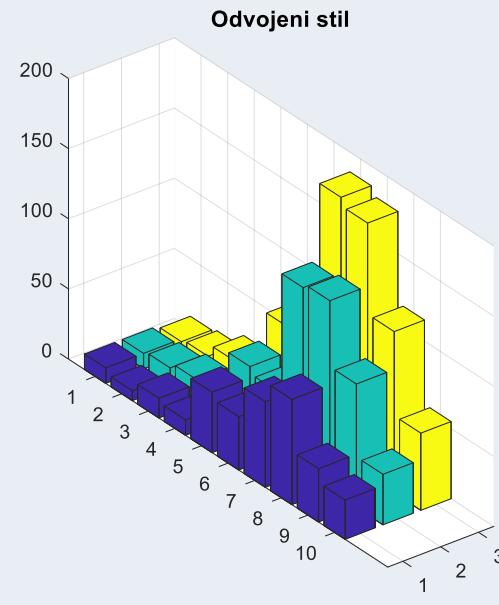
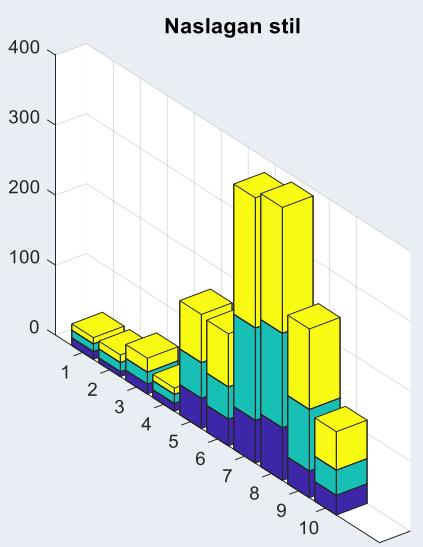
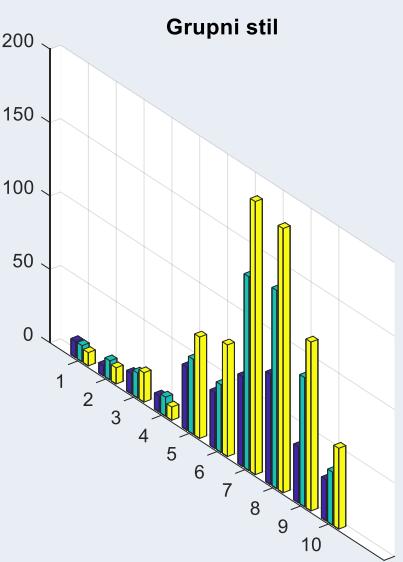
MatLab - Prozori figura (3D)

- Trodimenzionalni grafikoni obično prikazuju površinu definisanu funkcijom sa dve promenljive, $z = f(x,y)$. Za procenu z , prvo moramo definisati skup (x,y) tačaka preko domena pomoću funkcije **meshgrid**.

Funkcija	Funkcija	Funkcija
<pre>[X,Y] = meshgrid(-2:.2:2); Z = X .* exp(-X.^2 - Y.^2); surf(X,Y,Z)</pre>	<pre>[X,Y,Z] = cylinder(4*cos(t)); mesh(X,Y,Z); title('X,Y,Z');</pre>	<pre>t = 0:pi/10:2*pi; [X,Y,Z] = cylinder(4*cos(t)); mesh(Y); title('Y');</pre>
		

MatLab - Prozori figura (bar3(Y))

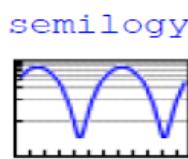
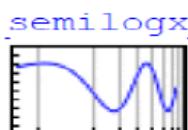
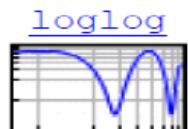
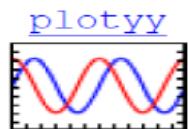
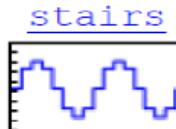
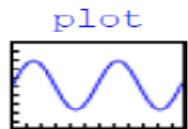
- **bar3(Y)** crta trodimenzionalni trakasti grafikon, gde svaki element u Y odgovara jednom redu. Kada je Y vektor, lestvica ose x kreće se od 1 do dužine(Y). Kada je Y matrica, lestvica ose x kreće se od 1 do veličine (Y,1) i elementi u svakom redu su grupisani zajedno.

Funkcija	Funkcija	Funkcija
<pre>load count.dat // ili formiramo vektor Y = count(1:10,:); figure bar3(Y) title('Odvojeni stil')</pre>	<pre>load count.dat Y = count(1:10,:); figure bar3(Y,'stacked') title('Naslagan stil')</pre>	<pre>load count.dat Y = count(1:10,:); figure bar3(Y,'grouped') title('Grupni stil')</pre>
		

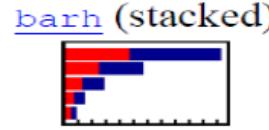
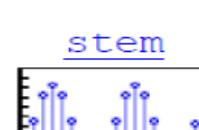
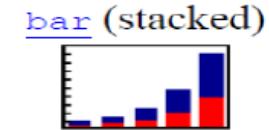
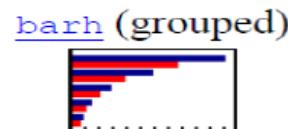
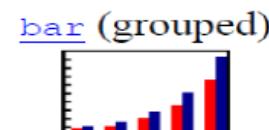
MatLab - Matrix Laboratory.

2D GRAFICI

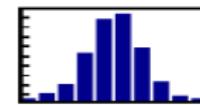
Line Graphs



Bar Graphs



[hist](#)



MatLab - Matrix Laboratory.

LINIJSKI GRAFIK `plot()`

Zavisnost $y(x)$

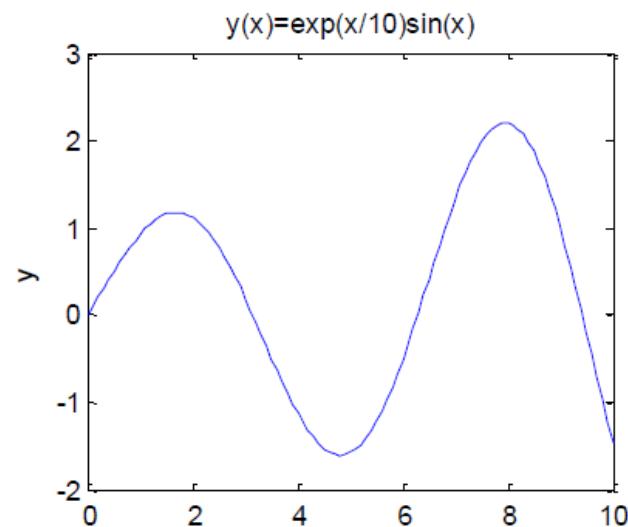
plot(x,y), xlabel('x'), ylabel('y'), title('naziv')

grid on /grid off

dodavanje mreže na grafiku

Primer. Nacrtati zavisnost $y(x)$ ako je $x=0, 0.1, 0.2, \dots 10$,
 $y=\exp(x/10)\sin(x)$.

```
x=0:0.1:10;  
y=exp(x/10).*sin(x);  
plot(x,y)  
xlabel('x')  
ylabel('y')  
title('y(x)=exp(x/10)sin(x)')
```



MatLab - Matrix Laboratory.

Zavisnost $y(x)$ sa kontrolom iscrtavanja linije

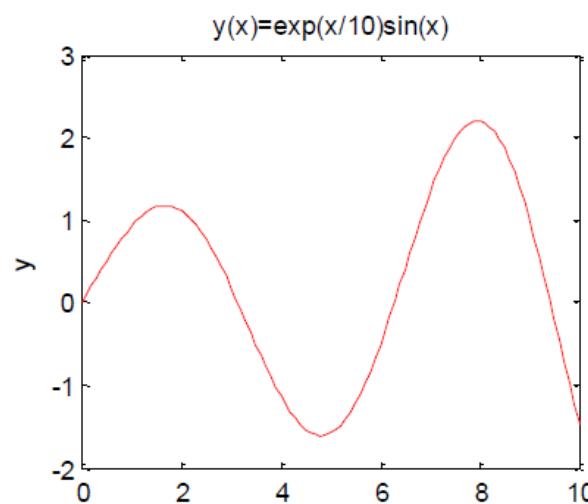
`plot(x,y,'karakteristika linije')`

Oznaka	Tip linije
-	Puna
:	Tačkasta linija
-.:	Crta tačka
--	Isprekidana linija

Oznaka	Boja
y	Žuta
m	Ljubičasta
c	Setoplava
r	Crvena

g	Zelena
b	Plava
w	Bela
k	Crna

```
x=0:0.1:10;
y=exp(x/10).*sin(x);
plot(x,y,'r--')
xlabel('x')
ylabel('y')
title('y(x)=exp(x/10)sin(x)')
```



Primer. Nacrtati funkciju iz prethodnog primjera i prikazati je isprekidanom linijom crvene boje.

MatLab - Matrix Laboratory.

Više zavisnosti na jednom grafiku ($y(x)$, $z(x)$, ...)

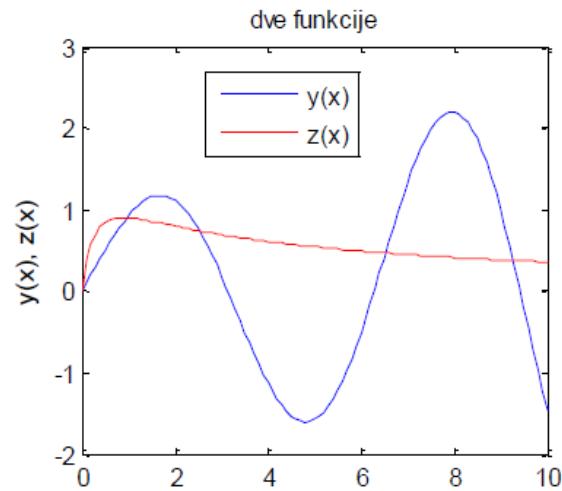
```
plot(x,y,x,z), legend('tekst1','tekst2')
```

Zavisnost $y(x)$ i $z(x)$ sa kontrolom iscrtavanja linija

```
plot(x1,y1,'r--', x2,y2,'bo')
```

Primer. Nacrtati zavisnosti $y(x)$ ako je $x=0,0.1,0.2, \dots 10$, $y=\exp(x/10)\sin(x)$, $z(x)=\log(x/2+1)/(x+1)$.

```
x=0:0.1:10;
y=exp(x/10).*sin(x);
z=log(5*x+1)./(x+1);
plot(x,y,x,z,:r')
xlabel('x')
ylabel('y(x), z(x)')
title('dve funkcije')
legend('y(x)','z(x)')
```

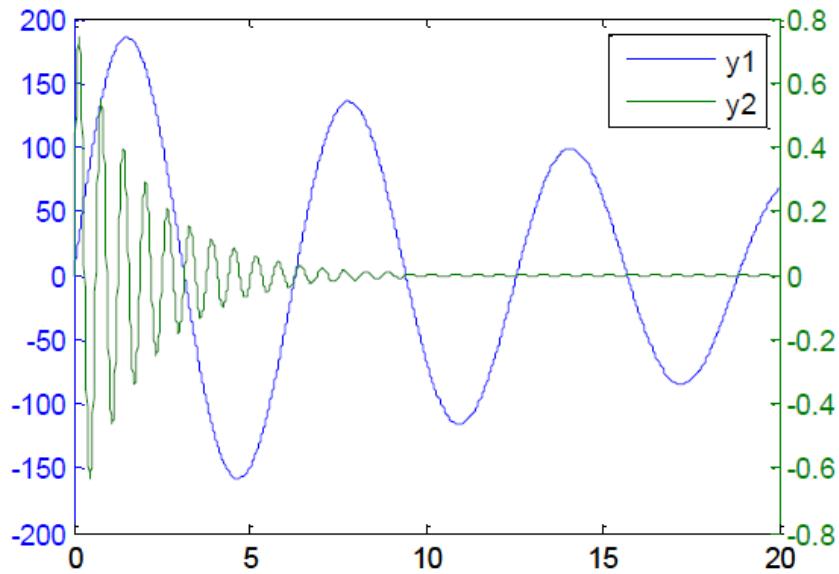


MatLab - Matrix Laboratory.

Dve y-ose na jednom grafiku - plotyy

```
plotyy(X1,Y1,X2,Y2,function)
```

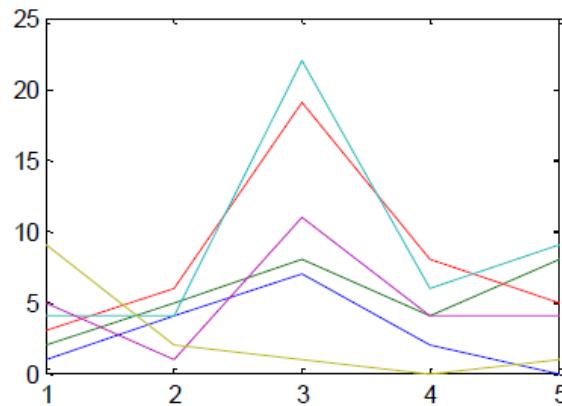
```
x = 0:0.01:20;  
y1 = 200*exp(-0.05*x).*sin(x);  
y2 = 0.8*exp(-0.5*x).*sin(10*x);  
plotyy(x,y1,x,y2,'plot');  
legend('y1','y2')
```



MatLab - Matrix Laboratory.

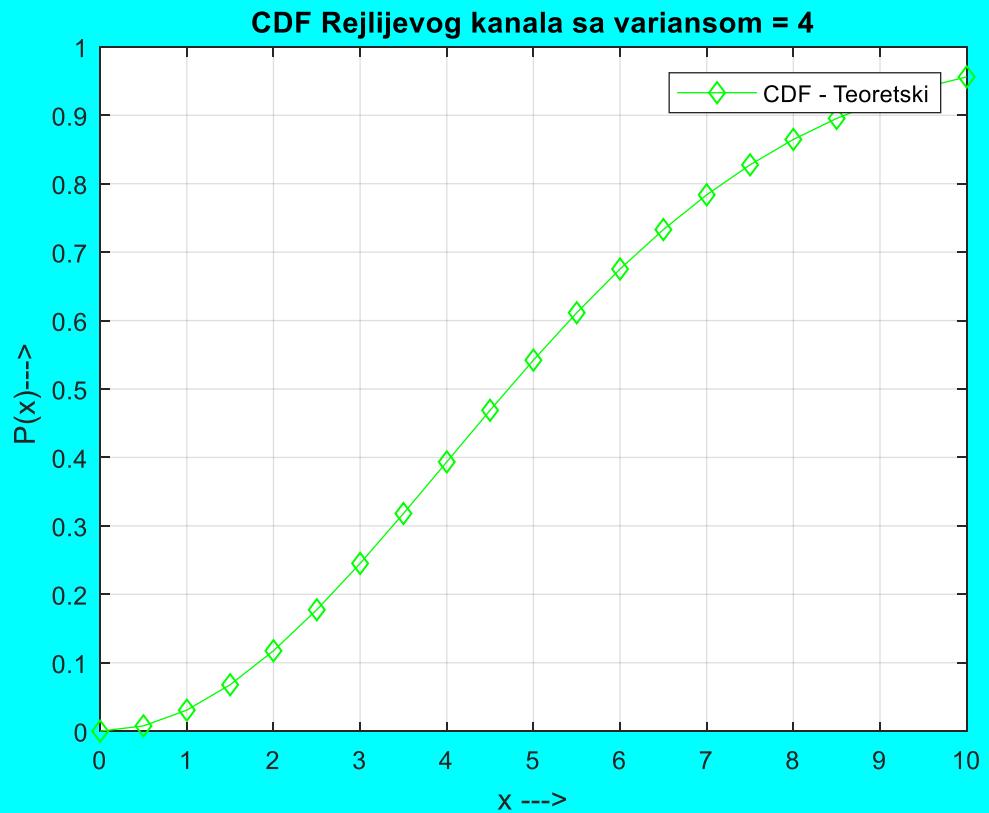
Primer. Grafički prikazati kolone matrice $A = \begin{bmatrix} 1 & 2 & 3 & 2 & 5 & 9 \\ 4 & 5 & 6 & 4 & 1 & 2 \\ 7 & 8 & 19 & 22 & 11 & 1 \\ 2 & 4 & 8 & 6 & 4 & 0 \\ 0 & 8 & 5 & 9 & 4 & 1 \end{bmatrix}$

```
A=[1 2 3 4 5 9; 4 5 6 4 1 2; 7 8 19 22 11 1; 2 4 8 6 4 0; 0 8 5 9 4 1];  
plot(A)
```



MatLab - Matrix Laboratory.

```
N=1000000; %Broj smplova za generisanje  
variance = 4; % Variansa Gausove slučajne promenljive  
%-----  
%Nezavisna Gausova promenljiva sa jediničnom  
x = randn(1, N);  
y = randn(1, N);  
%Anvelopa Rejljevog fedinga sa izabranom varijansom  
r = sqrt(variance*(x.^2 + y.^2));  
  
%Definisanje koraka i ranga za crtanje  
step = 0.5;  
range = 0:step:10;  
  
%Jednačina teoretskog CDF za Rejljev feding  
theoretical = 1 - exp(-range.^2/(2* variance ^2));  
% Crtanje figure u odgovarajućim osama  
%selektovanje osa  
%smeštanje figure u sistem koordinata  
plot(range,theoretical,'gd-');  
title('CDF Rejljevog kanala sa variansom = 4')  
legend('CDF - Teoretski ')  
xlabel('x --->');  
ylabel('P(x)--->');  
grid on;
```



MatLab - Matrix Laboratory.

Crtanje na logaritamskoj skali

semilogx
semilogy
loglog

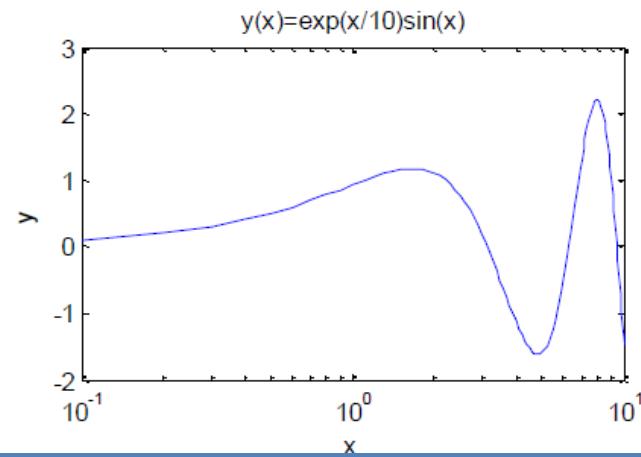
semilogx(x,y,'r');
semilogy(x,y,'r');
loglog(x,y,'r');

Nova slika

figure

figure, plot(x,y,'r');

```
x=0:0.1:10;  
y=exp(x/10).*sin(x);  
semilogx(x,y)  
xlabel('x')  
ylabel('y')  
title('y(x)=exp(x/10)sin(x)')
```



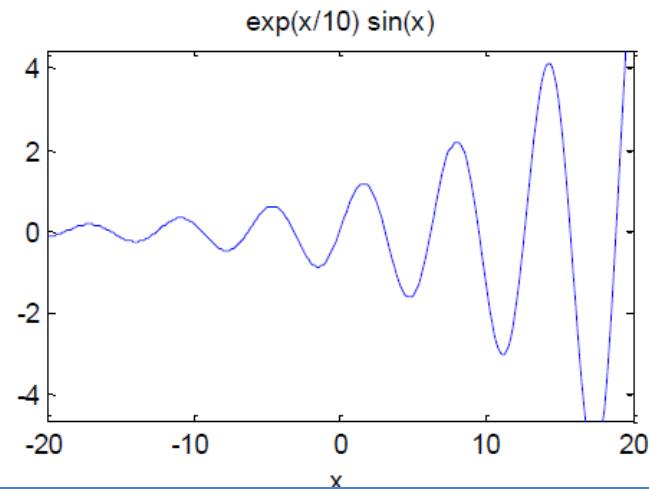
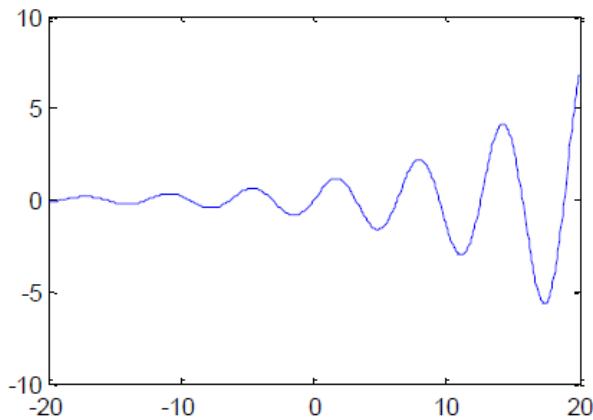
MatLab - Matrix Laboratory.

Grafik funkcije

```
fplot(fun,limits),    ezplot(fun,[xmin,xmax])
```

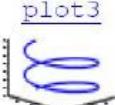
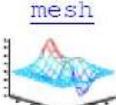
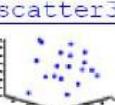
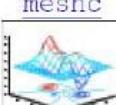
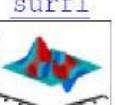
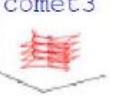
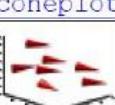
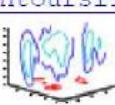
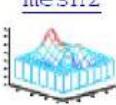
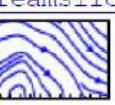
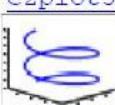
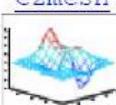
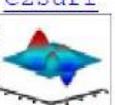
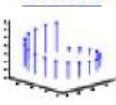
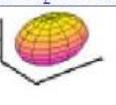
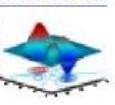
```
fplot('exp(x/10)*sin(x)',[-20 20])
```

```
ezplot('exp(x/10)*sin(x)',[-20 20])
```



MatLab - Matrix Laboratory.

3D GRAFIKA

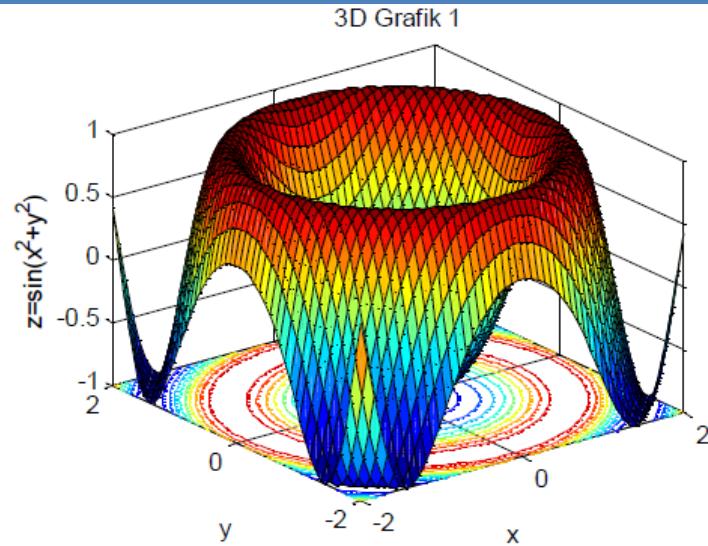
Line Graphs	Mesh Graphs and Bar Graphs	Area Graphs and Constructive Objects	Surface Graphs	Direction Graphs	Volumetric Graphs
<u>plot3</u> 	<u>mesh</u> 	<u>pie3</u> 	<u>surf</u> 	<u>quiver3</u> 	<u>scatter3</u> 
<u>contour3</u> 	<u>meshc</u> 	<u>fill3</u> 	<u>surfl</u> 	<u>comet3</u> 	<u>coneplot</u> 
<u>contourslice</u> 	<u>meshz</u> 	<u>patch</u> 	<u>surf</u> 	<u>streamslice</u> 	<u>streamline</u> 
<u>ezplot3</u> 	<u>ezmesh</u> 	<u>cylinder</u> 	<u>ezsurf</u> 	<u>streamribbon</u> 	
<u>waterfall</u> 	<u>stem3</u> 	<u>ellipsoid</u> 	<u>ezsurfc</u> 	<u>streamtube</u> 	

MatLab - Matrix Laboratory.

```
surf, surfc  
surface  
mesh  
contour  
contoursurf  
meshgrid
```

Primer. Nacrtati $z=\sin(x^2+y^2)$ na intervalu $-2 < x < 2$, $-2 < y < 2$

```
x1=linspace(-2,2,100);  
y1=linspace(-2,2,100);  
[x,y]=meshgrid(x1,y1);  
z=sin(x.^2+y.^2);  
surfc(x,y,z); xlabel('x'),  
ylabel('y')  
zlabel('z=sin(x^2+y^2)'),  
title('3D Grafik 1')
```



MatLab - Matrix Laboratory.

II. SYMBOLIC TOOLBOX

2. SIMBOLIČKO REŠAVANJE PROBLEMA – SYMBOLIC MATH TOOLBOX

Definicija simboličke promenljive

```
x=sym('x'), y=sym('y'),           --> syms x y  
x=sym('x','real'), y=sym('y','real') --> syms x y real  
c=sym('c','positive')               --> syms c positive  
                                --> syms c, assume(c>0)
```

Kreiranje simboličkih brojeva i matrica

```
a=sym('2/3'); mat=sym([1 2;3 4]);
```

Zamena simboličke varijable numeričkom vrednošću ili drugom simbol. var.

```
subs(S),          subs(S,new),          subs(S,old,new)
```

```
syms a b,  
subs(a+b,a,4)  
ans =  
b + 4
```

```
syms a b  
subs(a+b,{a,b},{2,3})  
ans =  
5
```

MatLab - Matrix Laboratory.

Pretvaranje numeričke matrice u simboličku matricu

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

```
bb=sym(b)  
bb =  
[ 1, 2, 3]  
[ 4, 5, 6]  
[ 7, 8, 9]
```

Određivanje simb. varijabli iz simb. izraza

```
f=sym('x^2+2*y^3-7*y+3+z');  
findsym(f)  
ans =  
x, y, z  
findsym(f,2)  
ans =  
x,y
```

MatLab - Matrix Laboratory.

Konverzija promenljivih

double konvertuje simboličke varijable u numeričke

poly2sym konvertuje numerički polinom u simbolički

sym2poly konvertuje simbolički polinom u numerički

```
syms a1 a2 a3 a4  
a=[a1 a2; a3 a4]  
a =  
[ a1, a2]  
[ a3, a4]  
  
b=double(a)  
b =  
1 2  
3 4
```

```
syms x  
f =x^2+2*x^3-6*x+3  
p=sym2poly(f)  
p =  
2 1 -6 3  
p2 =[1 2 3 4 5]  
f2=poly2sym(p2)  
f2 =  
x^4+2*x^3+3*x^2+4*x+5
```

MatLab - Matrix Laboratory.

Diferenciranje - **diff**

$$f(x) = x^2 \sin(x), \quad \frac{d^2 f(x)}{dx^2} = ?$$

```
syms x;
f=x^2*sin(x);
diff(f,2)
ans =
2*sin(x)+4*x*cos(x)-x^2*sin(x)
```

$$f(x,y) = x^2y + 2y^2x, \quad \frac{d^2 f(x,y)}{dxdy} = ?$$

```
syms x y;
f=x^2*y+2*y^2*x
diff(diff(f,x),y)
ans =
2*x+4*y
```

MatLab - Matrix Laboratory.

Integraljenje

$$f(x,y) = x^2y + 2y^2x,$$
$$\int f(x,y)dx = ?$$

```
syms x y;
f=x^2*y+2*y^2*x
int(f,x)
ans =
(x^2*y*(x + 3*y))/3
```

$$f(x,a) = e^{-ax^2}$$
$$\int_{-\infty}^{\infty} f(x)dx = ?$$

```
syms x, syms a, assume(a > 0);
f=exp(-a*x^2)
int(f,x,-inf,inf)
ans =
pi^(1/2)/a^(1/2)
```

$$f(x,a) = xe^{-x}$$
$$\int_a^b f(x)dx = ?$$

```
syms x a b;
f=x*exp(-x);
F=int(f,x,a,b)      % granice su parametri
F =
-b*exp(-b)-exp(-b)+a*exp(-a)+exp(-a)

subs(F, {a,b}, [0,10])  % smena granica
ans =
0.9995
```

$$\int_0^{10} f(x)dx = ?$$

MatLab - Matrix Laboratory.

Granične vrednosti

$$\lim_{x \rightarrow \infty} x \left(1 + \frac{a}{x}\right)^x \sin\left(\frac{b}{x}\right)$$

```
syms x a b;
f=x*(1+a/x)^x*sin(b/x);
limit(f,x,inf)
ans =
exp(a)*b
```

$$\lim_{x \rightarrow 0} \frac{1}{x}$$

$$\lim_{x \rightarrow 0^-} \frac{1}{x}$$

$$\lim_{x \rightarrow 0^+} \frac{1}{x}$$

```
syms x;
limit(1/x,x,0)
ans =
NaN

limit(1/x,x,0,'left')
ans =
-Inf

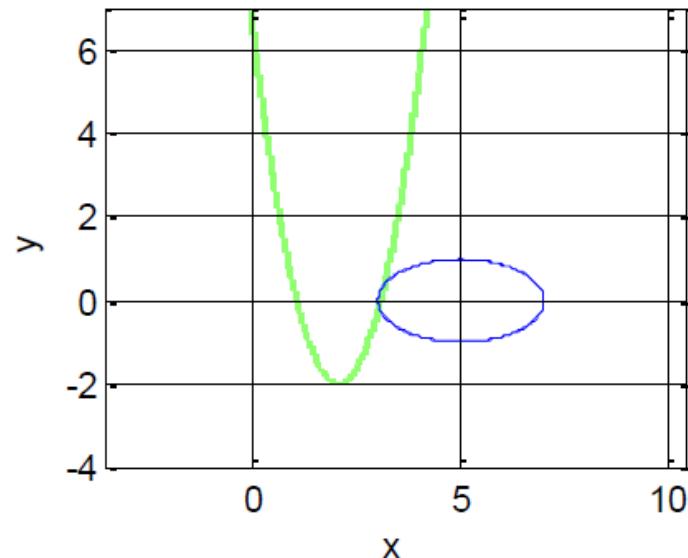
limit(1/x,x,0,'right')
ans =
Inf
```

MatLab - Matrix Laboratory.

Grafik simboličke funkcije

```
ezplot(f),  
ezplot(f,vektor_granica)  
ezplot(x,y,t_granica)
```

$$x = 2 \sin(2 t) + 5, y = \cos(2 t)$$



```
ezplot(sym(' -2*(x-2)^2+2+y'), [-2,8,-4,7])  
grid  
hold on  
ezplot(sym('2*sin(2*t)+5'), sym('cos(2*t)'))
```

MatLab - Matrix Laboratory.

ALGEBARSKE JEDNAČINE

- `solve(eq),`
- `solve(eq,var)`
- `solve(eq1,eq2,...,eqn)`

MatLab - Matrix Laboratory.

Primer. Rešiti i prikazati grafički sledeći sistem jednačina

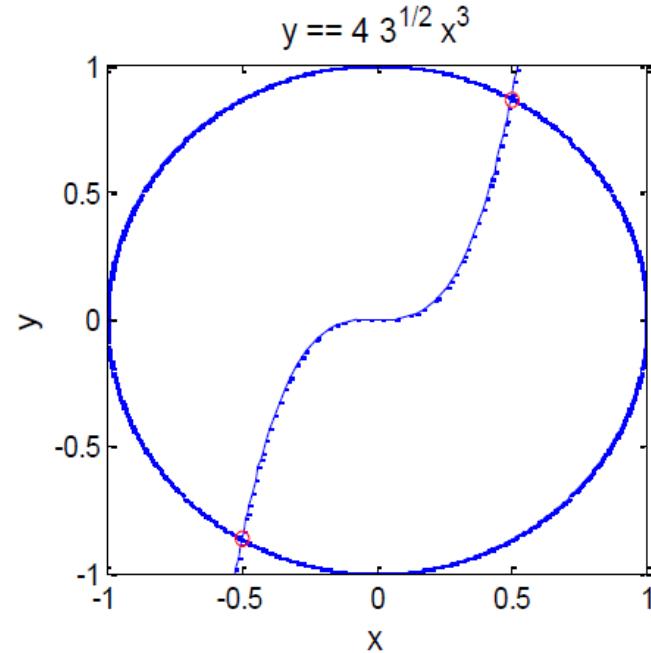
$$x^2 + y^2 - 1 = 0, \quad 0.75x^3 - y + 0.9 = 0$$

```
syms x y real
S=solve('x^2+y^2-1=0','y=4*sqrt(3)*x^3')
X=S.x, xx=double(X) % vector resenja po x
Y=S.y, yy=double(Y) % vector resenja po y
colormap([0 0 1])
ezplot(sym('x^2+y^2-1=0'),[-1,1,-1,1]),
hold on
ezplot(sym('y=4*sqrt(3)*x^3')),
plot(X(1),Y(1),'or',X(2),Y(2),'or')
hold off
```

```
S =
x: [2x1 sym]
y: [2x1 sym]
X =
1/2
-1/2
```

```
Y =
3^(1/2)/2
-3^(1/2)/2
xx =
0.5000
-0.5000
```

```
yy =
0.8660
-0.8660
```



MatLab - Matrix Laboratory.

DIFERENCIJALNE JEDNAČINE

dsolve(deq)

dsolve(deq,c1)

dsolve(deq1,deq2,c1,c2)

Primer 1. $y'(t) - y(t) = \sin(t)$

```
% Bez početnih uslova  
dsolve('Dy-y=sin(t)')  
ans =  
-1/2*cos(t)-1/2*sin(t)+exp(t)*c1
```

```
% Sa početnim uslovima  
dsolve('Dy = y + sin(t)', 'y(0)=1')  
ans =  
3/2*exp(t)-1/2*cos(t)-1/2*sin(t)
```

Primer 2. $f'(t) - f(t) - g(t) = 0, \quad g'(t) + f(t) - g(t) = 0$

```
S=dsolve('Df-f-g=0', 'Dg+f-g=0', 'f(0)=1', 'g(0)=2')  
S =  
f: [1x1 sym]  
g: [1x1 sym]  
S.f  
ans =  
exp(t)*cos(t) + 2*exp(t)*sin(t)  
S.g  
ans =  
2*exp(t)*cos(t) - exp(t)*sin(t)
```

MatLab - Matrix Laboratory.

REDOVI

`symsum(f), symsum(f,a,b)`

$$M = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots$$

```
syms k  
M=symsum(1/k^2,1,inf)  
M =  
pi^2/6
```

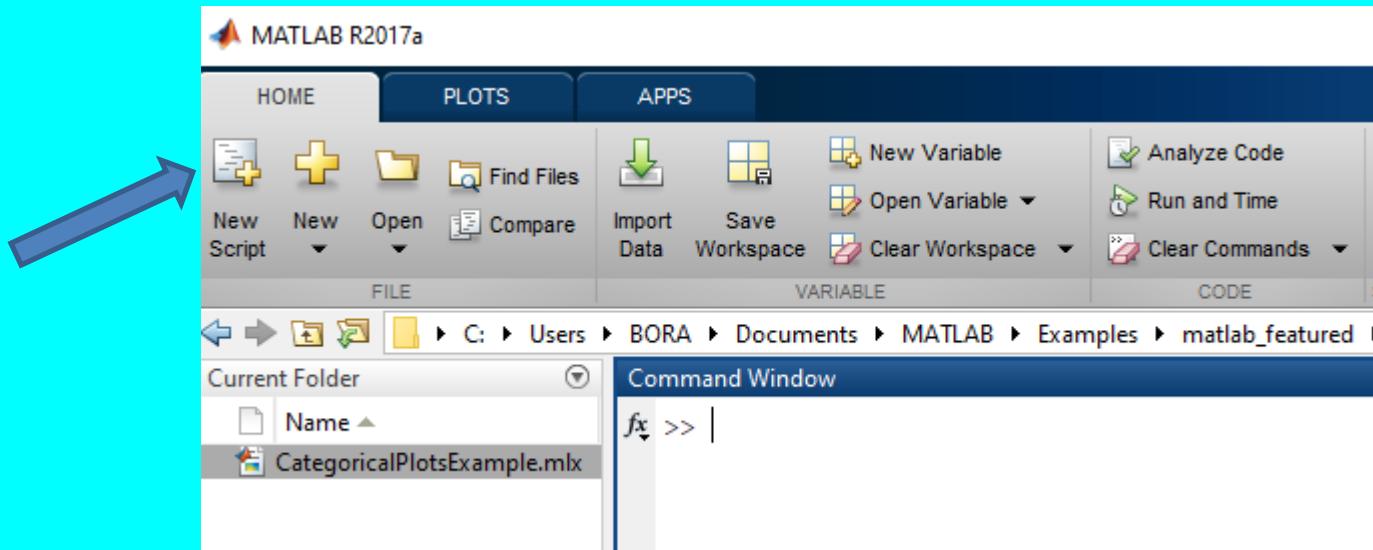
$$M = 1 + 2 + 3 + \dots + n$$

```
syms k n  
M=symsum(k,1,n)  
M =  
pi^2/6
```

MatLab - Matrix Laboratory.

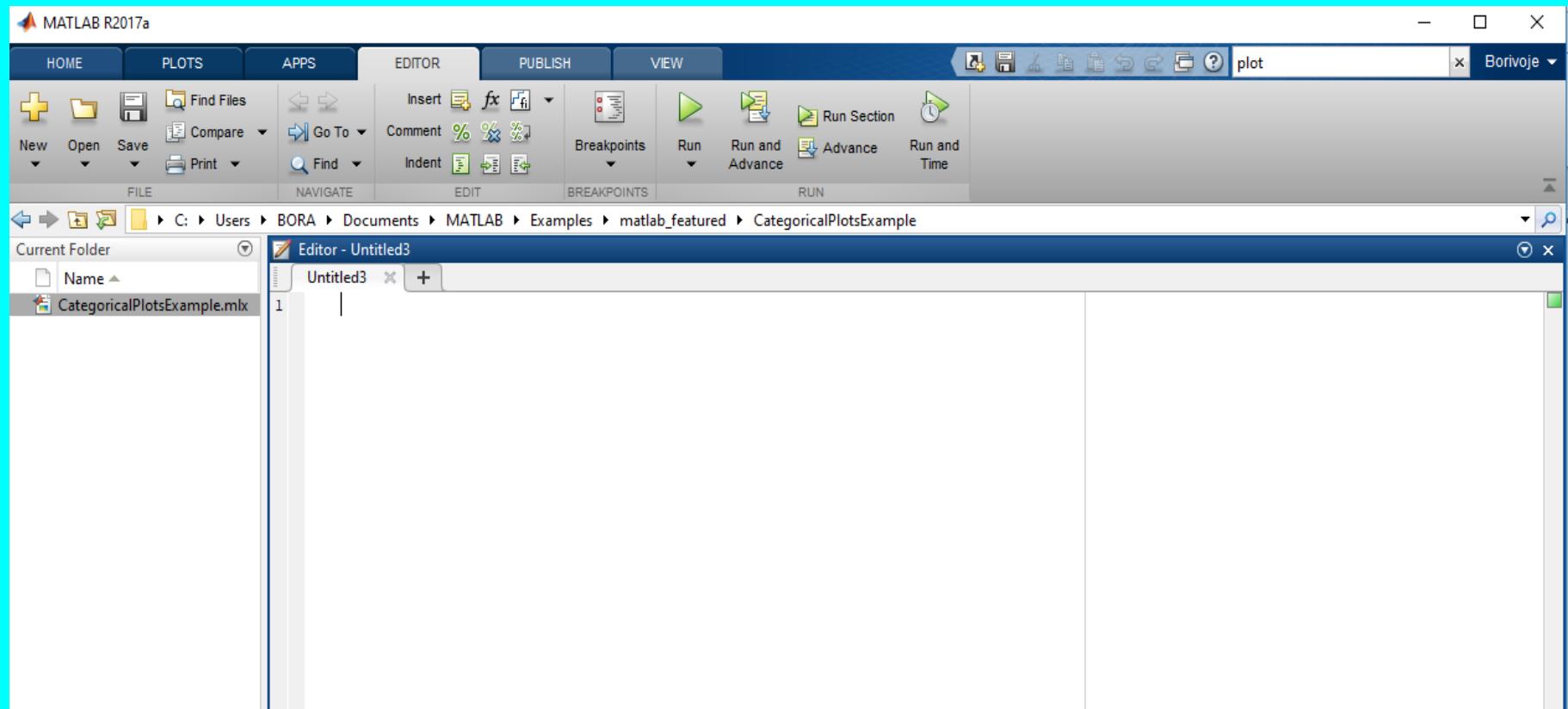
Programiranje u MATLAB-u

- Jednostavni MATLAB program MATLAB je potpun programski jezik u kojem je moguće napisati vlastite programske skriptove. Pojedine naredbe moguće je izvršiti uslovno ili ponoviti više puta. Pomoću ugrađenih funkcija i programskih paketa moguće je graditi nove programe. Svaki skup MATLAB naredbi napisan korišćenjem bilo kojeg teksta editora koji je zapamćen u datoteci sa nastavkom .m predstavlja jedan MATLAB program. Dakle, svi MATLAB programi se pamte u običnim tekstualnim datotekama a najjednostavnije ih je pisati korišćenjem ugrađenog editora koji se poziva naredbom **edit** ili direktno iz glavnog prozora ekranskim tasterom **New Script**:

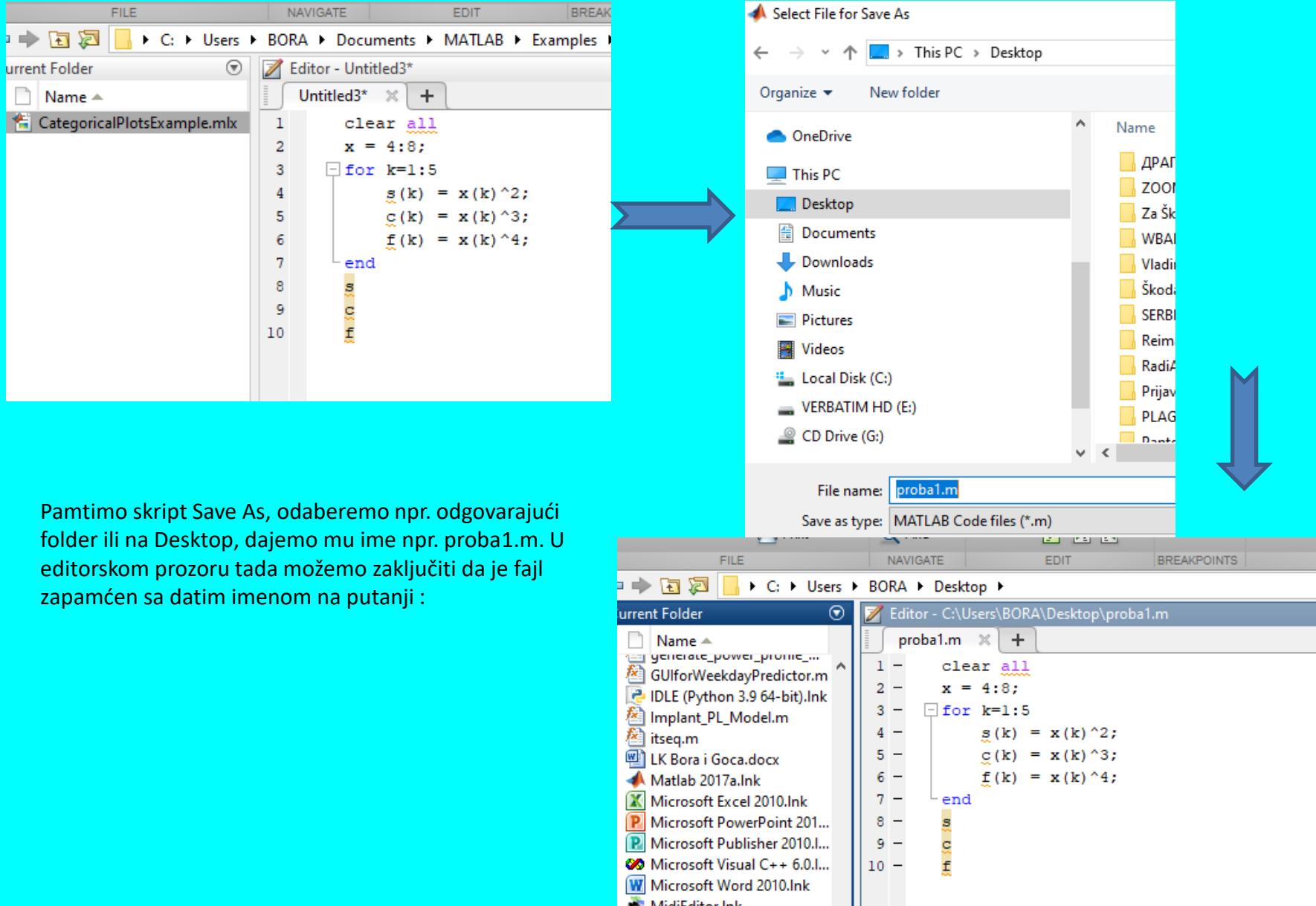


MatLab - Matrix Laboratory.

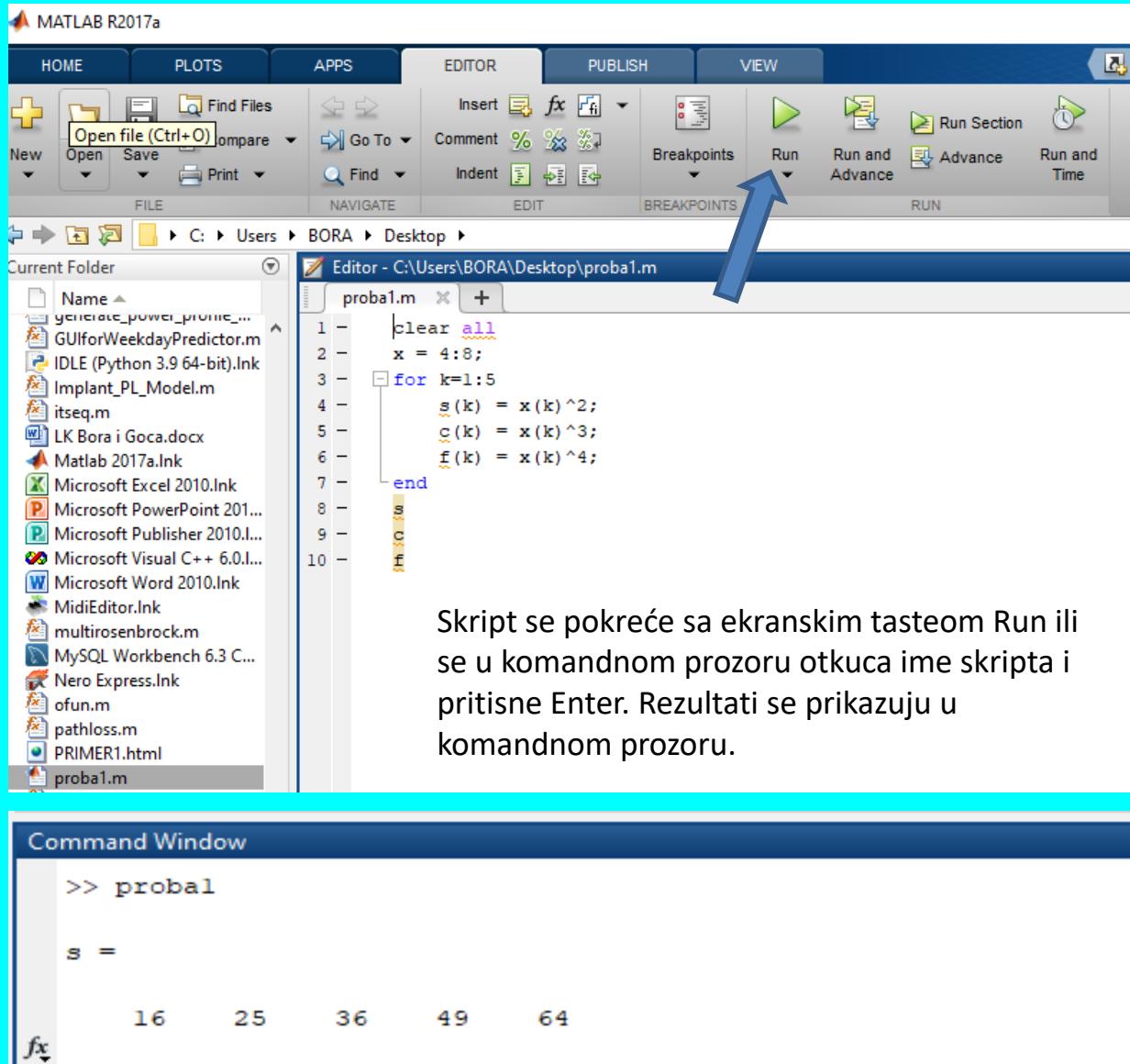
U glavnom prozoru se otvara prostor za editovanje, a u glavnom meniju jezičak sa komandama potrebnim za editovanje i izvršavanje skripta. Svaki skript koji smo zapisali mora se prvo zapamtiti na odgovarajućem direktorijumu pa onda tek izvršiti komandom **Run**. Matlab editor je u stanju da vam ukaže i označi mesta na kojima ste napravili grešku.



MatLab - Matrix Laboratory.



MatLab - Matrix Laboratory.



Skript se pokreće sa ekranskim tasteom Run ili se u komandnom prozoru otkuca ime skripta i pritisne Enter. Rezultati se prikazuju u komandnom prozoru.

Skripta vs. Funkcija

Skripte naspram funkcija

Ova tema govori o razlikama između skripti i funkcija i pokazuje kako pretvoriti skriptu u funkciju. I skripte i funkcije omogućuju nam ponovno korišćenje nizova naredbi zapamćenim u programskim datotekama. Skripte su najjednostavniji tip programa, budući da memorišu naredbe tačno onako kako smo ih upisali u naredbenu liniju. Međutim, funkcije su fleksibilnije i lakše proširive.

Ako napišemo skriptu u datoteci pod nazivom npr. povrtro.m koja izračunava površinu trougla:

```
b = 5; h = 3; a = 0,5*(b.*h)
```

Nakon što zapamtimo datoteku, možemo izvršiti skriptu iz komandnog prozora sa kucanjem njenog imena ili tasterom Run i ona će se ako nema grešaka izvršiti. Ali ovaj primer važi samo za pojedinačne slučajeve.

```
povrtro  
a = 7,5000
```

Šta ako moramo da promenimo vrednosti promenljivih?

Da bi izračunali površinu drugog trougla pomoću iste skripte, moramo ažurirati nove vrednosti b i h u skripti i ponovno je pokrenuti. Međutim, umesto ručnog ažuriranja skripte svaki put, svoj program možete učiniti fleksibilnijim pretvaranjem u funkciju. Zamenite izraze koji dodjeljuju vrednosti b i h izjavom deklaracije funkcije. Deklaracija uključuje ključnu reč funkcije, nazine ulaznih i izlaznih argumenata i naziv funkcije (može biti i bez argumenata).

```
function a = povrtro(b,h)  
a = 0.5*(b.*h);
```

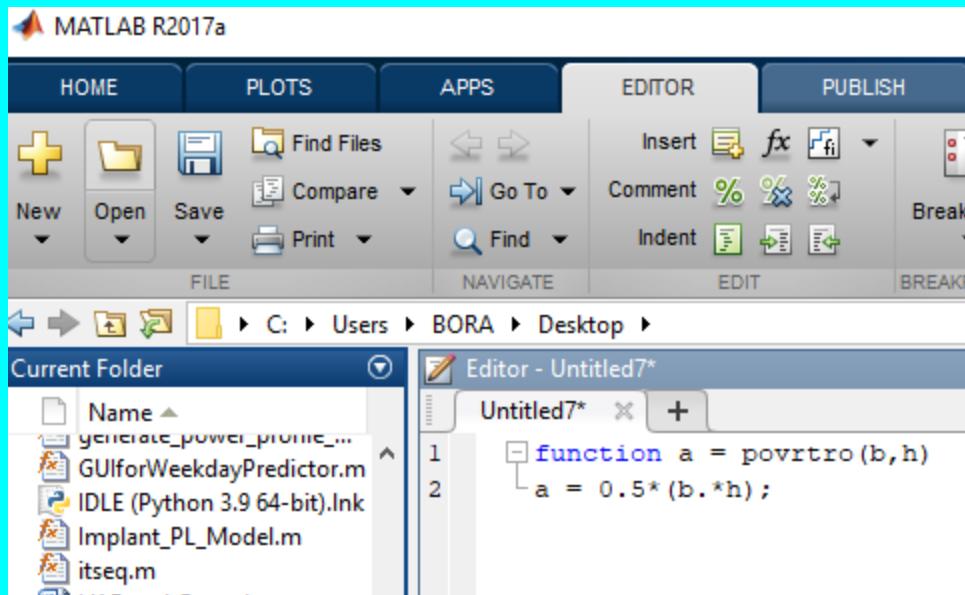
Nakon što zapamtimo datoteku, možemo pozvati funkciju sa različitim vrednostima osnove i visine iz komandnog retka bez menjanja skripte:

```
a1 = povrtro(1,5) a2 = povrtro(2,10) a3 = povrtro(3,6) pa respektivno dobijamo
```

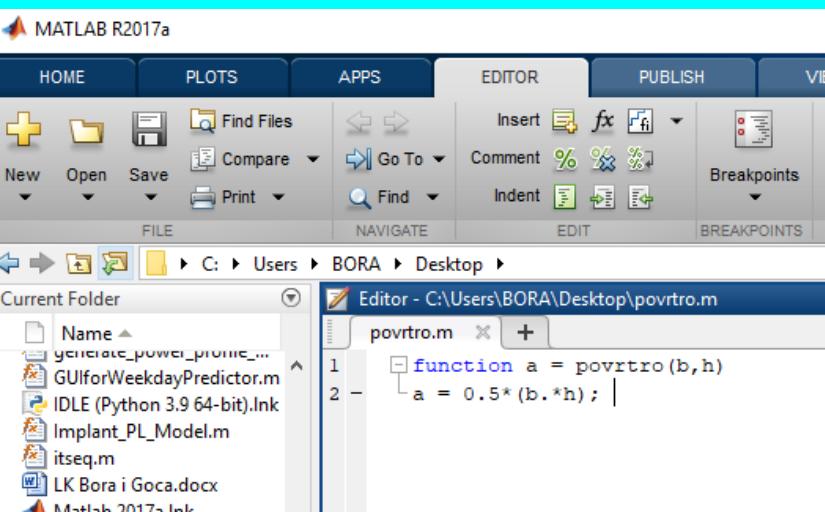
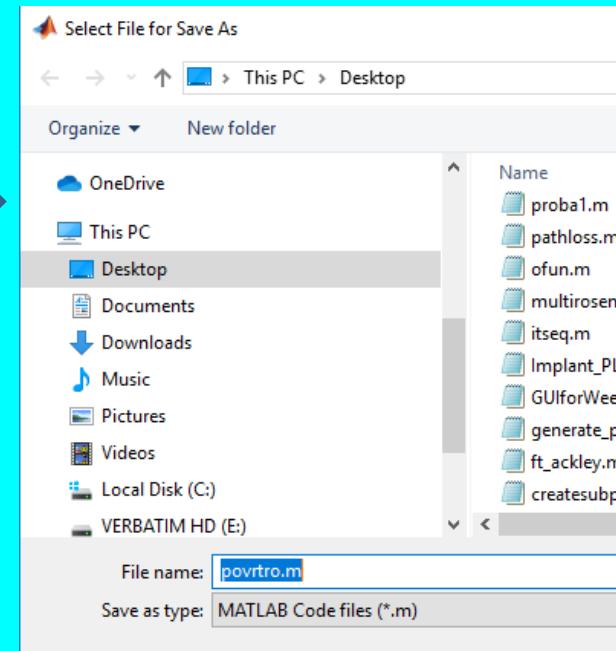
```
a1 = 2,5000 a2 = 10 a3 = 9
```

Funkcije imaju svoj radni prostor, odvojen od osnovnog radnog prostora a na folderu imaju oznaku fx. Važe znači za opšti slučaj I mogu se koristiti u bilo kom programu.

Skript vs. Funkcija



Taster Save i ime funkcije se automatski prikazuje u prozoru File name



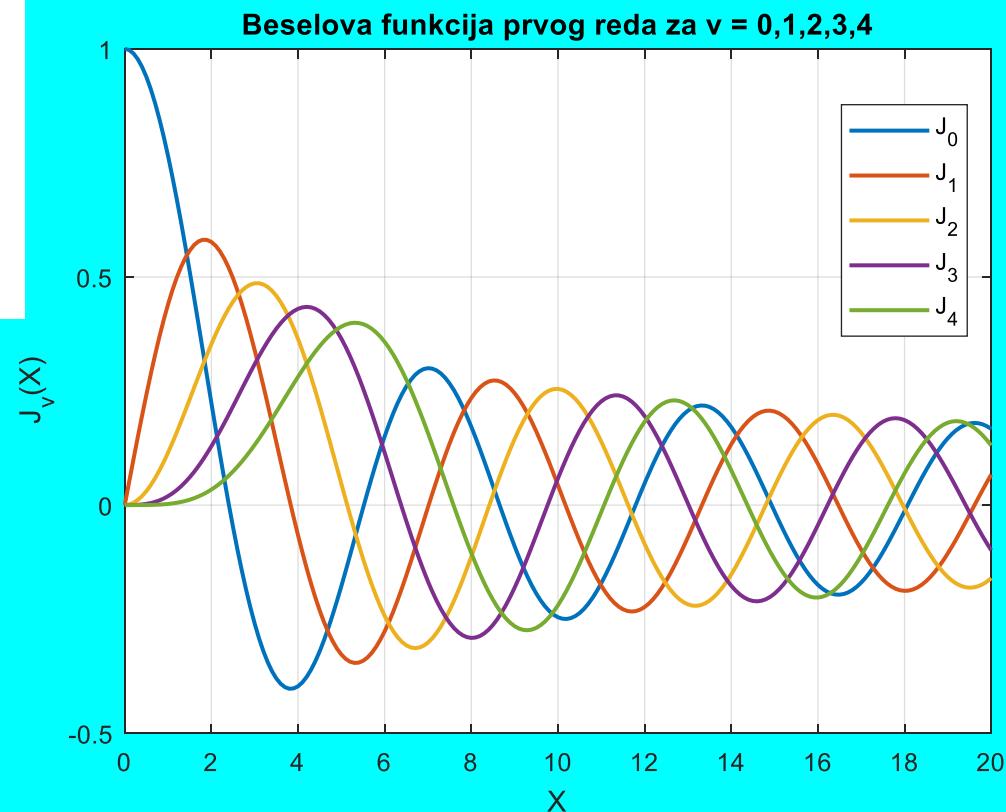
```
>> povrtro(3,4)
ans =
6
>> rezultat=povrtro(3,4)
rezultat =
6
```

Funkcije se pokreće imenom i izabranim parametrima u komandnom prozoru.

Matlab - komandni prozor primeri

Command Window

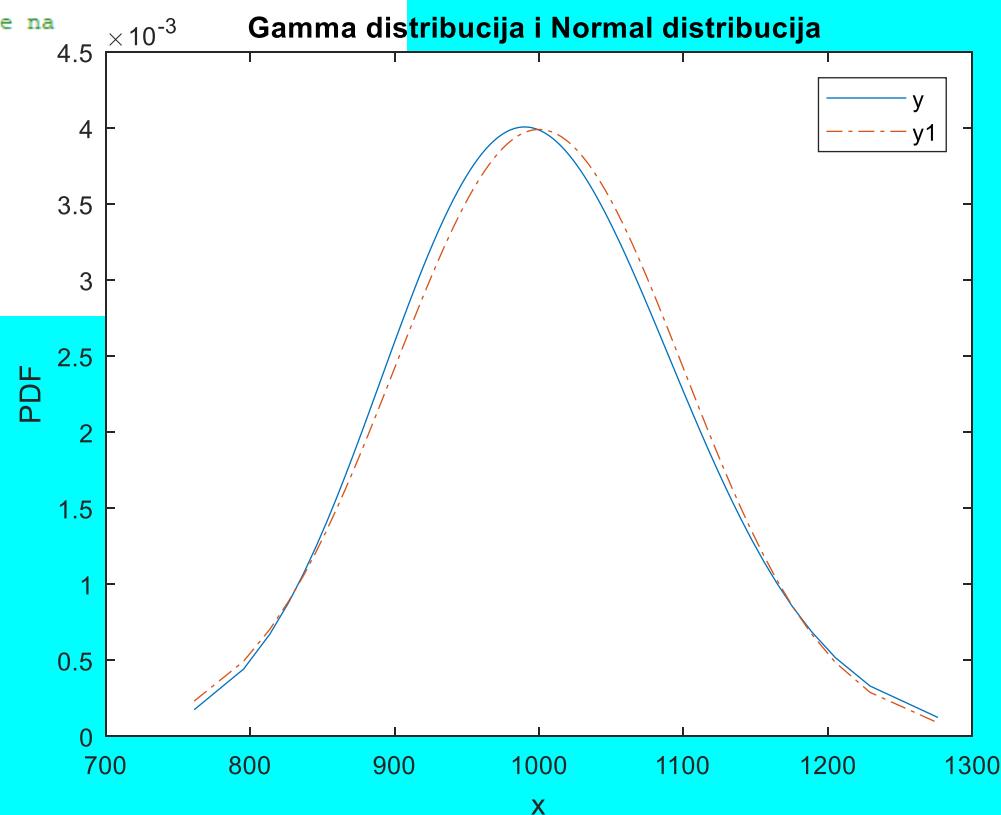
```
>> %Definisanje domena
X = 0:0.1:20;
% Izračunavanje prvih pet Beselovih
% funkcija prvog reda
J = zeros(5,201);
for i = 0:4
J(i+1,:) = besselj(i,X);
end
% Crtanje rezultata
plot(X,J, 'LineWidth',1.5)
axis([0 20 -.5 1])
grid on
legend('J_0','J_1','J_2','J_3','J_4', 'Location', 'Best')
title('Beselova funkcija prvog reda za v = 0,1,2,3,4')
xlabel('X')
ylabel('J_v(X)')
fx >> |
```



Matlab - komandni prozor primeri

Command Window

```
>> %% Izračunavanje gustine verovatnoće pdf  
%Gamma distribucije  
  
%%  
% Izračunavanje gustine verovatnoće Gamma distribucije sa parametrima |A = 100| i  
% |B = 10|. Za uporedjenje, izračunaće se pdf normalne distribucije  
% sa parametrima |mu = 1000| and |sigma = 100|.   
x = gaminv((0.005:0.01:0.995),100,10);  
y = gampdf(x,100,10);  
y1 = normpdf(x,1000,100);  
  
%%  
% Crtanje pdf Gamma distribucije i Normal distribucije na  
% istoj figuri.  
figure;  
plot(x,y,'-',x,y1,'-.')  
xlabel('x') ;  
ylabel('PDF');  
legend('y', 'y1')  
title('Gamma distribucija i Normal distribucija ' )  
fx >> |
```



Matlab - komandni prozor primeri

$\int \frac{-2x}{(1+x^2)^2} dx$	$\int \frac{x}{(1+z^2)} dz ; \int \frac{x}{(1+z^2)} dz$	$\int_0^1 x * \log(1+x) dx$	MATEMATIKA
<pre>syms x int(-2*x/(1 + x^2)^2)</pre>	<pre>syms x z int(x/(1 + z^2), x) int(x/(1 + z^2), z)</pre>	<pre>syms x int(x*log(1 + x), 0, 1)</pre>	MATLAB
<pre>ans = 1/(x^2 + 1)</pre>	<pre>ans = x^2/(2*(z^2 + 1)) ans = x*atan(z)</pre>	<pre>ans = 1/4</pre>	REZULTAT

Programiranje u C ili C++ jeziku

- MATLAB nam omogućuje pisanje programa i u nekom od običnih programskih jezika. Ovde ćemo ukratko pokazati kako napisati novu funkciju u programskom jeziku C.
- Razmotrimo najpre jednostavan primer koji samo ispisuje pozdravnu poruku i broj argumenata:

```
#include <mex.h>
/* Jednostavni MEX program */
void mexFunction(int nlhs, mxArray *plhs[],
int nrhs, const mxArray *prhs[])
{ mexPrintf("Hello MATLAB!\n");
if (0 != nrhs)
{ mexPrintf("Imamo %d ulaznih parametara.\n", nrhs); }
/* if */
if (0 != nlhs)
{ mexPrintf("Imamo %d izlaznih parametara.\n", nlhs); }
/* if */
return;
/* mexFunction */
```

Kada pišemo program za MATLAB ulazna funkcija nije main već je mexFunction. Toj funkciji MATLAB prosleđuje broj ulaznih (nrhs) i izlaznih (nlhs) argumenata te same argumente kao polja pokazivača (*plhs i *prhs).

Kada smo napisali program potrebno ga je zapamtitи као C program i prevesti. MATLAB se isporučује са слободном доступним обичним C преводицем LCC којег можемо искористити за превођење програма. Нека се наš програм налази у датотeci **hello.c**. Преводимо га коришћењем нaredbe **mex** i izvršавамо:

```
» mex hello.c <ENT>
» what <ENT>

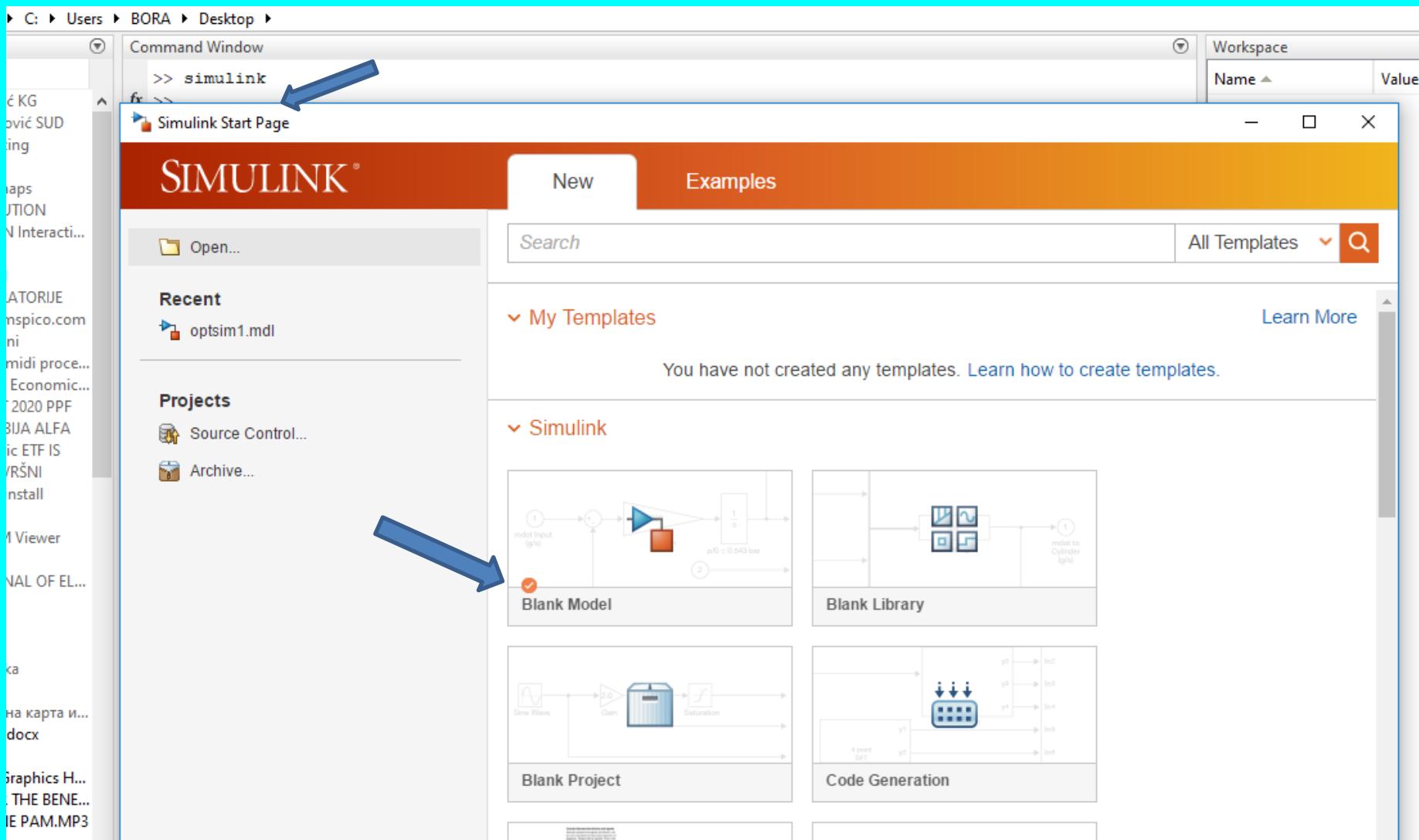
» hello <ENT>
Hello MATLAB!
```

```
% prevodimo program hello.c
% u trenutnom директорију сада постоји m i
% MEX функција истог имена
% u tom slučaju се увјејк poziva MEX функција
```

MATLAB Simulink

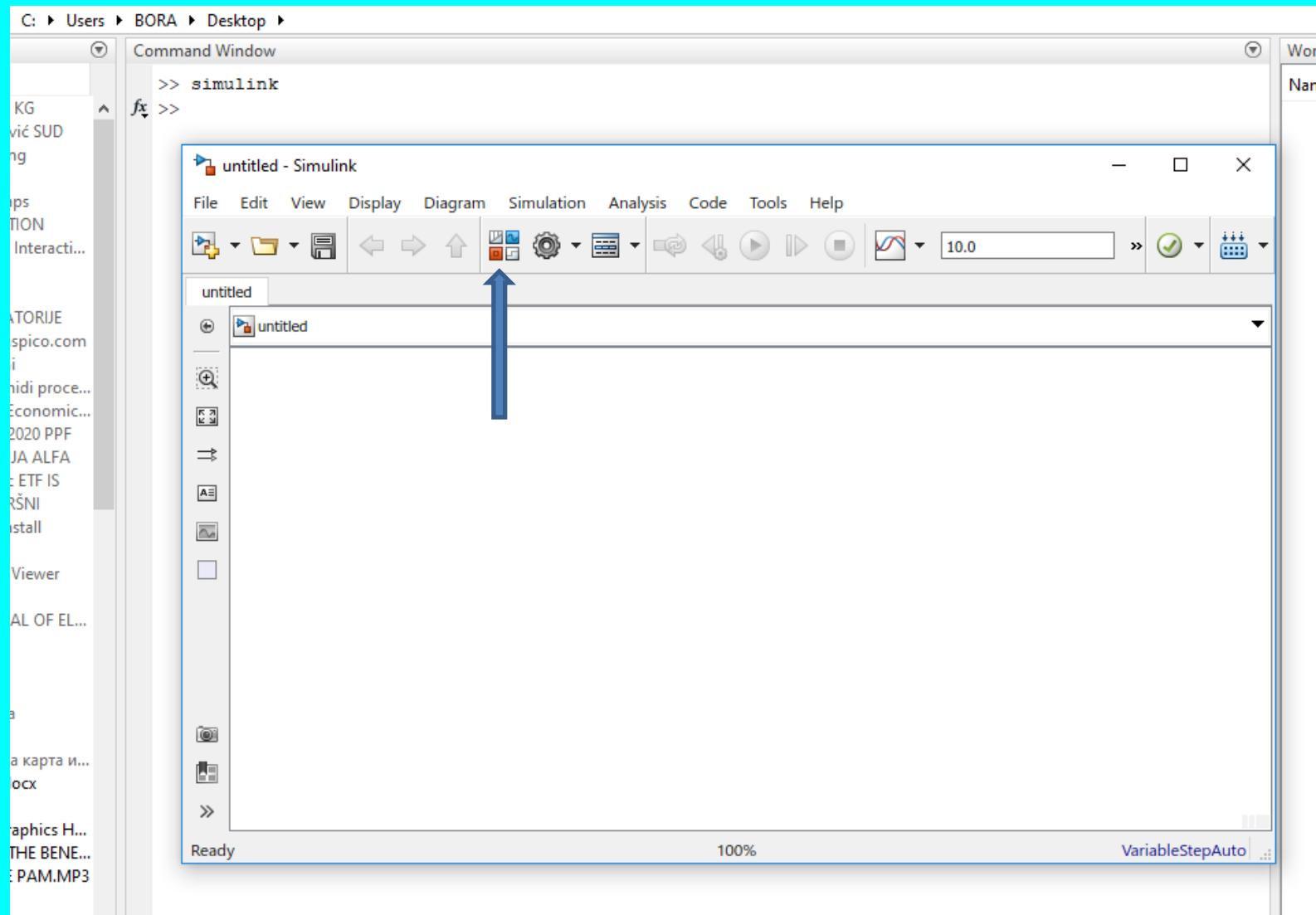
- Simulink je deo MATLAB-a namenjen simuliranju dinamičkih sistema. Za sam unos i opis sistema koji se simulira koristi se jednostavno grafičko okruženje u kome sastavljamo/crtamo model kombinovajući gotove komponente. Takvim pristupom je simulacija sistema značajno olakšana jer se od korisnika ne zahteva unos diferencijalnih ili diferencijskih jednačina koje opisuju sistema uz poznavanje MATLAB programskog jezika, već je dovoljno znati blok-schemu sistema.
- Simulink se pokreće unutar MATLAB-a zadavanjem naredbe **simulink** ili odabirom ikone iz alatne trake. Nakon pokretanja Simulinka otvara se prozor Simulink Library Browser prikazan na slici.
- Simulink je vremenski baziran softverski paket koji je uključen u Matlab i njegov je glavni zadatak numeričko rešavanje obične diferencijalne jednačine (ODE). Potreba za numeričkim rešenjem proizlazi iz činjenice da ne postoji analitičko rešenje za sve DE, posebno za one koje su nelinearne.
- Cela ideja je razbiti ODE na male vremenske segmente i numerički izračunati rešenje samo za mali segment. Dužina svakog segmenta naziva se "veličina koraka". Budući da je metoda numerička, a ne analitička, doći će do pogreške u rešenju. Pogreška zavisi od specifične metode i veličine koraka (obično označeno sa h).

MATLAB Simulink



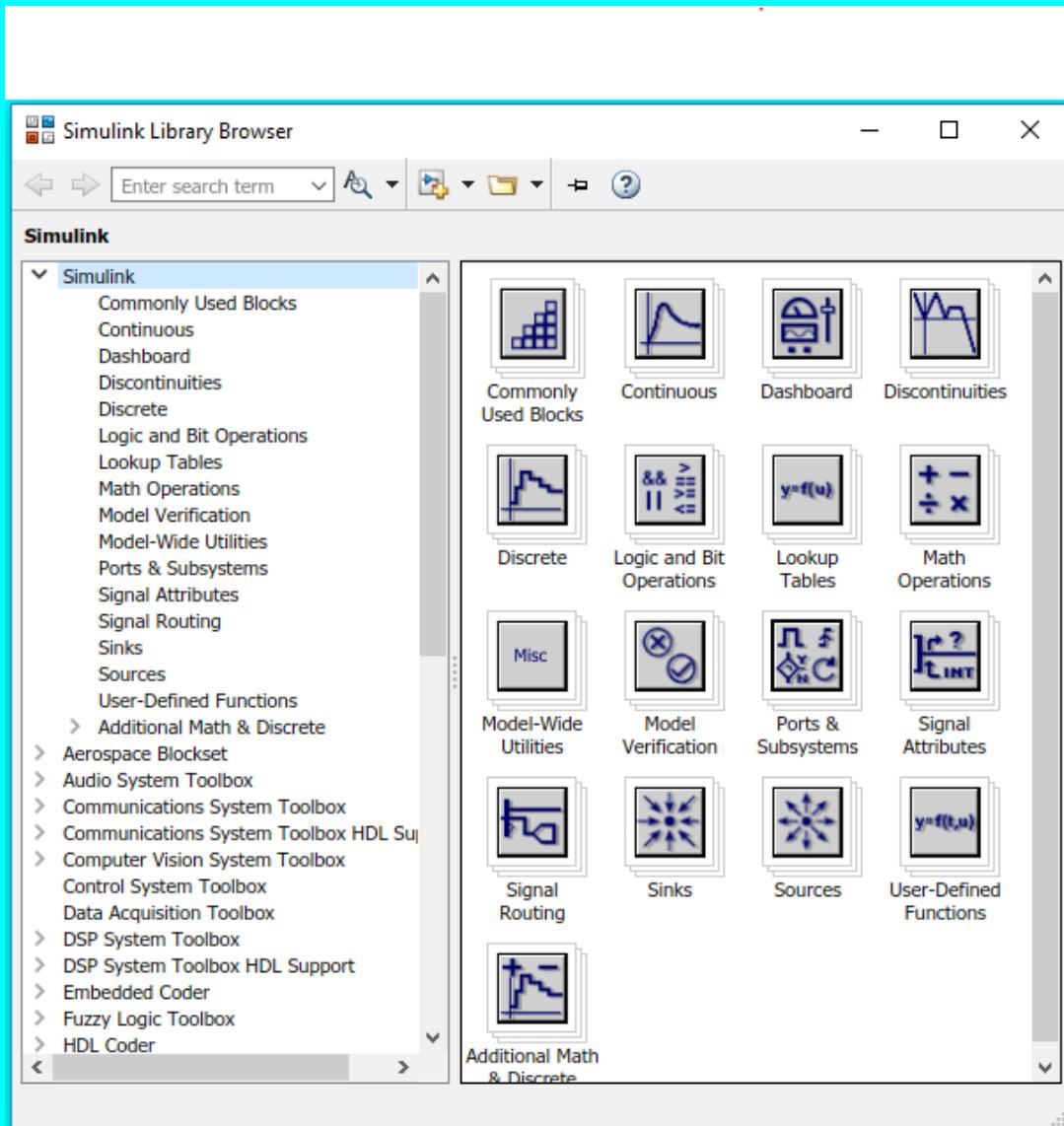
Da bi kreirali zadati model kliknemo na Blank model:

MATLAB Simulink



Za pregled dostupnih biblioteka u Simulinku, kliknite Library browser, otvorice se prozor:

MATLAB Simulink

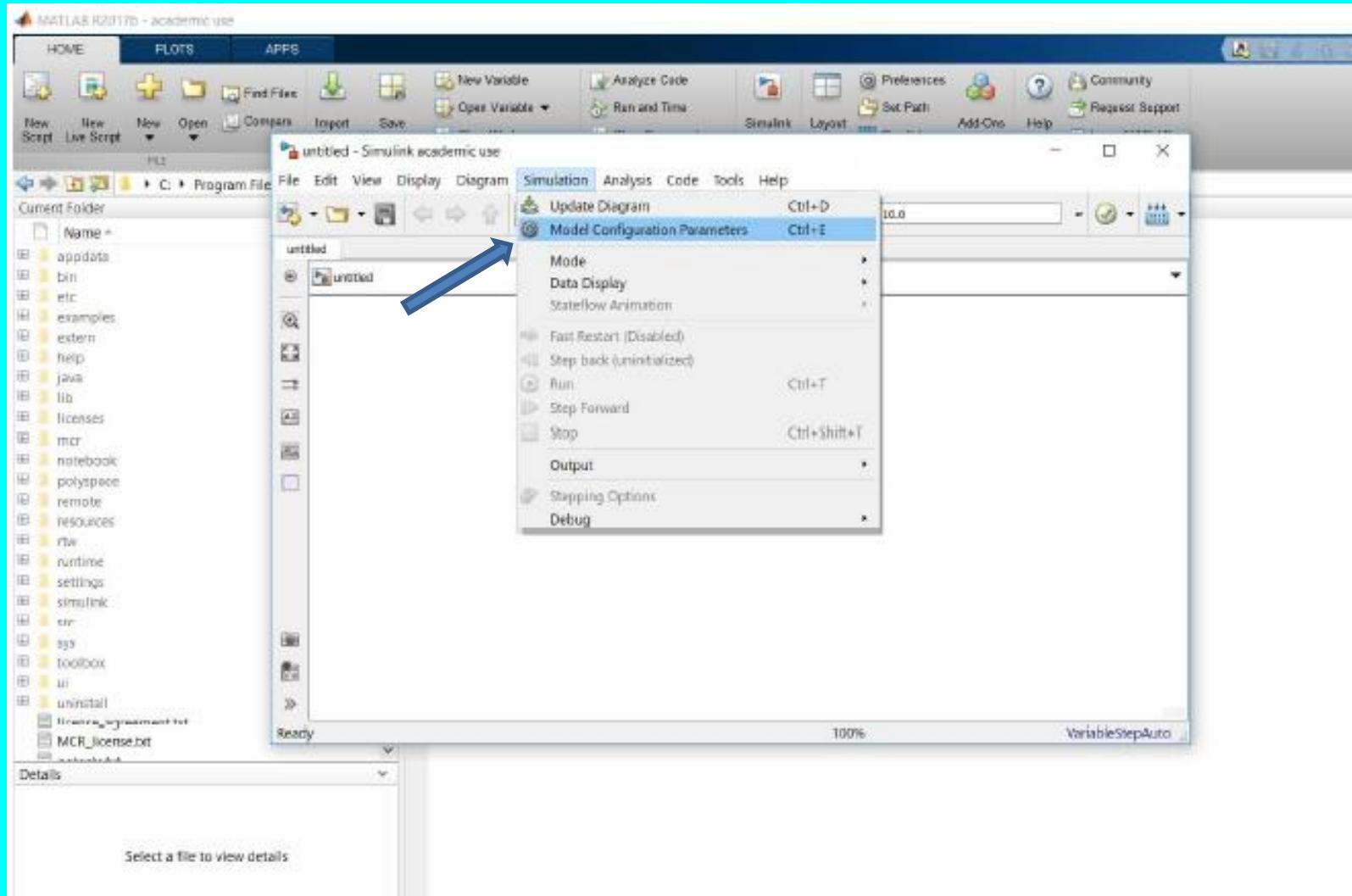


Ovo su biblioteke Simulink-a. Kao što se može videti, ima ih mnogo, a sub-biblioteka još više. Kako bi mogli pronaći odgovarajuće blokove, morate provesti neko vreme tražeći te biblioteke.

Posle nekog vremena moći ćete brzo pronaći sve blokove koji vam mogu zatrebatи.

Najvažniji izbornik koji morate znati je izbornik parametara koji može biti:

MATLAB Simulink



Pojavlje se prozor:

MATLAB Simulink

Configuration Parameters: untitled/Configuration (Active)

★ Commonly Used Parameters All Parameters

Select: Solver

Simulation time

Start time: 0.0 Stop time: 10.0

Solver options

Type: Variable-step Solver: auto (Automatic solver selection)

Additional options

Ovde možemo definisati vreme početka i zaustavljanja simulacije i opcije rešavača gde možemo odabrati varijabilnu ili fiksnu veličinu koraka, metodu rešavanja i veličinu koraka.
Ako odaberemo promenjivu veličinu koraka, zapamtite da minimalna veličina koraka mora biti manja od maksimalne.
Rešimo sada vrlo jednostavan ODE (Ordinary Differential Equations).

OK Cancel Help Apply

MATLAB Simulink - Primer 1

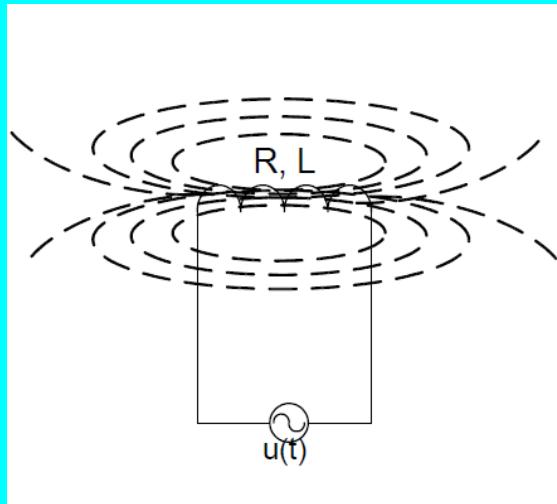
Razmotrite zavojnicu prikazanu na sledećoj slici.

Napon napajanja je jednak:

$$u(t) = i(t)R + \frac{d\psi(t)}{dt}.$$

Pod pretpostavkom da je induktivnost zavojnice konstantna gore navedena jednačina postaje:

$$u(t) = i(t)R + L \frac{di(t)}{dt}.$$

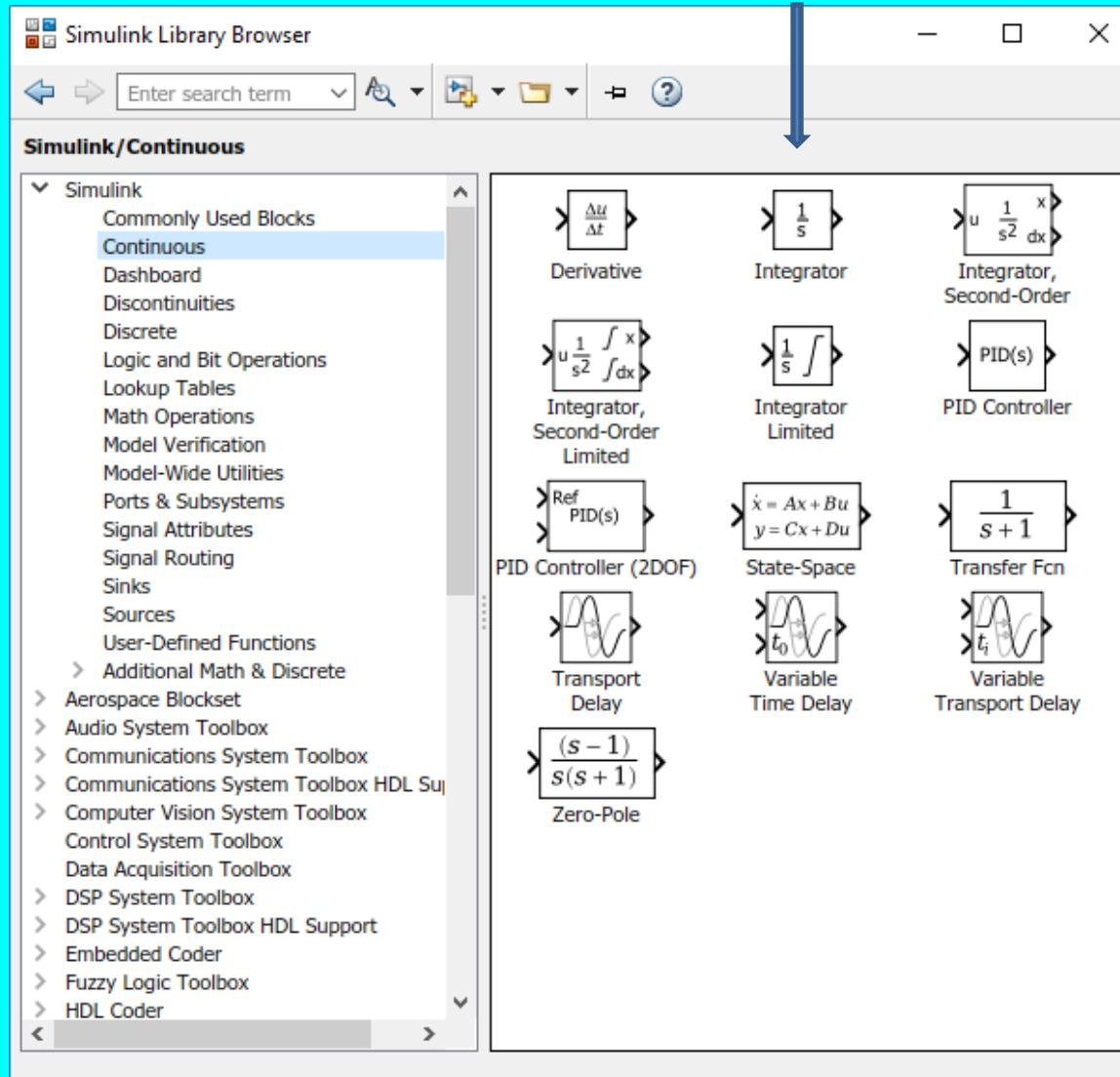


Ovo je linearna ODE prvog reda. Kakav je odgovor struje na naglu promenu napona, uz pretpostavku nultog početnog uslova? Da bi odgovorili na ovo, moramo rešiti gore navedenu ODA jednačinu. Postoje različiti načini da se to reši (Laplace...). Ovde ćemo to pokušati rešiti numerički uz Simulink.

Korak 1: Pre svega moramo izdvojiti najvišu derivaciju :
Korak 2: Koristićemo onoliko integratora koliki je redosled DE-a koji želimo rešiti: Integratorski blok je u:

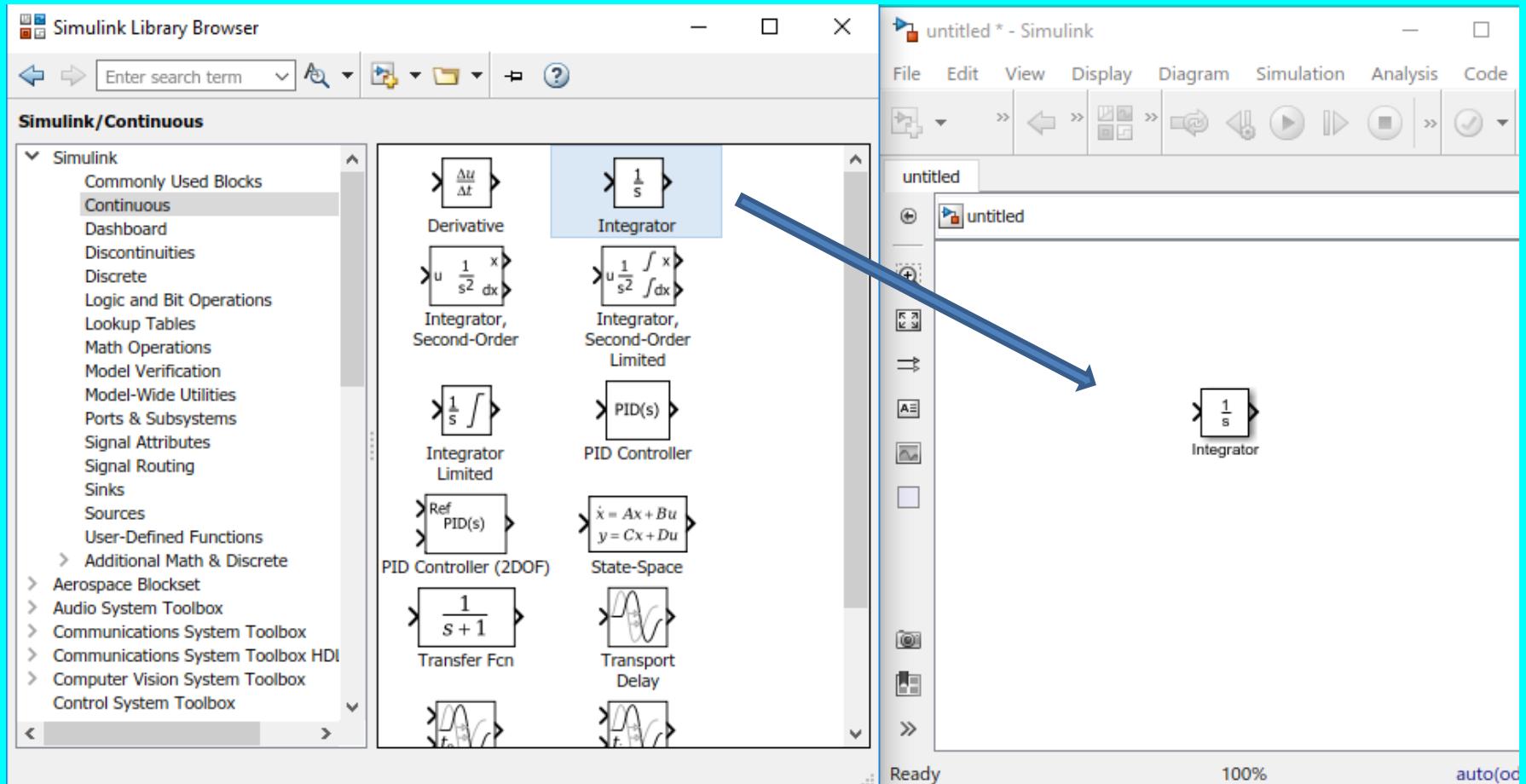
$$\frac{di(t)}{dt} = \frac{1}{L} (u(t) - i(t)R)$$

MATLAB Simulink - Primer 1



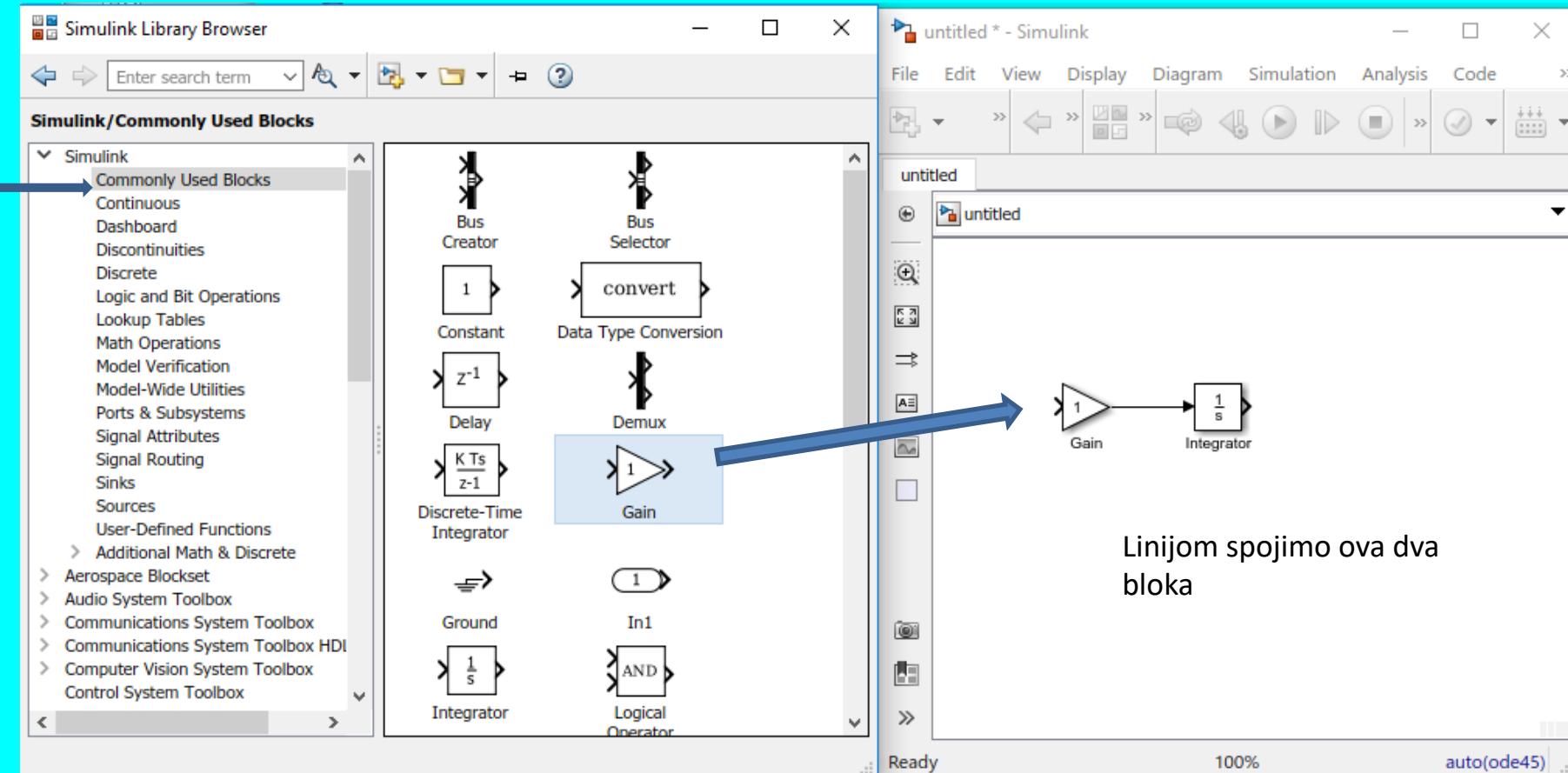
Samo kliknuti na model **Integrator** i prevući ga u već otvorenu radnu površinu našeg, za sada, praznog modela

MATLAB Simulink - Primer 1



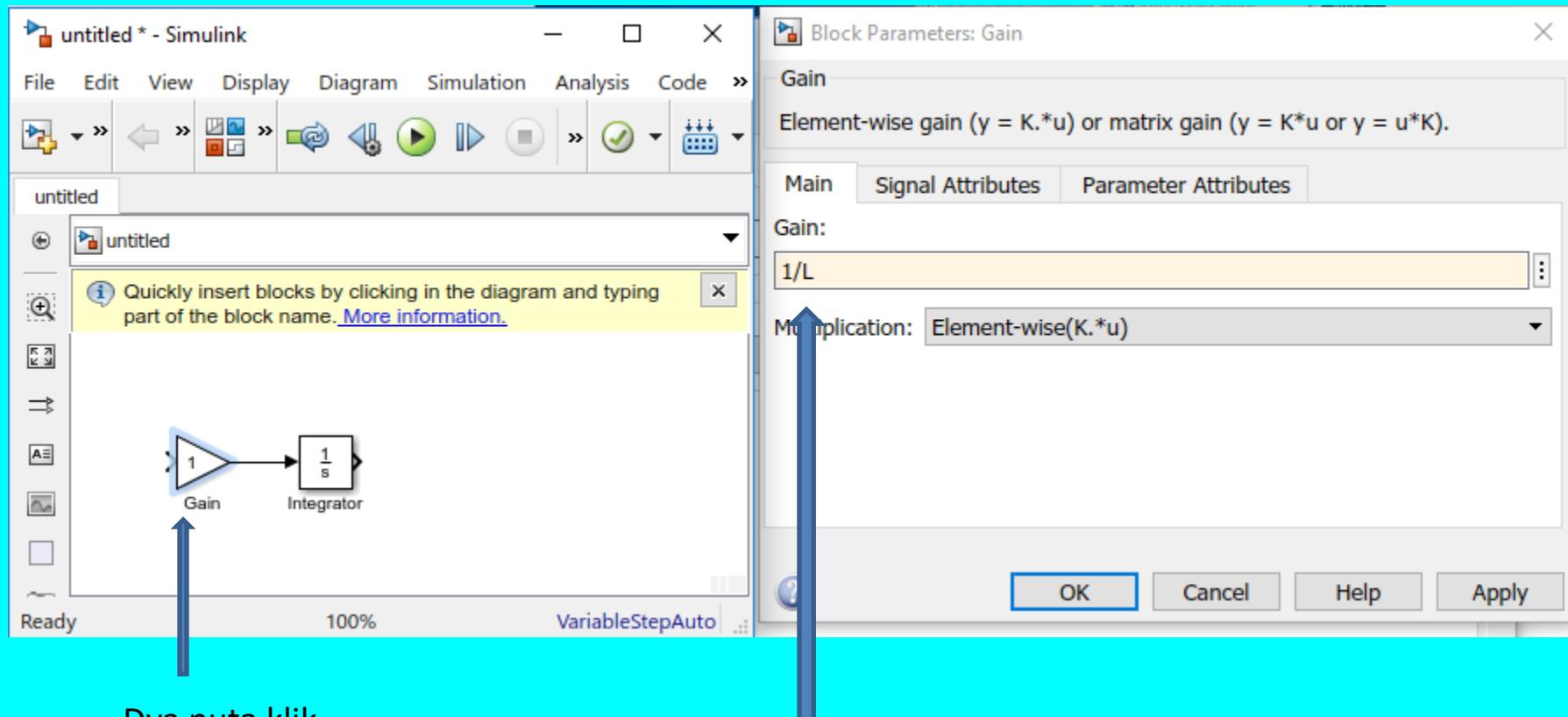
MATLAB Simulink - Primer 1

- Korak 3: Počevši od ulaza integratora konstruišemo ono što nam treba, stoga ovde moramo stvoriti faktor $\frac{1}{L}(u(t) - i(t)R)$ koji je ekvivalentan $di(t)$. Prvo definишемо pojačanje $1/L$.



MATLAB Simulink - Primer 1

- Za postavljanje vrednosti bloka pojačanja dvaput kliknite na njega i zatim promenite nihovu vrednost:

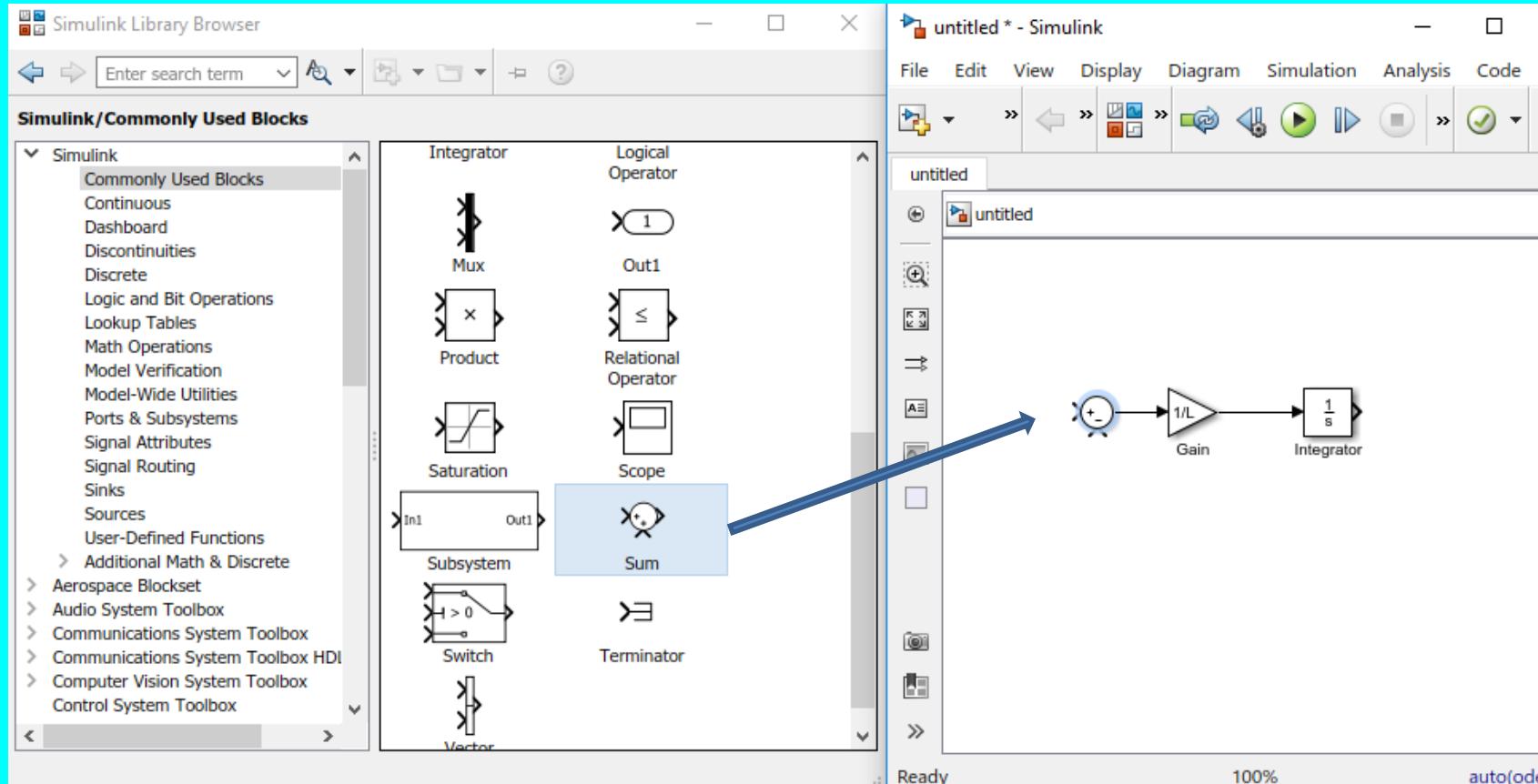


Dva puta klik

Unosimo vrednost pojačanja $1/L$ i overimo tasterom OK

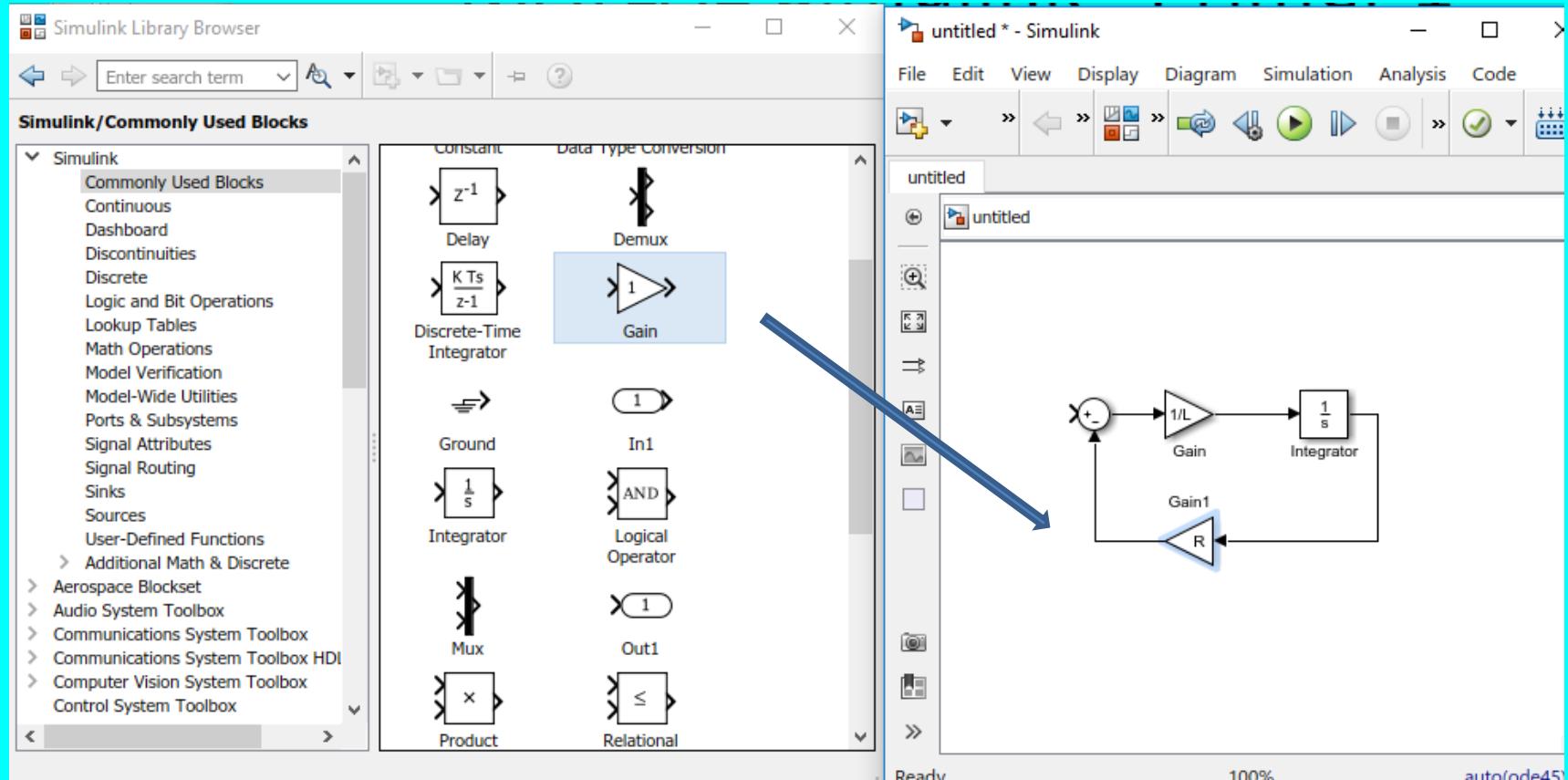
MATLAB Simulink - Primer 1

Korak 4: Sada se mora konstruisati pojam $[u(t)-I(t)R]$, trebaće nam tačka zbrajanja i još jedno pojačanje:



Dva puta kliknuti na sabirač i postaviti njegove parametre na: L+- . Zatim ga povezati sa Gain.

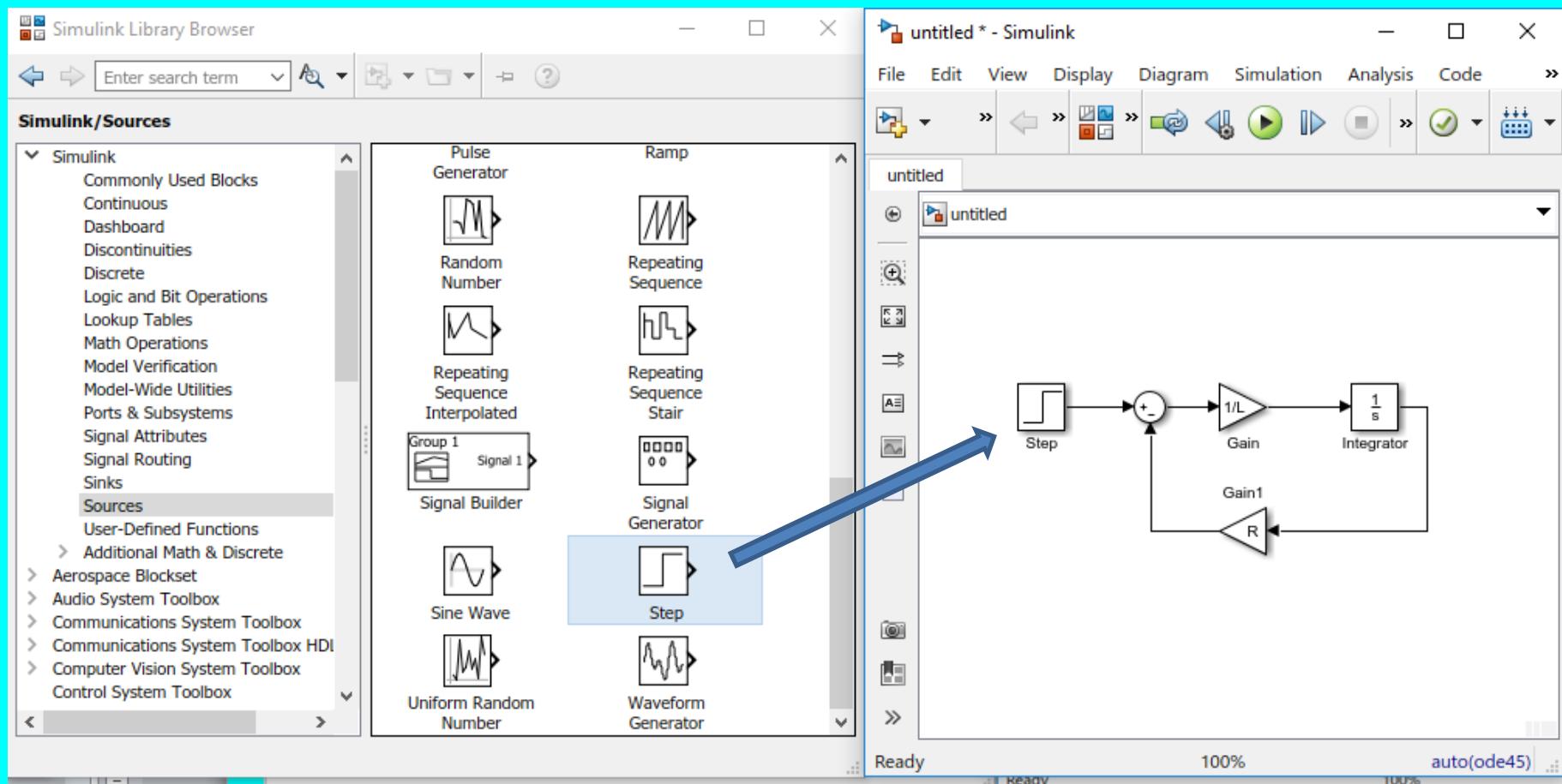
MATLAB Simulink - Primer 1



Prevucimo u naš radni prostor i alat za pojačanje, označimo ga i iz menija **Diagram** iz padajućeg prozora **Rotate and Flip** zakrenuti blok nalevo. Zatim ga povezati kao na slici, dva puta kliknuti na njega i postaviti parametar R.

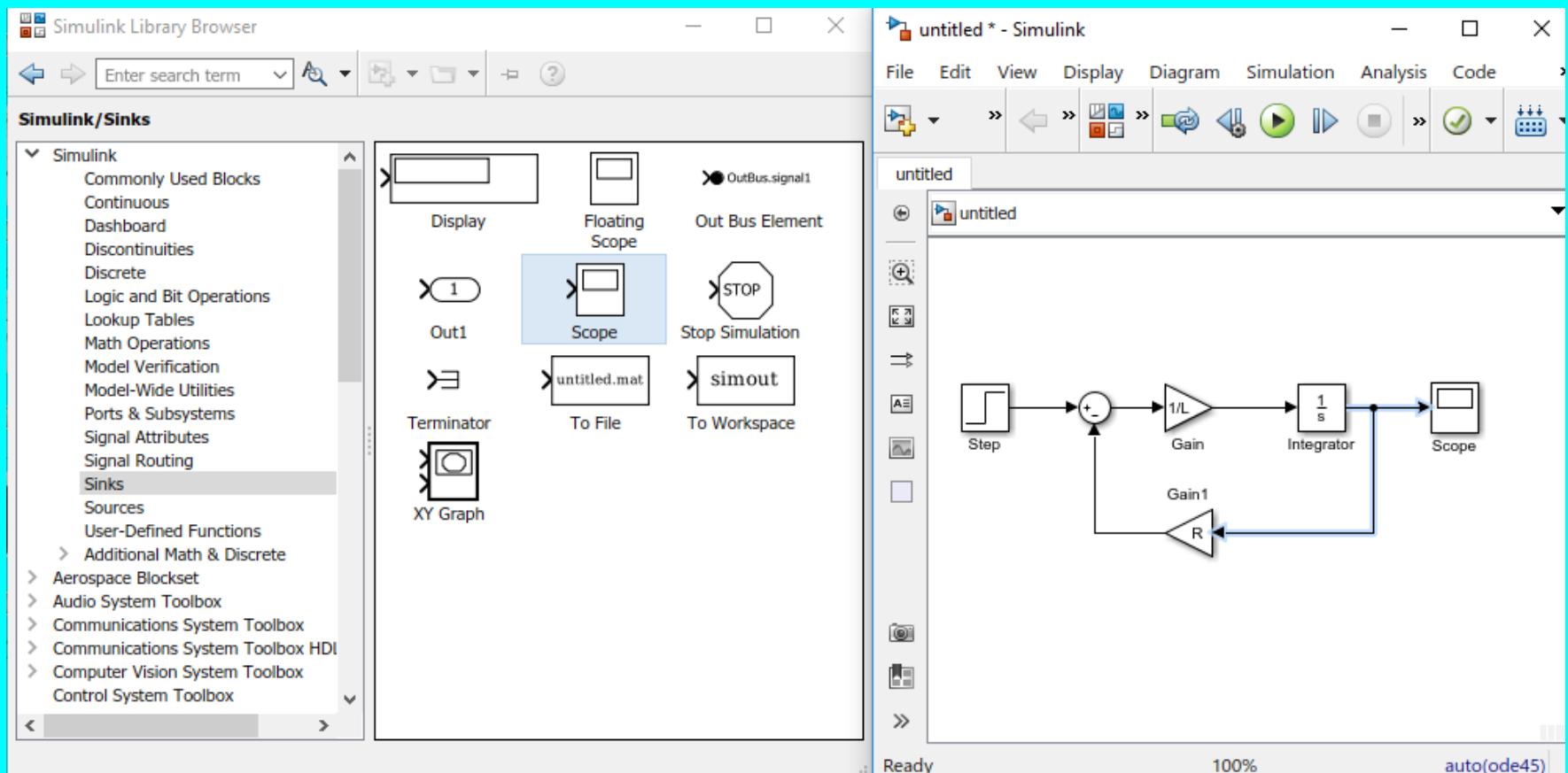
MATLAB Simulink - Primer 1

Korak 5a: Sada moramo dodati ulazni signal za simulaciju promene napona i nešto da vidimo reakciju struje. Za promenu napona odlučili smo koristiti koračnii ulaz amplitude 1, prenesemo ga u simulator i povežemo:



MATLAB Simulink - Primer 1

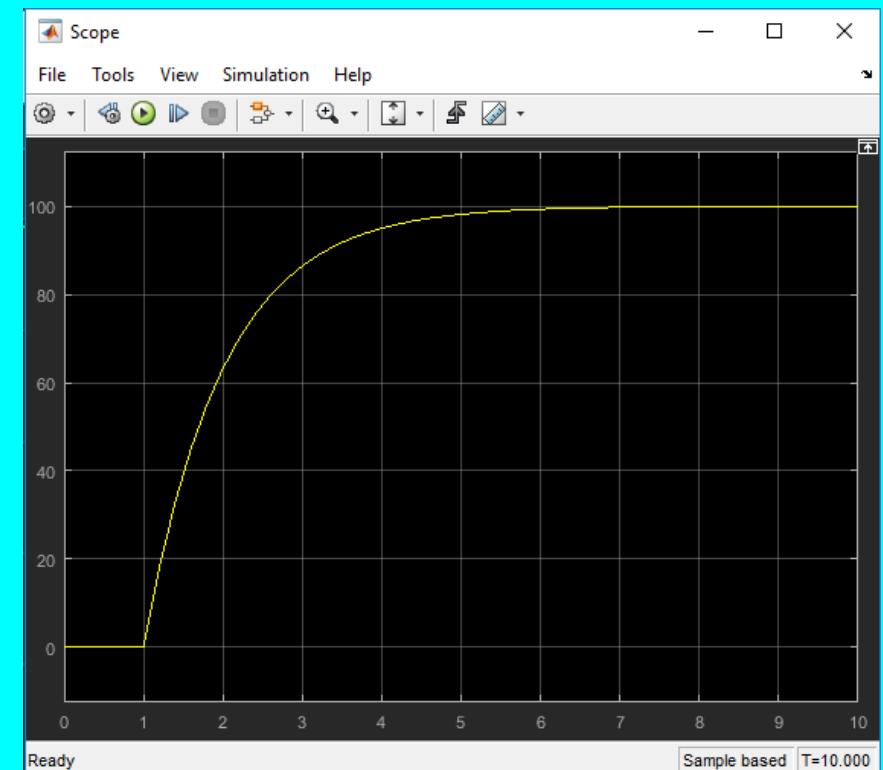
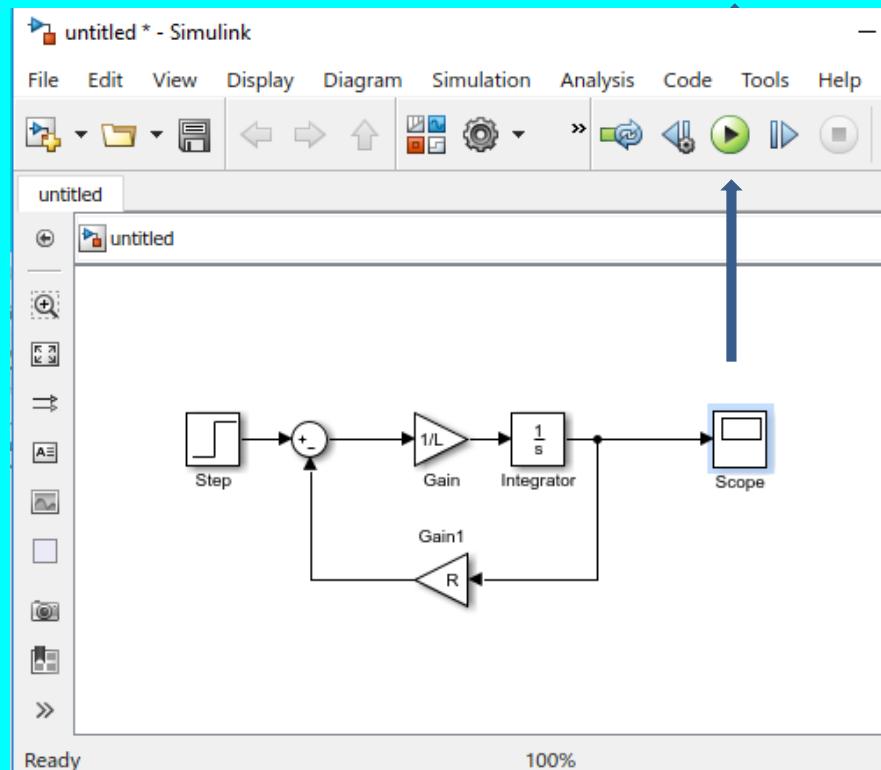
Korak 5b: Sada moramo dodati ulazni signal za simulaciju promene napona i nešto da vidimo reakciju struje. Za promenu napona odlučili smo koristiti koračnii ulaz amplitude 1, a za izlaz možemo koristiti osciloskop:



Proverićemo parametre Gain, Gain 1 i Integrator na $1/L ; R ; 1/s$

MATLAB Simulink - Primer 1

Sada u komandnom prozoru Matlab_a moramo uneti vrednosti za L i R (`>>R=0.01;` `>>L=0.01;`). Pokrenuti ekranski taster **Run**. Dva puta kliknuti mišem na osciloskop da bi videli rezultat:



The Command Window shows the following MATLAB commands:

```
>> simulink
>> simulink
>> R=0.01;
>> L=0.01;
```

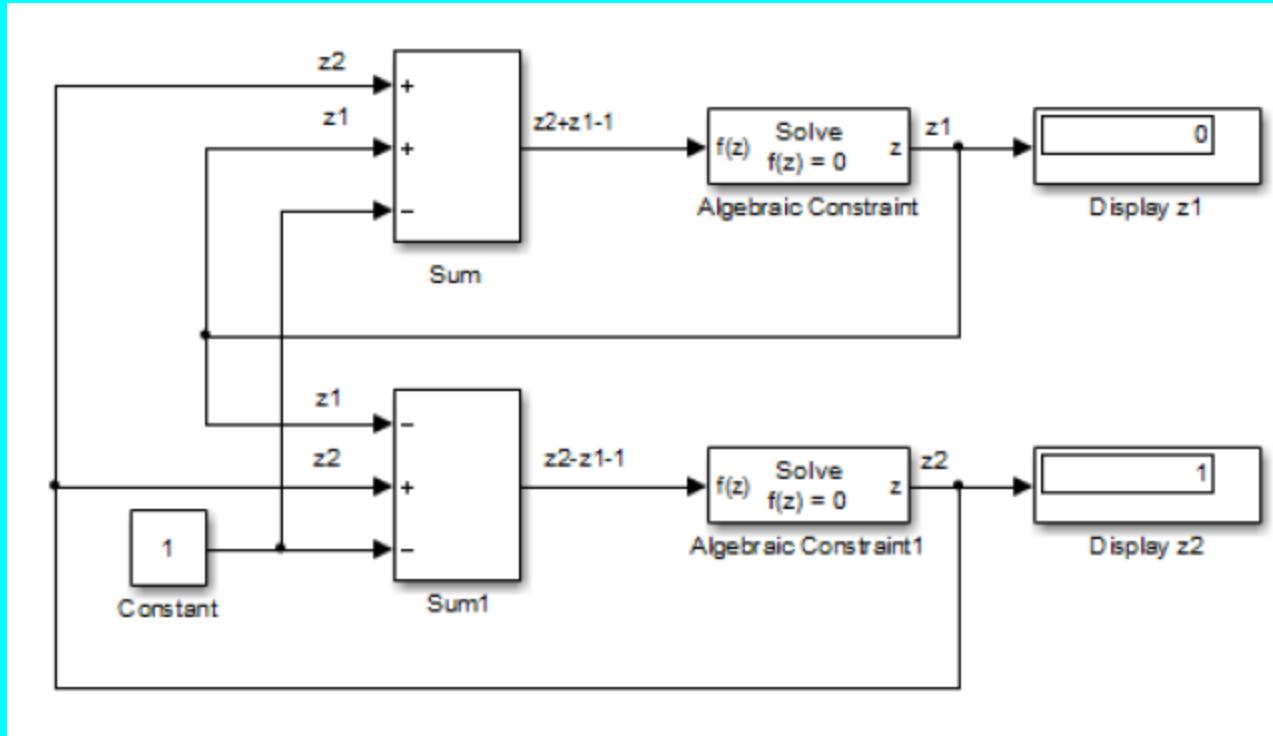
MATLAB Simulink - Primer 2

Na primer, sledeći simulink model rešava ovaj sistem jednačina:

$$z_2 + z_1 = 1$$

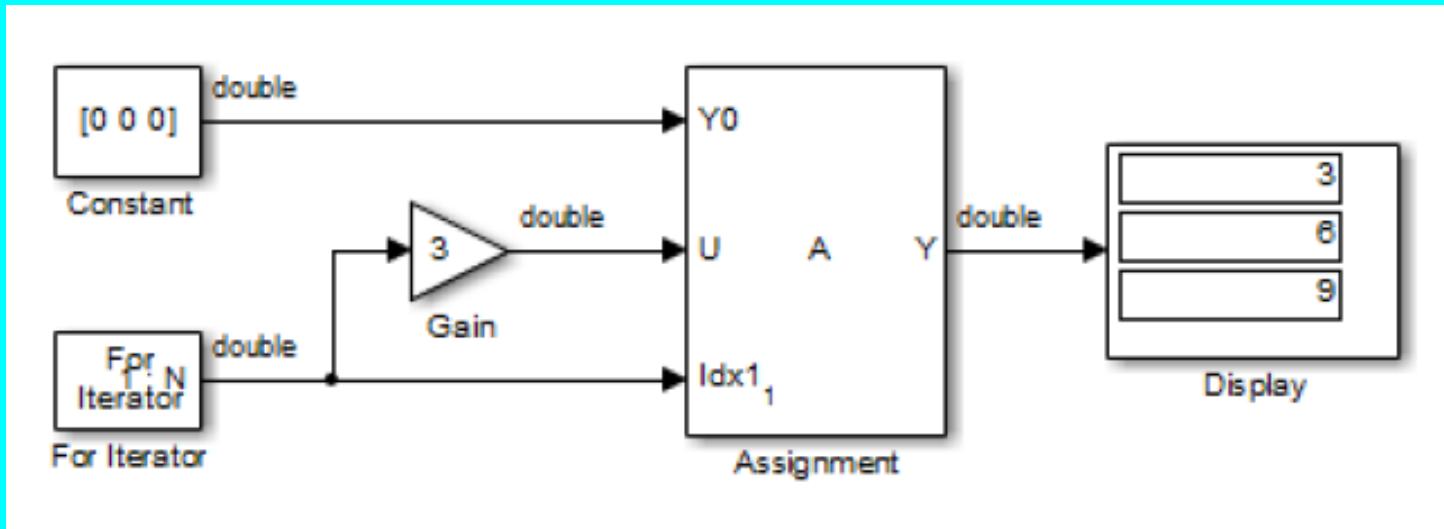
$$z_2 - z_1 = 1$$

Rešenja su $z_2 = 1$, $z_1 = 0$, kao što je prikazano na displeju.



MATLAB Simulink - Primer 3

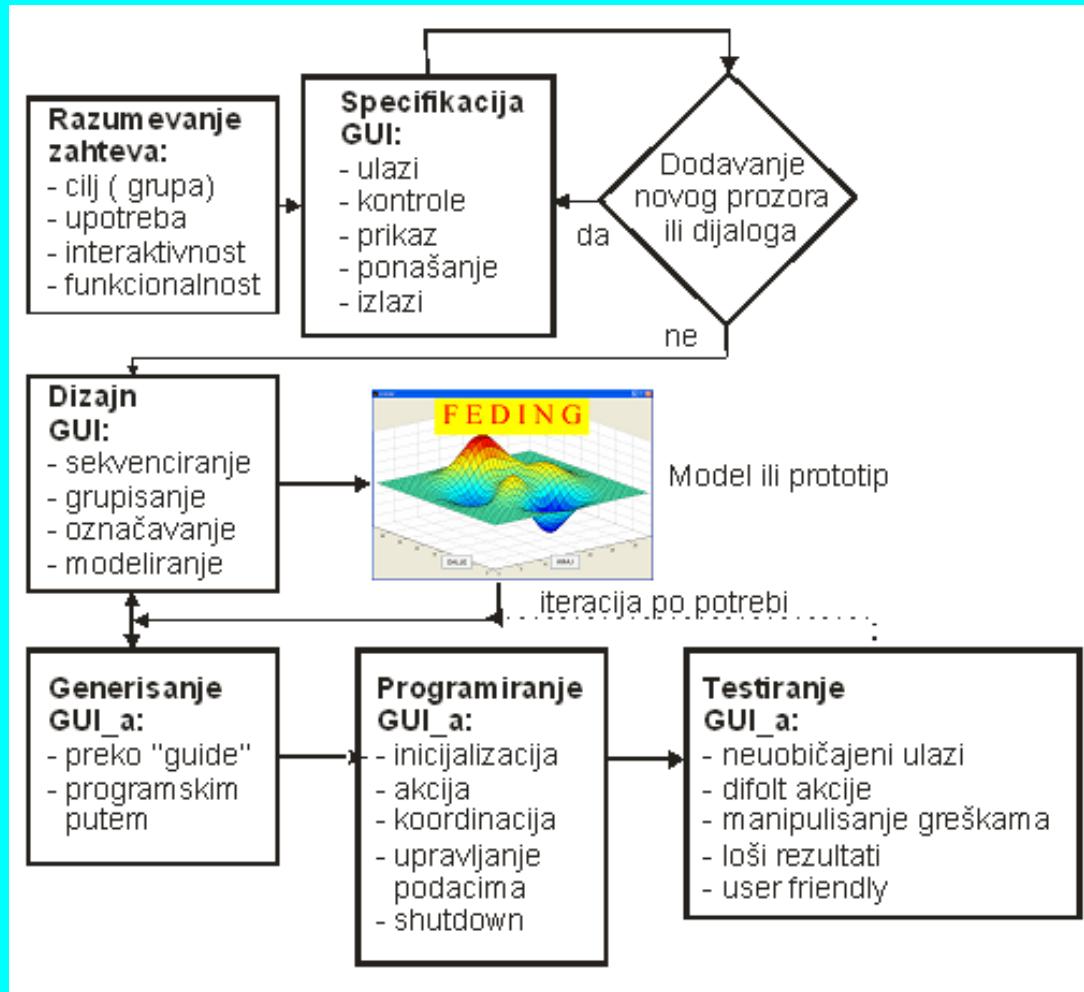
- Blok Assignment možemo koristiti za dodelu vrednosti izračunatih u iterativnoj petlji For ili While nad uzastopnim elementima vektora, matrice ili višedimenzionalnog signala u jednom vremenskom koraku. Na primer, sledeći model koristi blok For Iteratorsa za stvaranje vektora čiji je svaki element jednak 3^i gde je i indeks elementa. Znači, svaki element reda pomnožen je sa 3.



Matlab - Graphical User Interface -GUI

- Kada se bilo koji softver dizajnira, neophodno je razumeti ciljeve koje novi GUI mora da zadovolji i ispuni. Dizajner, ili potencijalni korisnik GUI_a, mora dokumentovati korisničke zahteve, precizno i koliko je to moguće kompletno, pre nego što počne da ih izgrađuje.
- Ovo podrazumeva specificiranje ulaza, izlaza, načina prikaza podataka, ponašanje GUI_a i aplikacije koja ga kontroliše i obezbedjenje interakcije.
- Posle dizajniranja GUI_a, moramo programirati svaku njegovu kontrolu da bi korektno i usklađeno mogla da manipuliše sa dodeljenim zadacima.
- Na kraju, moramo testirati prototip ili kompletno završeni GUI, da bi bili sigurni da je njegovo ponašanje upravo onako kako to zahtevaju zadati realistični uslovi. Ako test ne daje odgovarajuće rezultate i jave se greške u dizajnu ili programskom toku, dizajn se mora iterativno modifikovati sve dok se ne postigne potpuna satisfakcija njegovog rada i ponašanja. Sledeći graf manifestuje i ilustruje glavne aspekte ovog procesa.

Matlab - Graphical User Interface -GUI



Matlab - Graphical User Interface -GUI

MATLAB GUI se predstavlja skupom prozorskih figura kojima se dodaju korisničko-operatorske kontrole. Možemo editovati, selektovati, menjati veličinu i poziciju ovih komponenti onako kako želimo. Korišćenjem funkcija za odaziv kontrola (callback) možemo učiniti da komponenta radi ono što želimo, kada ih korisnik pokrene ili sa njima manipuliše preko ekranskih tastera.

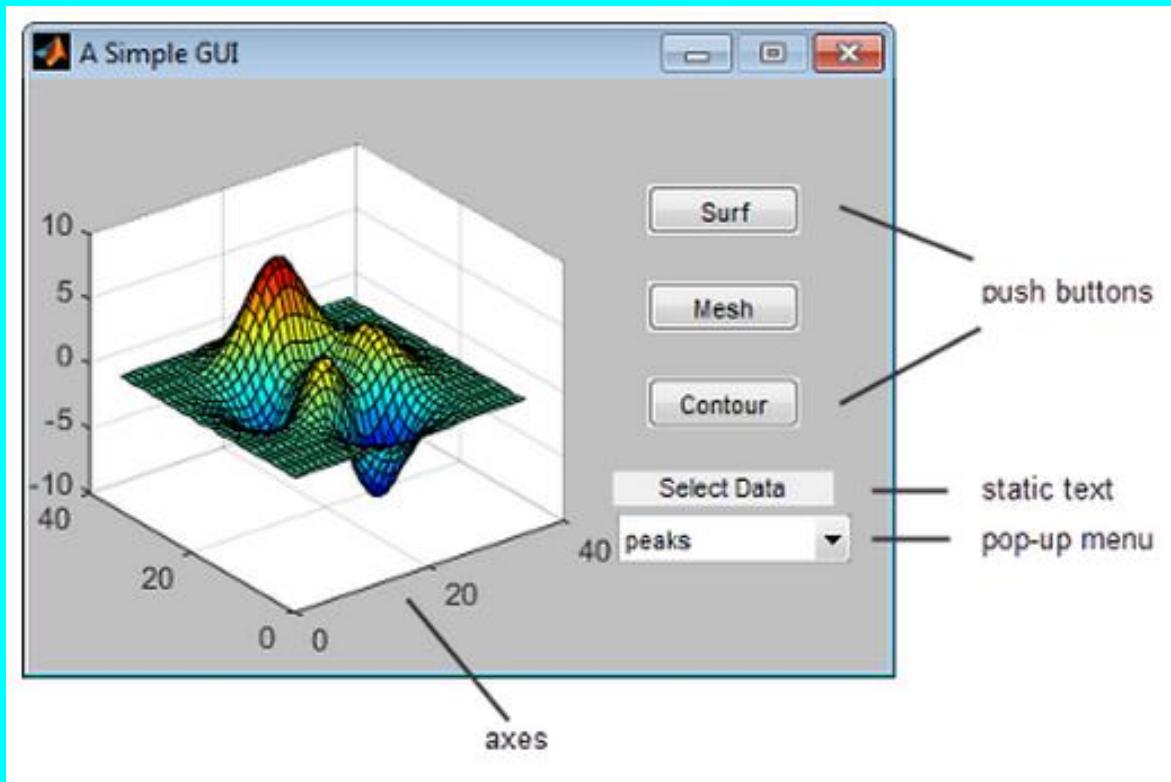
- MATLAB GUI možemo graditi na dva načina:
- Korišćenjem komande guide okruženja za razvoj GUI-a (GUI Development Environment), kao izvanredan interaktivni GUI alat
- Kreiranjem koda za generisanje GUI-a kao funkcije ili skripta (programiranje GUI konstrukcije)

Matlab - Graphical User Interface -GUI

- Prvi pristup startuje sa figurom koju dizajniramo komponentama iz editora za planiranje grafike (graphic layout editor). Tada GUI automatski kreira pridruženi kod koji sadrži sve opise i načine ponašanja komponenti, njihove callback funkcije i smešta ih u MATLAB fajl *.m. Pored toga, na odabranom folderu, MATLAB automatski kreira fajl dizajnirane figure *.fig. Otvaranje ma kojeg ovako organizovanog i zapamćenog fajla, takođe otvara i drugi fajl i izvršava GUI.
- U drugom, programerskom pristupu izgradnje GUI_a, kreiramo kod koji definiše sva ponašanja i osobine ugrađenih komponenti, i kada korisnik izvršava ove fajlove, on kreira figure, dizajnirane skupovima komponenti ili kontrola, i rukuje njihovom interakcijom. Kao rezultat, ovaj skup kodova izgleda drugačije od automatski generisanih kodova u prvom pristupu. Programski GUI fajlovi su generalno duži, zbog toga što eksplicitno definišu svaku osobinu figure i njenu kontrolu, a takođe i čitav skup callback funkcija za svaku kontrolu ponaosob. GUIDE GUI međutim definiše većinu ovih osobina, ali ne i samu figuru. Ove osobine su zapamćene u FIG fajlu, više nego u sopstvenom fajlu izvršnog koda. Ovaj kod sadrži callback funkcije i druge funkcije koje inicijalizuju GUI kada se on otvara.

Kreiranje jednostavne aplikacije korišćenjem guide

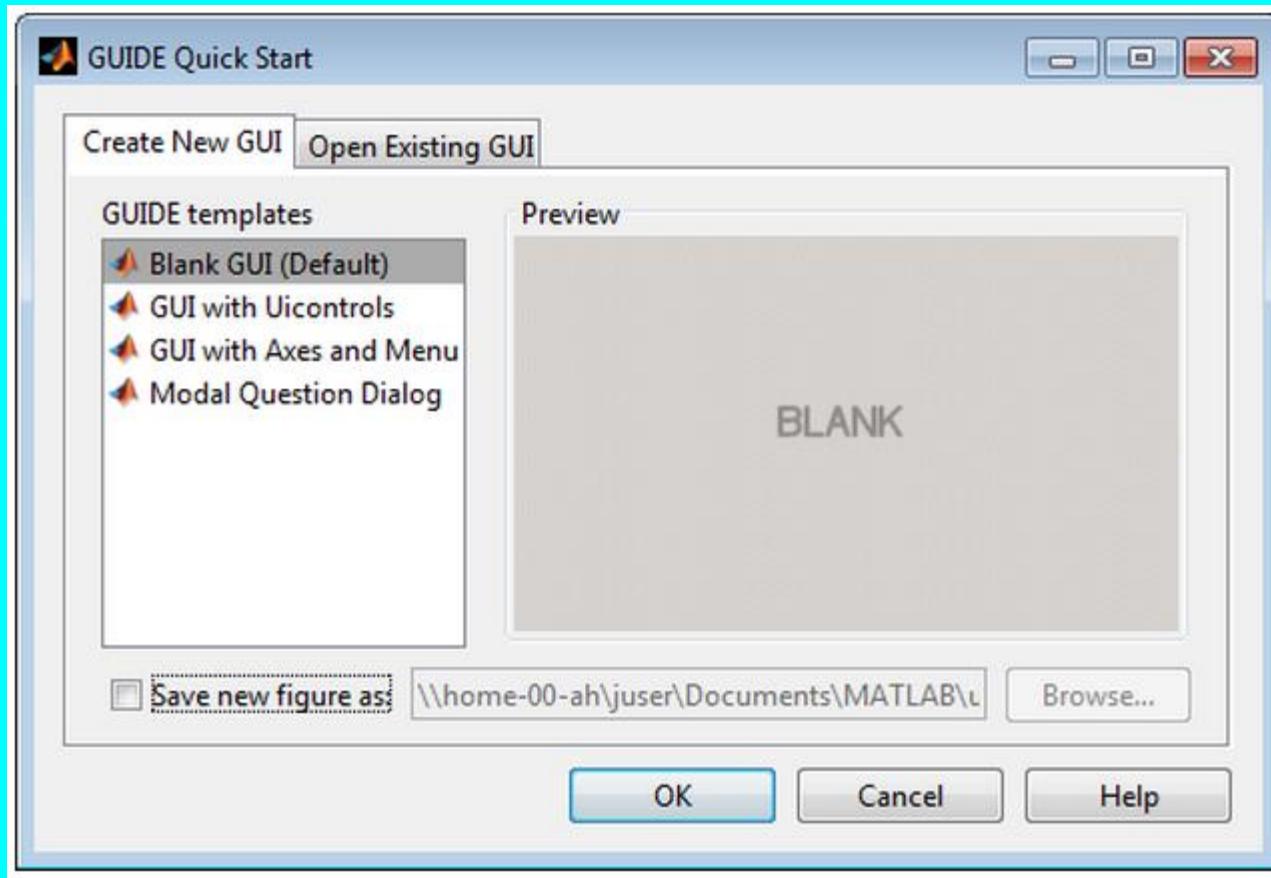
Ovaj primer pokazuje kako koristiti GUIDE da bi kreirali aplikaciju koja ima jednostavan korisnički interfejs (UI), naprimjer, onakav kao na slici.



Kreiranje jednostavne aplikacije korišćenjem guide

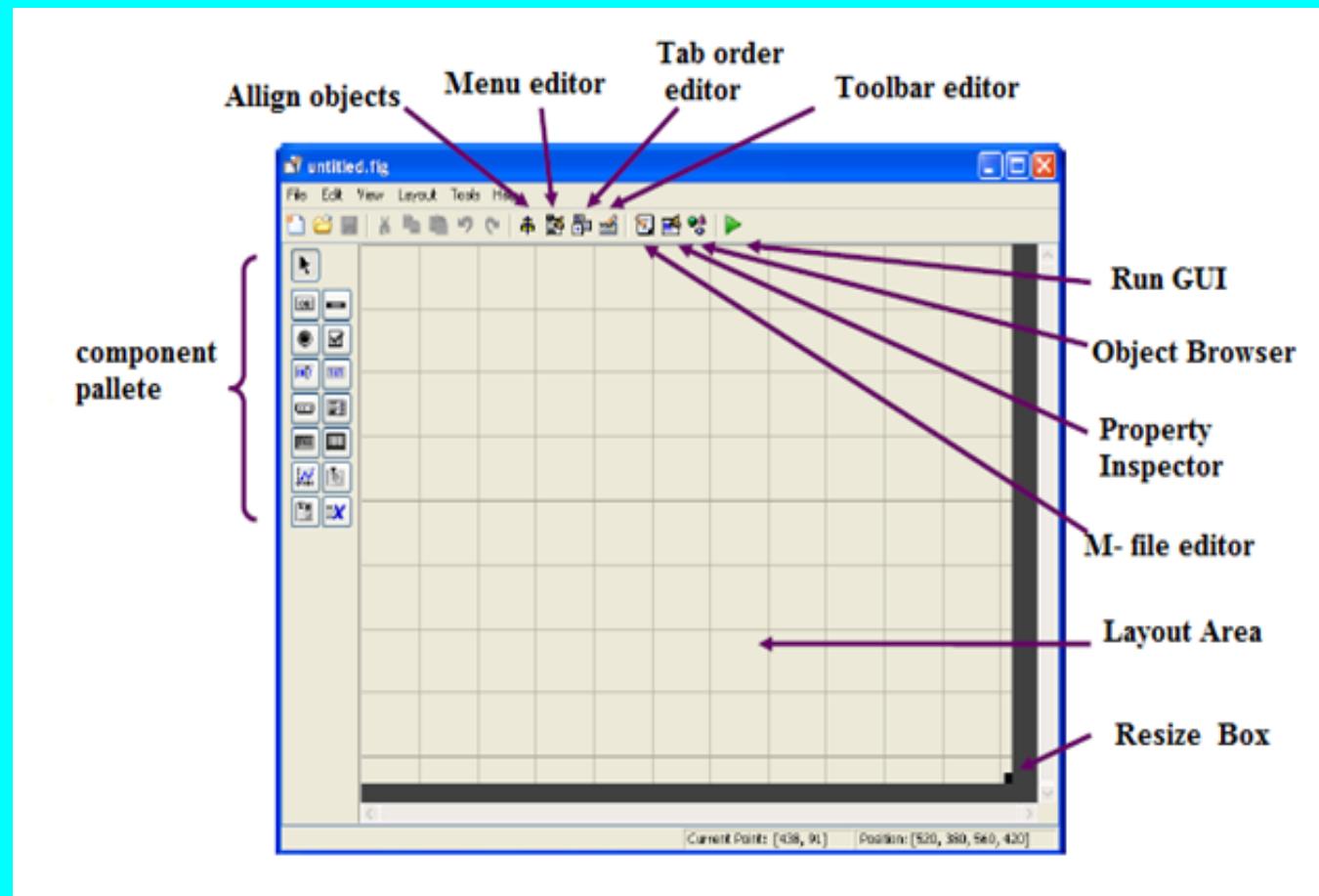
Otvoriti novi User Interface - UI u GUIDE Editoru

1. Startovati GUIDE kucanjem komande **guide** u MATLAB promptu.



Kreiranje jednostavne aplikacije korišćenjem guide

2. U GUIDE Quick Start dialog boksu, selektovati **Blank GUI (Default)** templejt. Otvara se nova figure **untitled.fig** i kliknuti **OK**



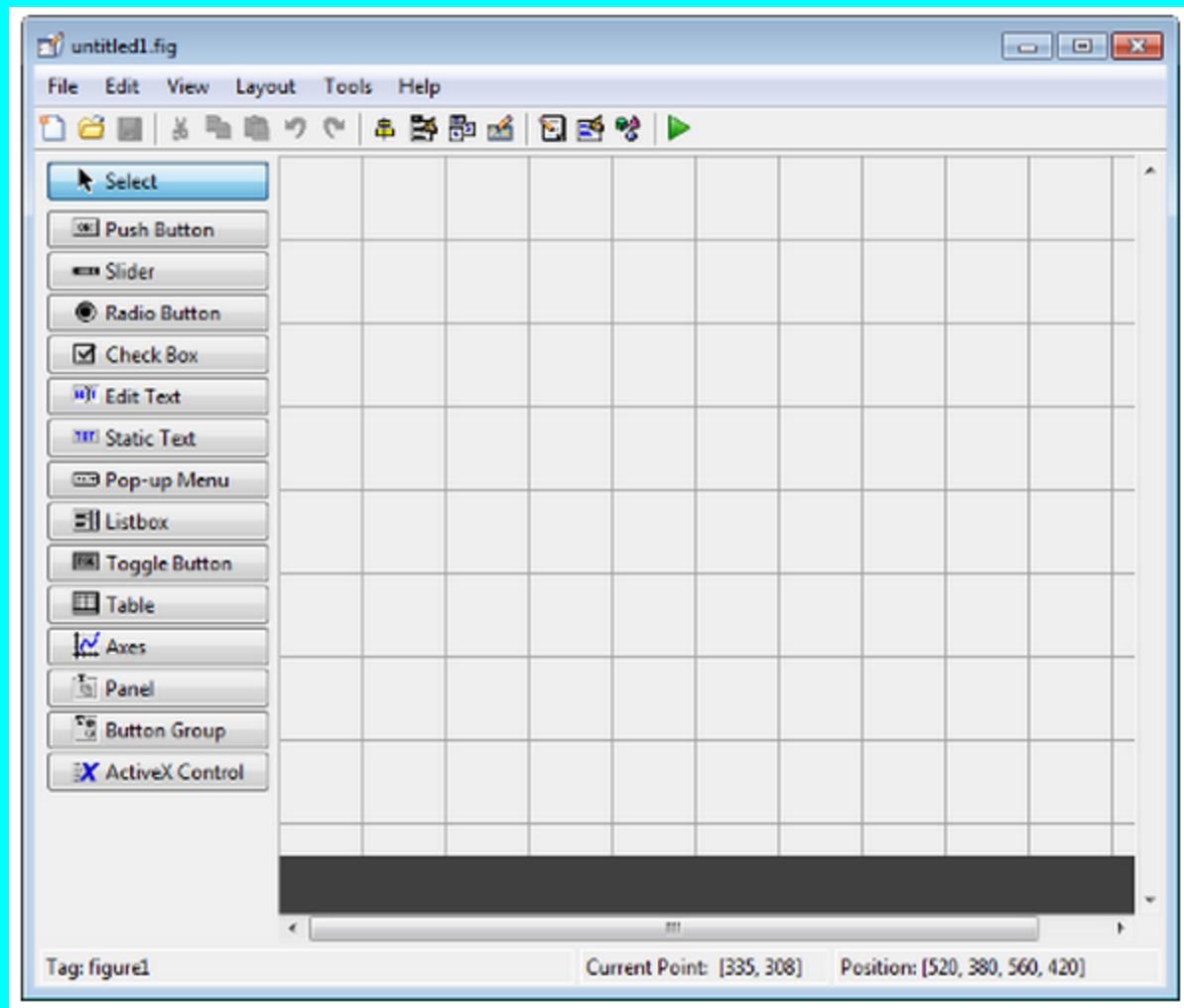
Kreiranje jednostavne aplikacije korišćenjem guide

3. Prikazaće se imena svih komponenti u component palette:

Selekt **File > Preferences > GUIDE.**

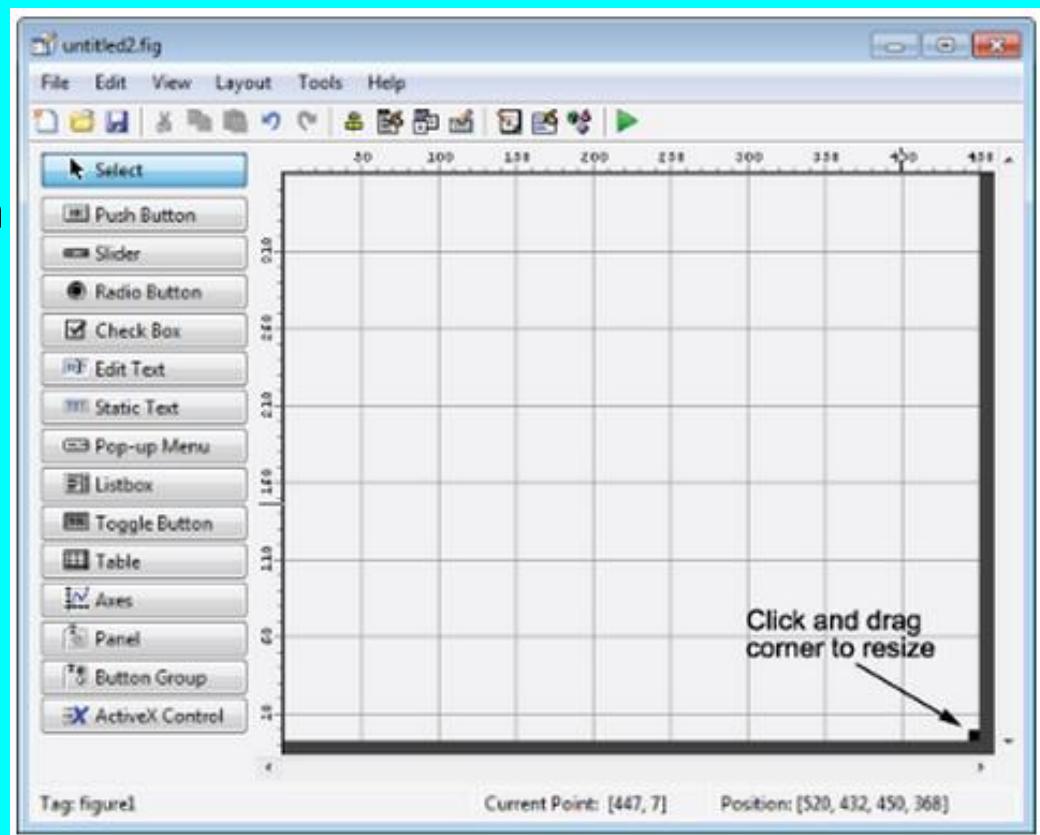
Selekt **Show names in component palette.**

kliknuti **OK.**



Kreiranje jednostavne aplikacije korišćenjem guide

Postavite veličinu prozora promenom veličine područja mreže u izgledu editora. Kliknite na donji desni ugao i povucite ga dok platno ne bude približno 3 inča visoko i 4 inča široko. Ako je potrebno, povećajte platno.



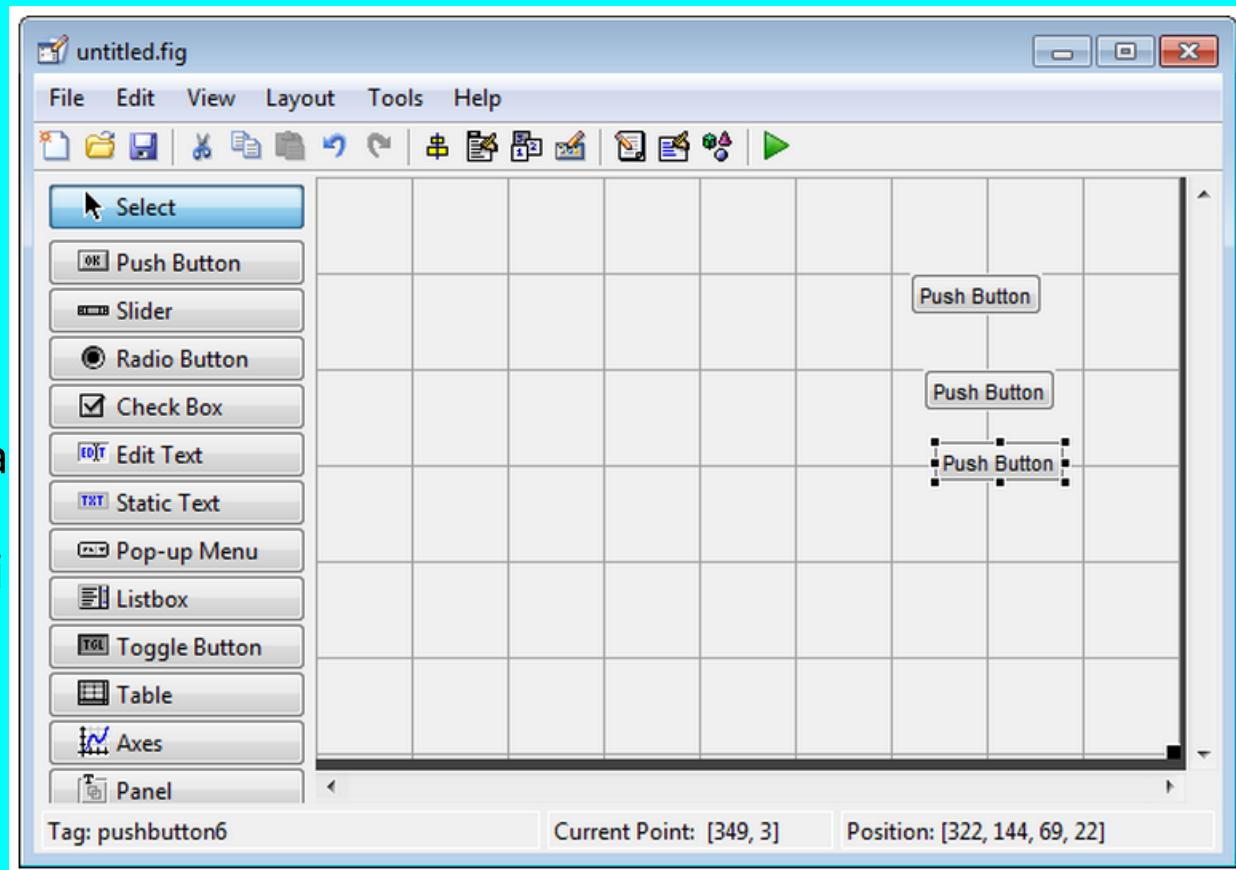
Kreiranje jednostavne aplikacije korišćenjem guide

Izgled korisničkog interfejsa

Dodajte, poravnajte i označite komponente u korisničkom interfejsu.

1. Dodajte tri tastera korisničkom interfejsu.

Odaberite alat sa tasterima iz paleta komponenti na levoj strani Layout Editora i prevucite ga u područje izgleda. Napravite tri tastera, pozicionirajući ih otprilike kao što je prikazano na sedećoj slici.



Kreiranje jednostavne aplikacije korišćenjem guide

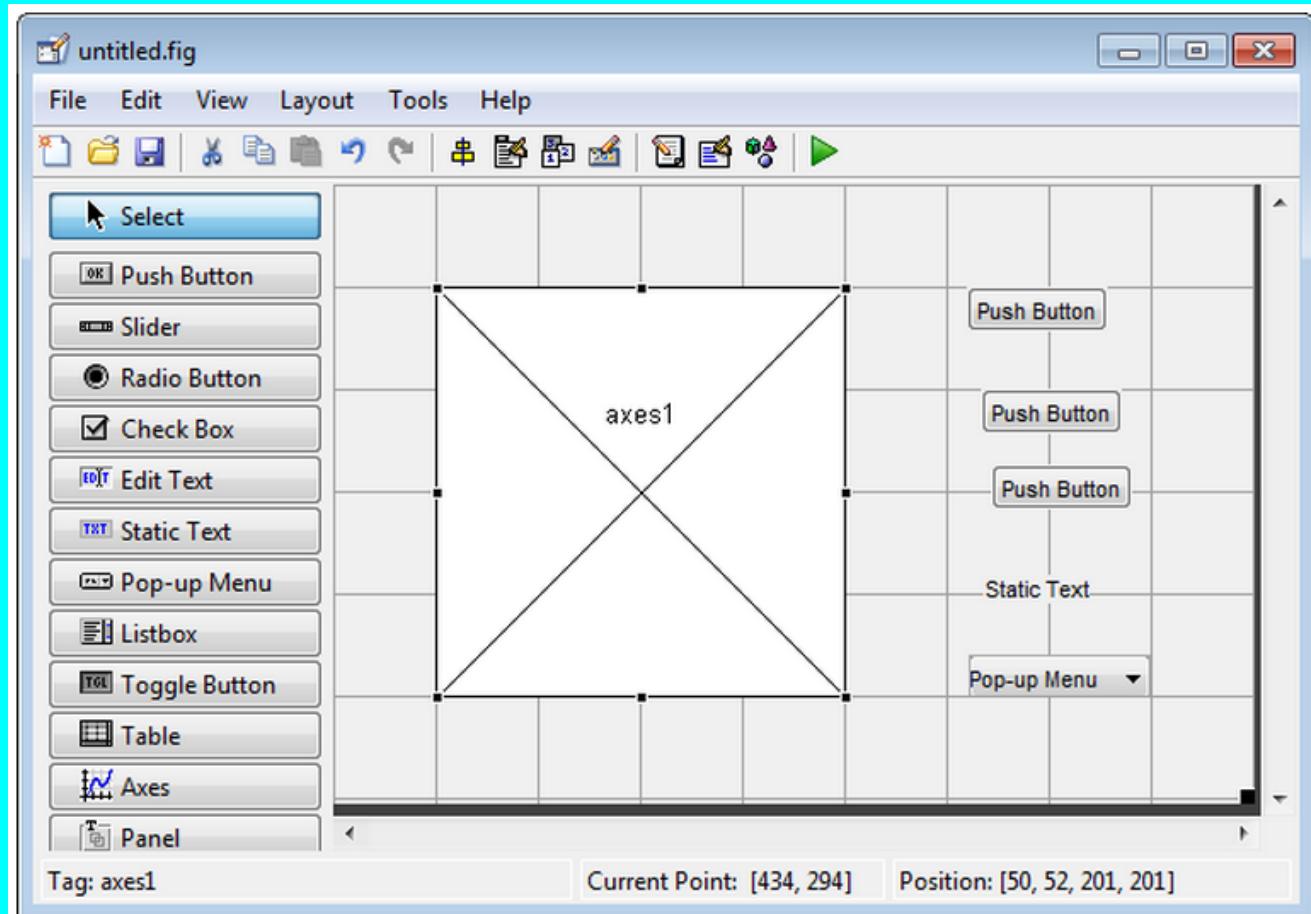
2. Dodaj ostale komponente u IU:

A static text area

A pop-up menu

An axes

Rasporedite komponente kako je prikazano na sledećoj slici. Promenite veličinu komponenti osa na otprilike 2x2 inča.



Kreiranje jednostavne aplikacije korišćenjem guide

Poravnavanje komponenti

Od nekoliko komponenti koje imaju isto poreklo, možete koristiti Alat za poravnavanje da bi ste ih međusobno poravnali. Za poravnavanje tri tastera:

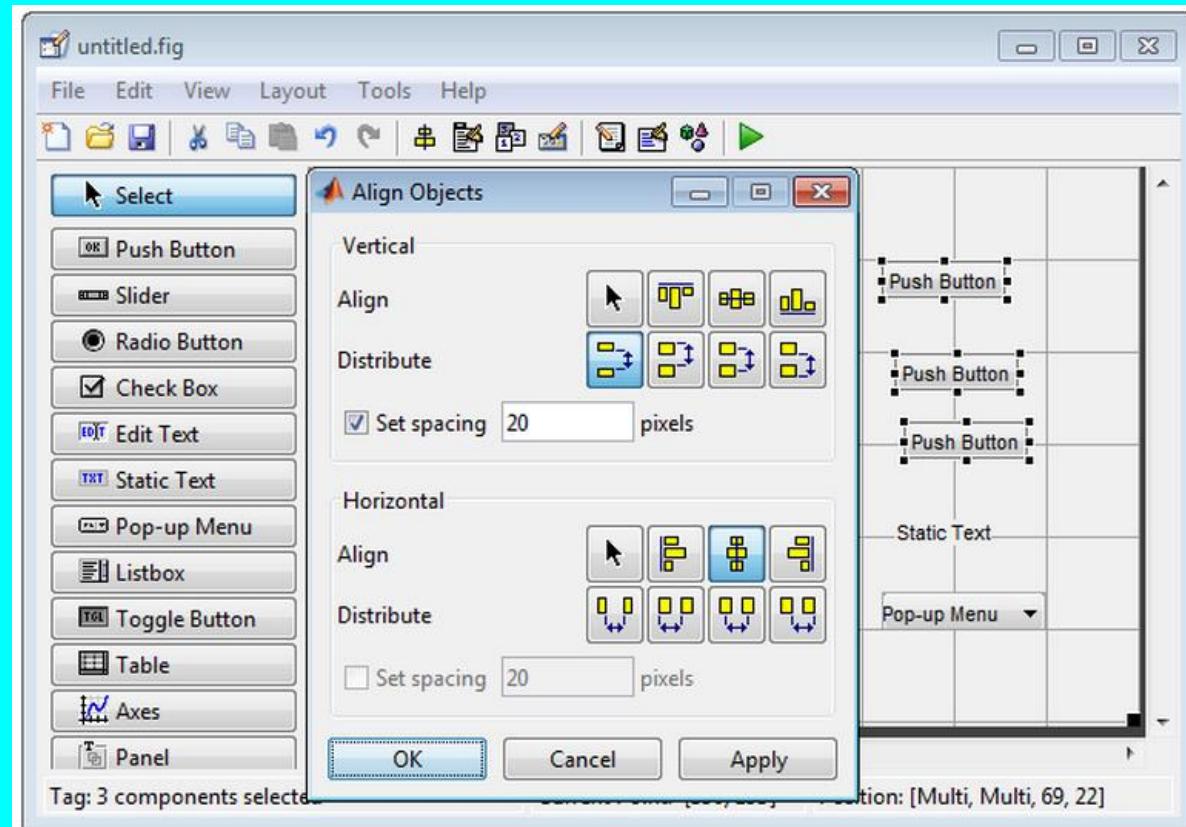
Selektovati sva tri Push Buttons pritiskom na **Ctrl** i kliknuti na njih.

Selekt **Tools > Align Objects**.

3. Napravite ove postavke u Alatu za poravnavanje - Alignment Tool:

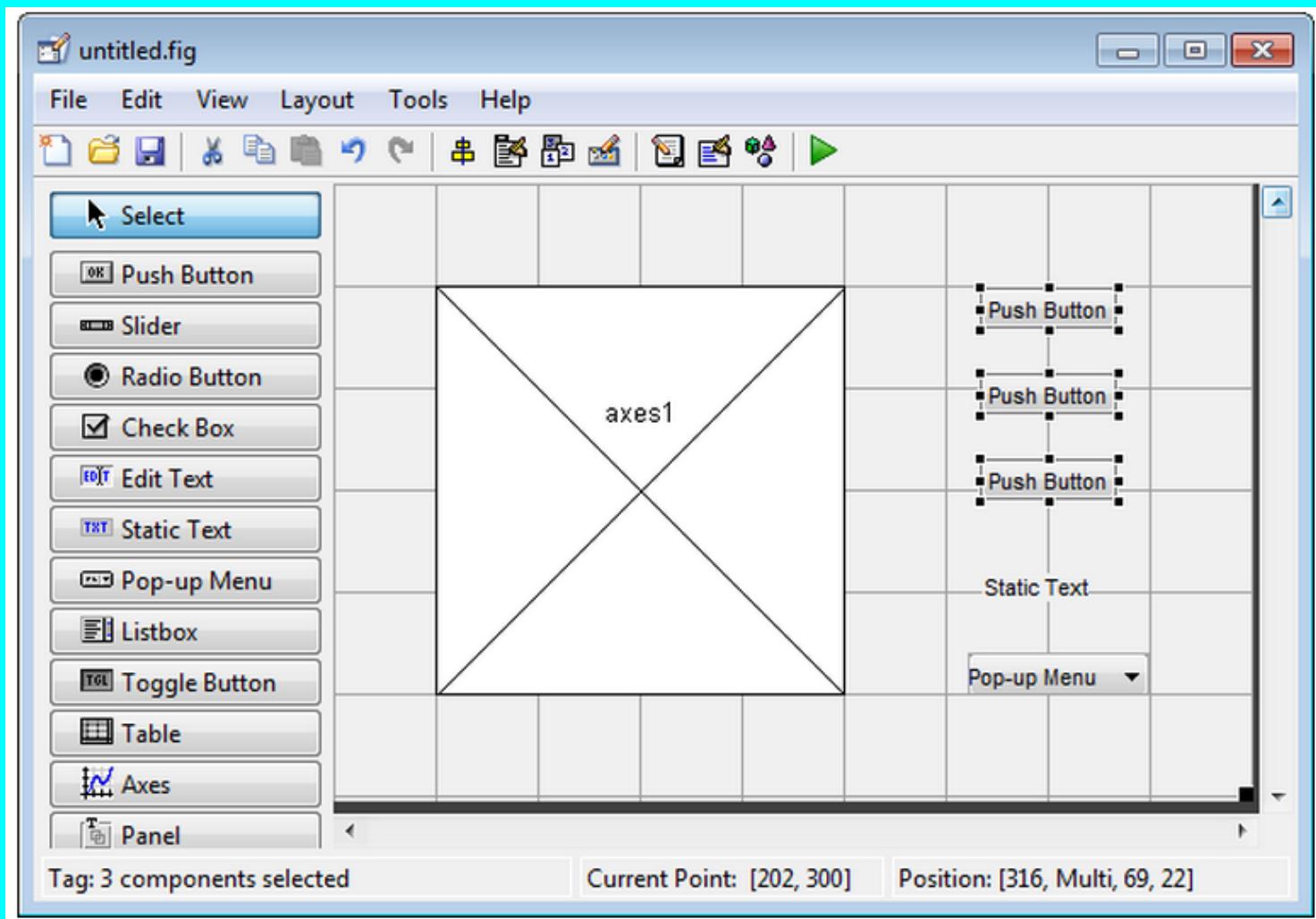
Poravnamo uлево у вodorавном смеру.

Razmak од 20 piksela између push buttoni u uspravnom smeru.



Kreiranje jednostavne aplikacije korišćenjem guide

4. Kliknuti OK



Kreiranje jednostavne aplikacije korišćenjem guide

Označite Push Buttons

Za svako od tri dugmeta odredujemo vrstu grafikona labelama: **surf, mesh i contour**.

Ovaj odeljak pokazuje kako označiti gugmad sa tim opcijama. Prvo Push Button selektovati pa:

1.

Izabrati **View > Property Inspector**, dva puta kliknuti na dugme pored String i u otvorenom prozoru upisati labelu surf. Tako uraditi i za ostala dva dugmeta.

2. In the layout area, click the top push



3. In the Property Inspector, select the



4. Press the **Enter** key. The push button



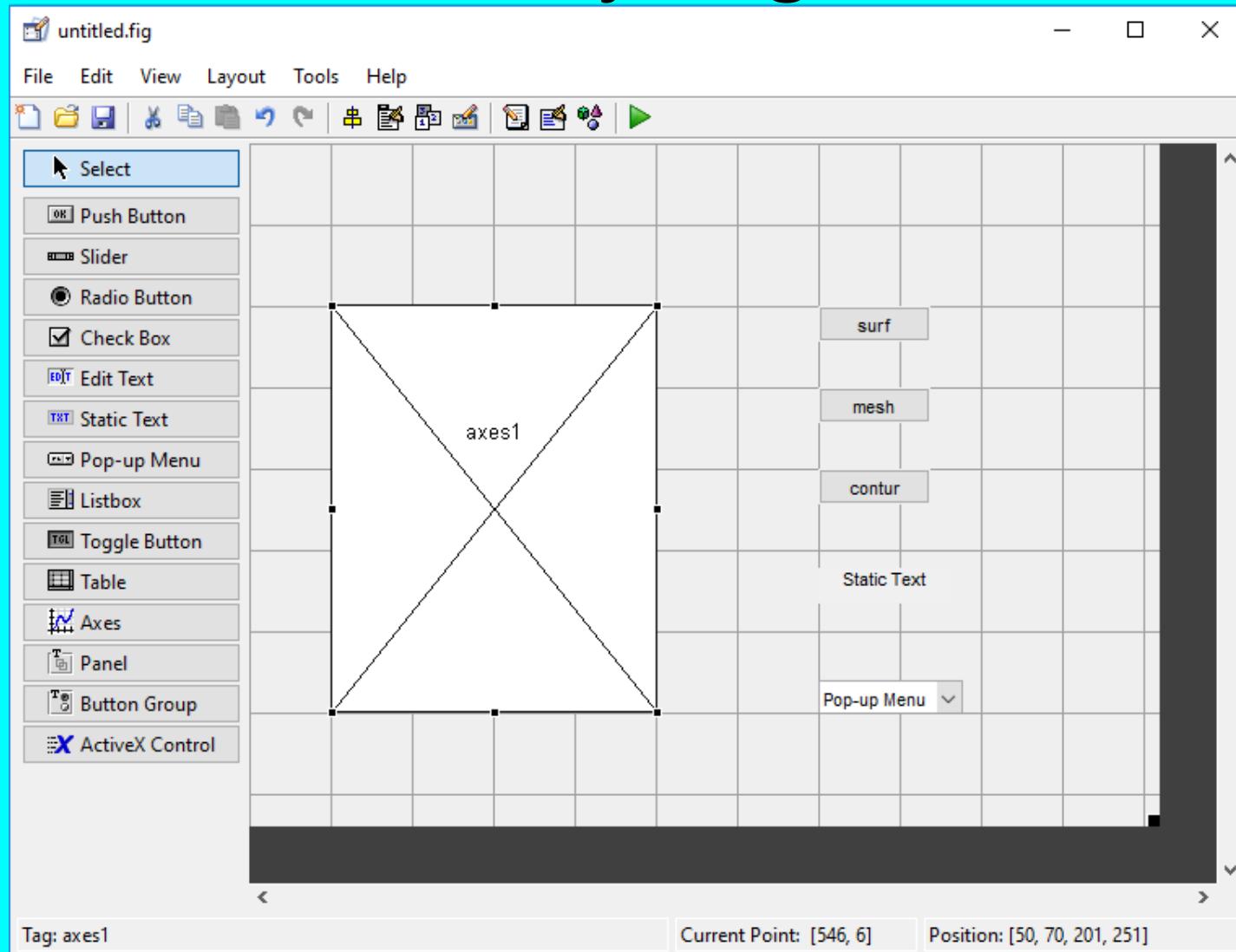
The screenshot shows the MATLAB GUIDE interface with the Property Inspector open. Three separate dialog boxes are displayed, each showing the 'String' property being set:

- The top dialog box shows 'String' set to 'surf'.
- The middle dialog box shows 'String' set to 'mesh'.
- The bottom dialog box shows 'String' set to 'contur'.

The Property Inspector lists various properties for the selected push button, including:

- Callback: %automatic
- CreateFcn: (empty)
- DeleteFcn: (empty)
- Enable: on
- Extent: [0 0 12.4 1.462]
- FontAngle: normal
- FontName: MS Sans Serif
- FontSize: 8.0
- FontUnits: points
- FontWeight: normal
- ForegroundColor: black
- HandleVisibility: on
- HorizontalAlign...: center
- InnerPosition: Multi
- Interruptible: On (checked)
- KeyPressFcn: (empty)
- KeyReleaseFcn: (empty)
- ListboxTop: 1.0
- Max: 1.0
- Min: 0.0
- OuterPosition: Multi
- Position: Multi
- SliderStep: [1x2 double]
- String: (highlighted)
- Style: pushbutton
- Tag: (empty)
- TooltipString: (empty)
- UIContextMenu: <None>
- Units: characters
- UserData: (empty)
- Value: 0.0
- Visible: on (checked)

Kreiranje jednostavne aplikacije korišćenjem guide



Kreiranje jednostavne aplikacije korišćenjem guide

Listanje Pop-Up postavki menija

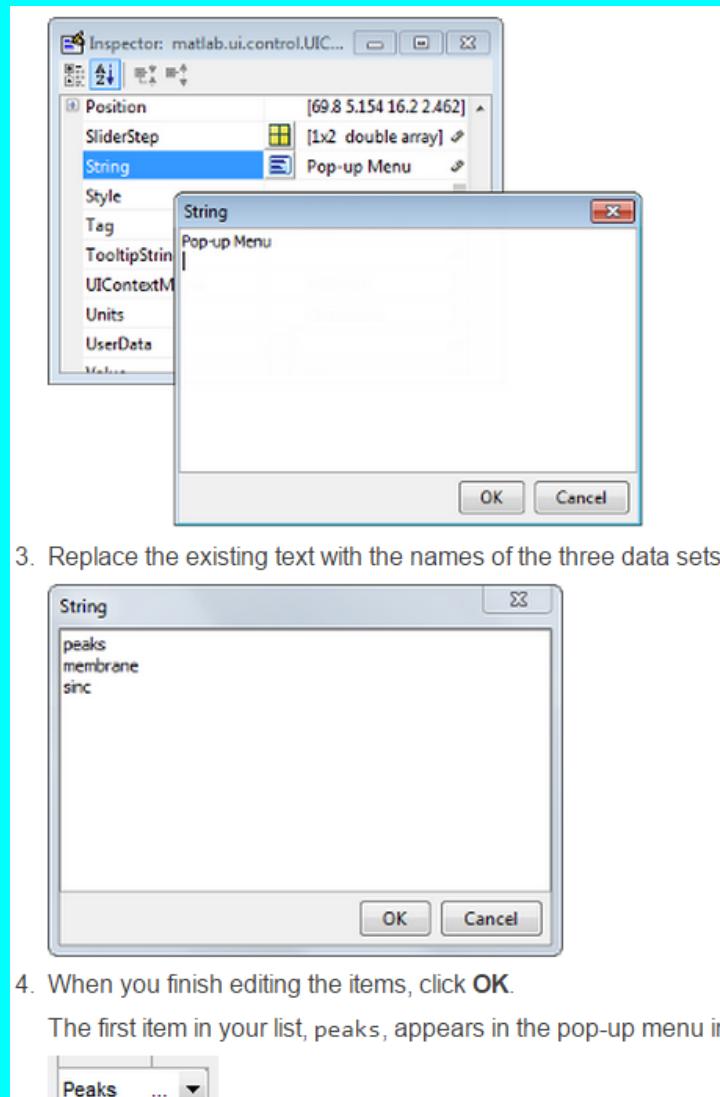
Pop-up meni nudi izbor izmedju tri skupa podataka :

peaks,
membrane, i
sinc.

Ovi skupovi podataka odgovaraju istoimenim MATLAB funkcijama. Ovaj odjeljak pokazuje kako navesti te skupove podataka kao izbore u pop-up meniju.

U području izgleda kliknite **pop-up menu** – **padajući prozor** i selektujemo ga.

U **Property Inspector** (ekranski taster ispod glavnog menija) kliknite dugme pored String. Prikazuje se dijaloški okvir String. U njega unesemo predvidjene funkcije. Isto kao prethodni postupak.



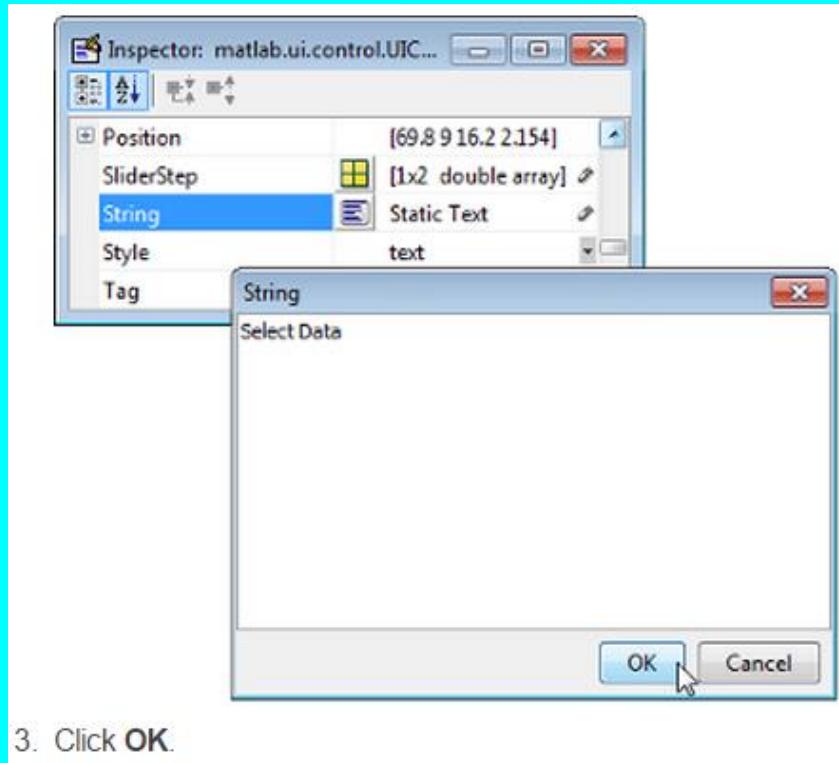
Kreiranje jednostavne aplikacije korišćenjem guide

Izmenite statički tekst

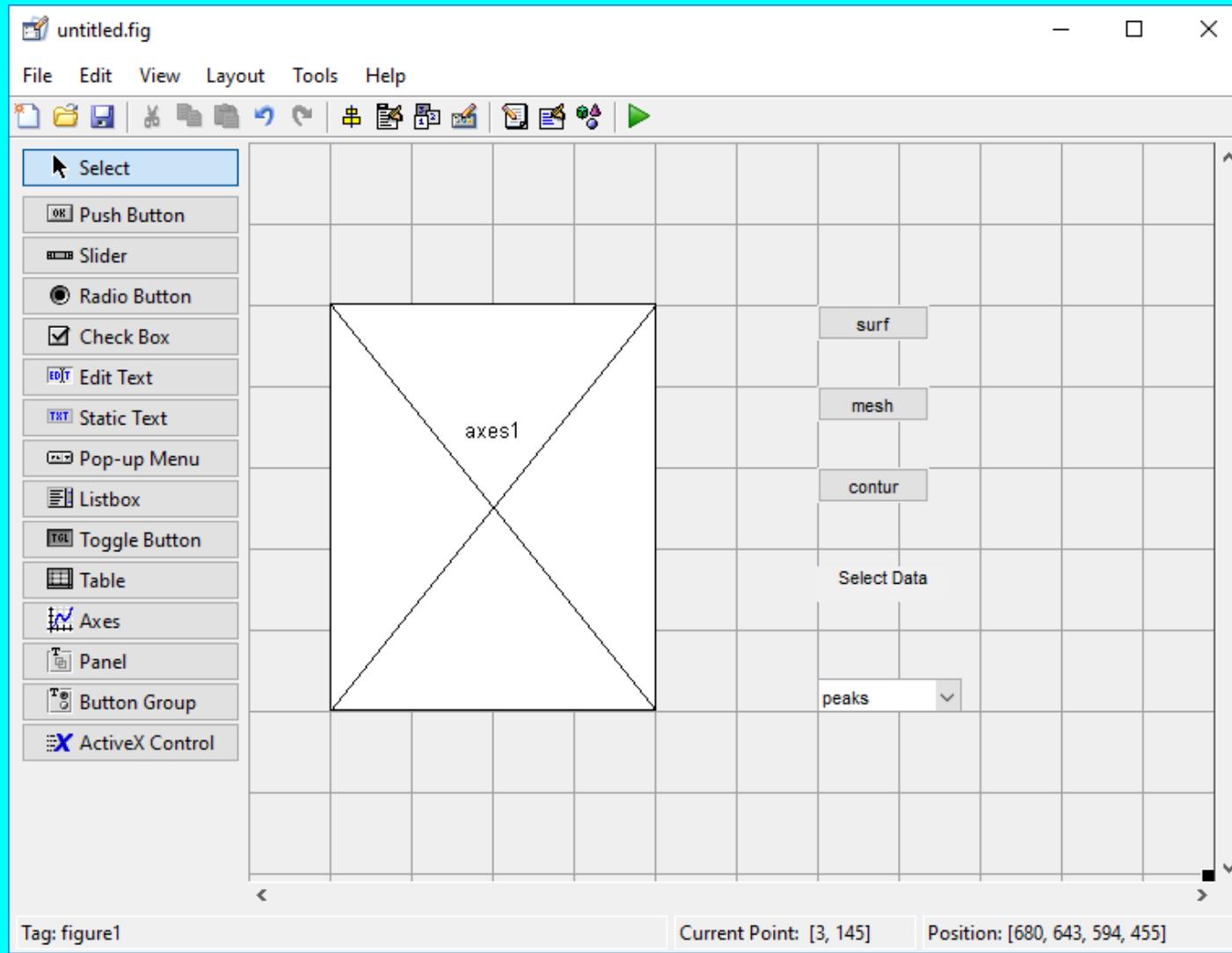
U ovom korisničkom interfejsu, statički tekst služi kao labela za pop-up menu. Ovaj odeljak pokazuje kako promeniti statički tekst da bi označio pop-up menu labelom *Select Data*.

U području izgleda kliknite na static text i označite ga.

U Property Inspector kliknite dugme pored String. U dijaloškom prozoru String koji se prikazuje zamenite postojeći tekst frazom *Select Data*.



Kreiranje jednostavne aplikacije korišćenjem guide



Kreiranje jednostavne aplikacije korišćenjem guide

- **Zapamtite izgled**
- Kada pamtimo izgled - novi korisnički interfejs, GUIDE stvara dve datoteke, FIG-datoteku i datoteku koda. FIG-datoteka, sa nastavkom .fig, binarna je datoteka koja sadrži opis - sliku izgleda. Datoteka koda, sa ekstenzijom .m, sadrži MATLAB funkcije koje kontrolišu ponašanje aplikacije.
- Zapamtite i pokrenite program izborom **Tools > Run**. GUIDE prikazuje dijaloški okvir koji prikazuje: "Aktiviranjem će se zapamtiti promene vaše figure i MATLAB koda. Želite li nastaviti?
- Kliknite Da.
- GUIDE otvara dijaloški prozor Save As u vašem trenutnom direktorijumu i traži od vas naziv FIG datoteke.
- Unesite naziv datoteke **proba1_gui** za FIG-datoteku. GUIDE sprema i FIG-datoteku i kodnu datoteku koristeći ovaj naziv.
- GUIDE pamti datoteke **proba1_gui.fig** i **proba1_gui.m**, a zatim pokreće program. Takodje otvara kodnu datoteku u vašem difoltnom editoru.
- Aplikacija se otvara u novom prozoru. Primetite da prozoru nedostaje standardna traka menija i alatna traka koju prikazuju MATLAB prozori sa slikama. Možete dodati vlastite menije i tastere na alatnoj traci uz GUIDE, ali prema zadatim postavkama aplikacija ova GUIDE ne uključuje nijednu od ovih komponenti.
- Kada pokrenete **proba1_gui**, možete odabratи skup podataka u pop-up meniju i kliknuti dugme, ali ništa se ne događa. To je zato što datoteka koda ne sadrži izjave za servisiranje pop-up menuija i dugmadi. Datoteka koda se piše u pozadini - CASE alat reverznog inžinjerstva, koju automatski vidimo u editoru i možemo je bez problema dopunjavati i izvršiti.

Datoteke proba1_gui.fig i proba1_gui.m



```
Editor - C:\Users\BORA\Desktop\MATLAB proba\proba1_gui.m
proba1_gui.m  X  +
1 function varargout = proba1_gui(varargin)
2 % PROBA1_GUI MATLAB code for proba1_gui.fig
3 %   PROBA1_GUI, by itself, creates a new PROBA1_GUI or raises the
4 %   singleton*.
5 %
6 %   H = PROBA1_GUI returns the handle to a new PROBA1_GUI or the
7 %   existing singleton*.
8 %
9 %   PROBA1_GUI('CALLBACK', hObject, eventData, handles,...) calls
10 %   function named CALLBACK in PROBA1_GUI.M with the given input
11 %   arguments.
12 %
13 %   PROBA1_GUI('Property','Value',...) creates a new PROBA1_GUI
14 %   existing singleton*. Starting from the left, property value
15 %   applied to the GUI before proba1_gui_OpeningFcn gets called
16 %   unrecognized property name or invalid value makes property
17 %   stop. All inputs are passed to proba1_gui_OpeningFcn via varargin.
18 %
19 %   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
20 %   instance to run (singleton)".
21 %
22 % See also: GUIDE, GUIDATA, GUIHANDLES
23 %
24 % Edit the above text to modify the response to help proba1_gui
25 %
26 % Last Modified by GUIDE v2.5 22-Nov-2021 19:38:26
27 %
28 % Begin initialization code - DO NOT EDIT
29 gui_Singleton = 1;
30 gui_State = struct('gui_Name',         mfilename, ...
31                   'gui_Singleton',    gui_Singleton, ...
32                   'gui_OpeningFcn',   @proba1_gui_OpeningFcn, ...
33                   'gui_OutputFcn',    @proba1_gui_OutputFcn, ...
34                   'gui_LayoutFcn',    @proba1_gui_LayOutFcn, ...
35                   'gui_Callback',     []);
36
37 % Explicitly set guidata(hObject,gui_State);
38
39 % End initialization code
```

CASE struktura

Kreiranje jednostavne aplikacije korišćenjem guide

Kodirajte ponašanje aplikacije

- Kada ste spremili izgled u prethodnom odeljku, Save the Layout, GUIDE je stvorio dve datoteke: FIG-datoteku, proba1_gui.fig, i programsku datoteku proba1_gui.m. Međutim, aplikacija ne reaguje jer proba1_gui.m ne sadrži nikakve izjave koje izvode radnje. Ovaj odeljak pokazuje kako dodati kod u datoteku da bi aplikacija postala funkcionalna.

Generišite podatke za crtanje

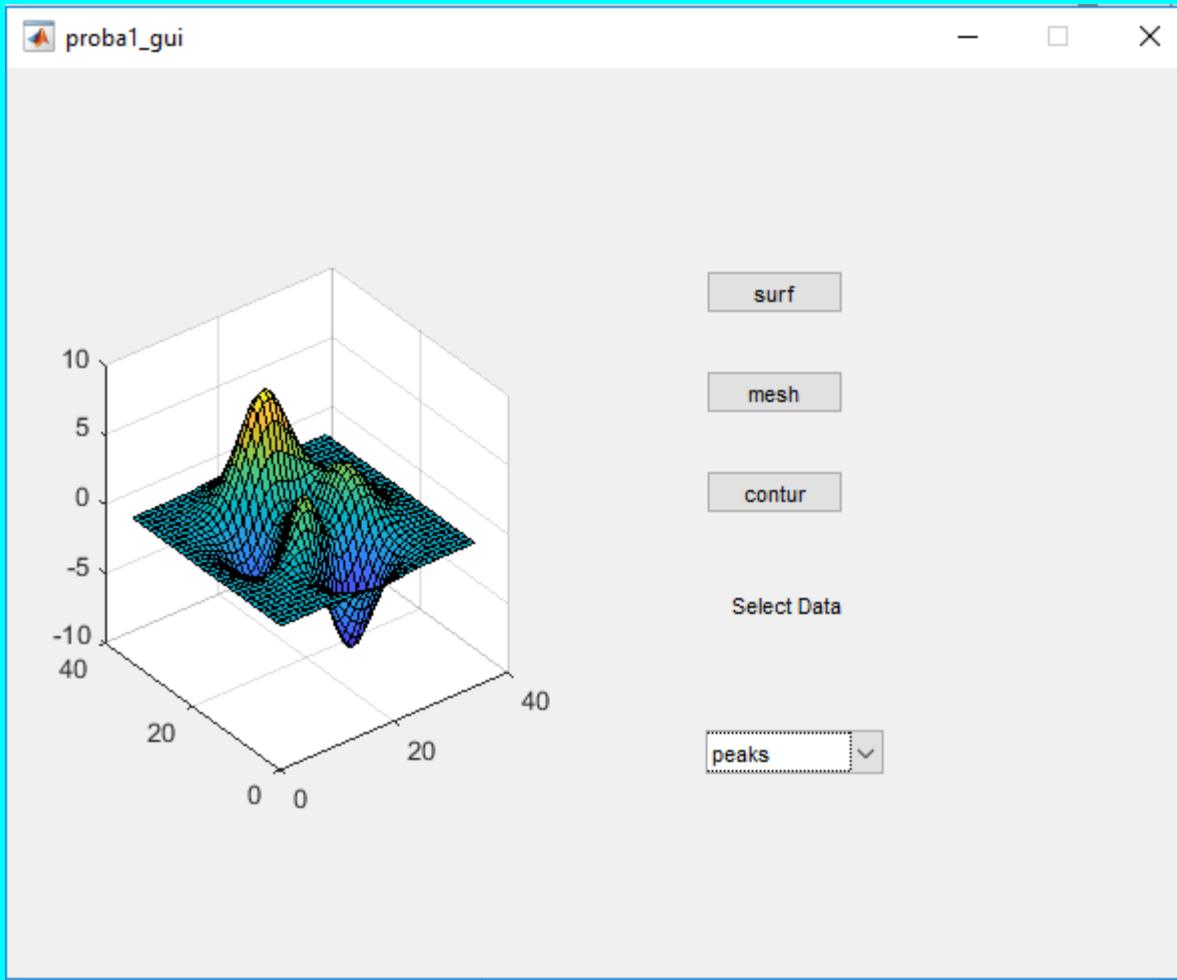
- Ovaj odeljak vam pokazuje kako generisati podatke koji će se iscrtati (grafik funkcije) kada korisnik klikne dugme. Funkcija otvaranja inicijalizuje korisnički interfejs kada se otvori, i to je prvi povratni poziv - callback u svakoj kodnoj datoteci koju generiše GUIDE.
- U ovom primeru u funkciju otvaranja dodajemo kod koji stvara tri skupa podataka. Kod koristi MATLAB funkcije peaks, membrane i sinc.
- Otvorite funkciju proba1_gui.m u MATLAB Editoru.
- Editor sada prikazuje ovaj kod:

Kreiranje jednostavne aplikacije korišćenjem guide - kod

```
%Na mestu skripta proba1_gui.m koji počinje sa
% --- Executes just before simple_gui is made visible.
function proba1_gui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to simple_gui (see VARARGIN)
%ubacujemo prvi kod za crtanje grafika surf
% Create the data to plot.
handles.peaks=peaks(35);
handles.membrane=membrane;
[x,y] = meshgrid(-8:.5:8);
r = sqrt(x.^2+y.^2) + eps;
sinc = sin(r)/r;
handles.sinc = sinc;
% Set the current data value.
handles.current_data = handles.peaks;
surf(handles.current_data)
.....
```

- Prvih šest izvršnih reda u kodu stvaraju podatke koristeći MATLAB funkcije peaks, membrane i sinc. Oni pamte podatke u strukturu markera - handles, argument koji se daje svim povratnim pozivima - callbacks. Povratni pozivi definisani za tastere mogu dohvatiti podatke iz strukture handles.
- Posljednja dva reda stvaraju trenutnu vrednost podataka i postavljaju ih na peaks, a zatim prikazuju grafikon surfanja za peaks. Sledeća slika prikazuje kako korisnički interfejs izgleda kada se zapamti i prikaže.

Kreiranje jednostavne aplikacije korišćenjem guide - kod

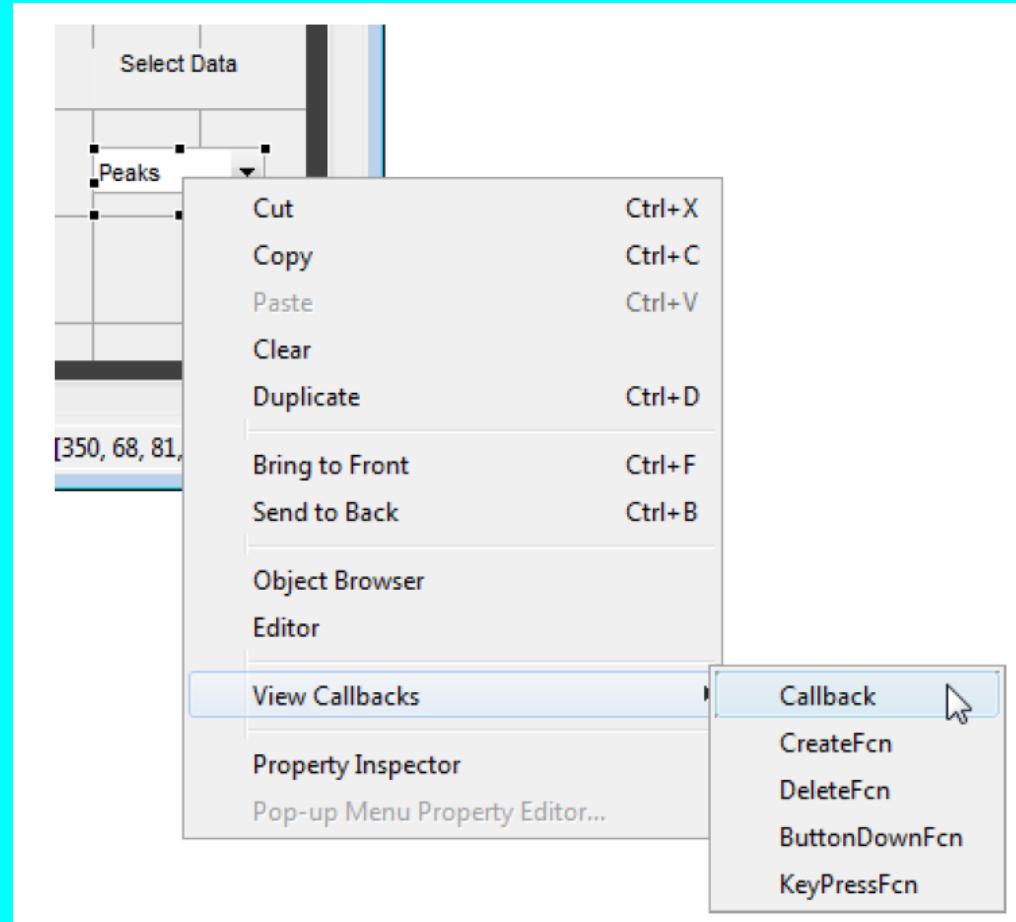


Kreiranje jednostavne aplikacije korišćenjem guide - kod

Ponašanje Pop-Up meni koda

Pop-Up Menu predstavlja opcije za iscrtavanje podataka. Kada korisnik odabere jedan od tri dijagrama, softver MATLAB postavlja svojstvo Pop-Up Menu Value property na indeks odabranog niza. Pop-up menu callback čita pop-up menu Value property da bi odredio stavku koju izbornik trenutno prikazuje i prema tome postavlja handles.current_data.

- 1 Prikazati pop-up menu callback u MATLAB Editoru. U GUIDE Layout Editoru, desnim tasterom kliknuti na pop-up menu component, i selektovati **View Callbacks >Callback**.



Kreiranje jednostavne aplikacije korišćenjem guide - kod

2 Dodajemo sada novi kod u popupmenu5_Callback posle komantara % handles...

% --- Executes on selection change in popupmenu1.

```
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

%dodajemo drugi kod

% Determine the selected data set.

```
str = get(hObject, 'String');
```

```
val = get(hObject, 'Value');
```

% Set current data to the selected data set.

```
switch str{val};
```

case 'peaks' % User selects peaks.

```
handles.current_data = handles.peaks;
```

case 'membrane' % User selects membrane.

```
handles.current_data = handles.membrane;
```

case 'sinc' % User selects sinc.

```
handles.current_data = handles.sinc;
```

```
end
```

% Save the handles structure.

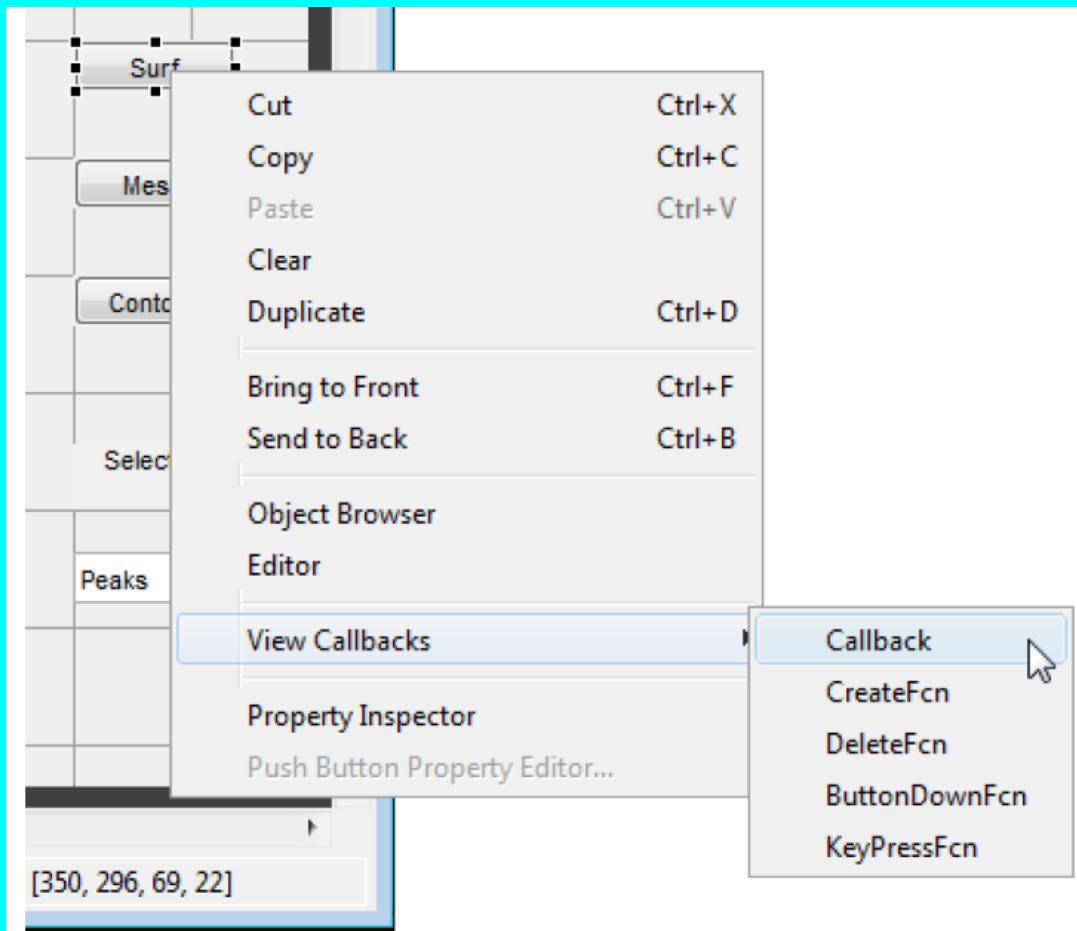
```
guidata(hObject,handles)
```

Kreiranje jednostavne aplikacije korišćenjem guide - kod

Ponašanje Push Button

Svako od dugmadi stvara različitu vrstu dijagrama koristeći podatke odredjene trenutnim izborom u pop-up meniju. Push button callbacks dobijaju podatke iz handles strukture i zatim ih crtaju.

1 Prikažite surf push button callback u MATLAB Editoru. U editoru desnom tasterom miša kliknite dugme Surf, a zatim odaberite View Callbacks > Callback



Kreiranje jednostavne aplikacije korišćenjem guide - kod

2 Editorski dodajemo sada novi kod u function pushbutton1_Callback posle komantara % handles...

```
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Display surf plot of the currently selected data.
surf(handles.current_data);
```

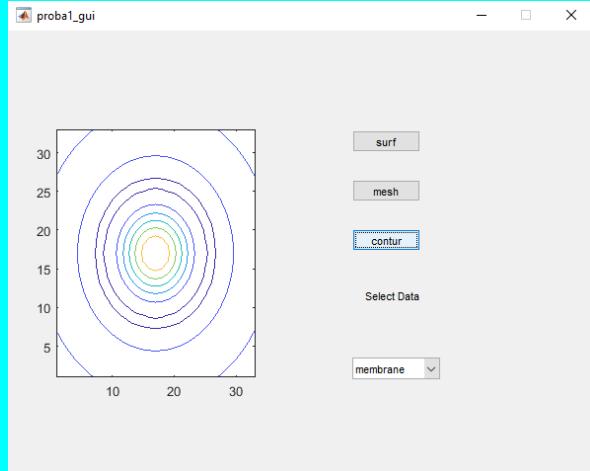
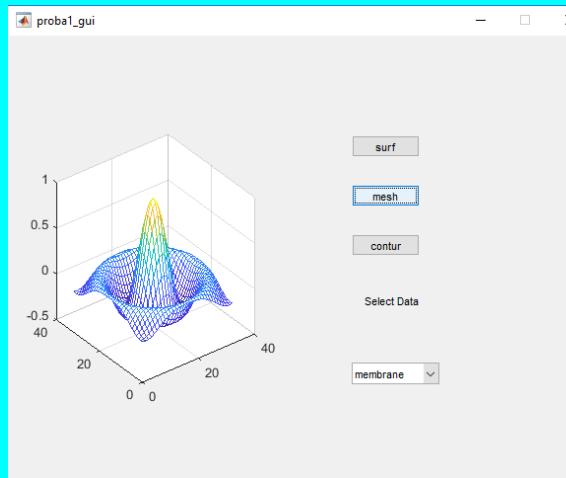
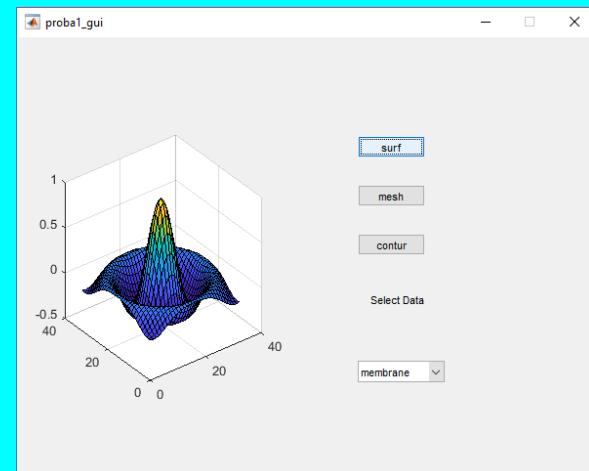
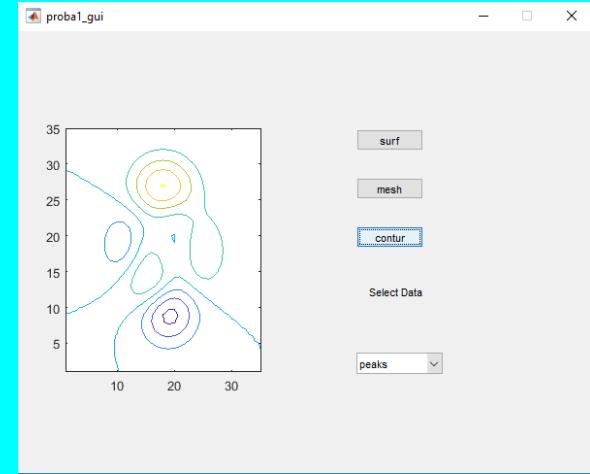
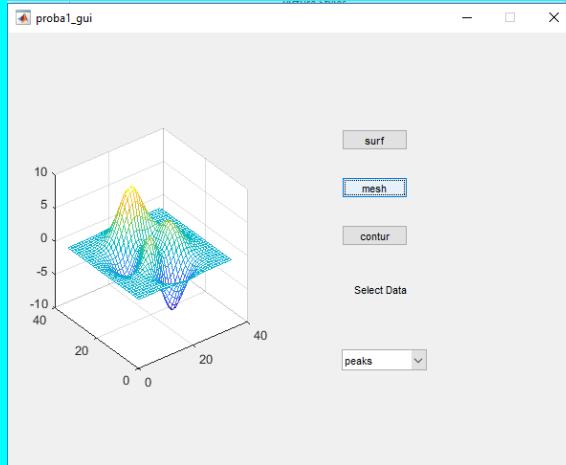
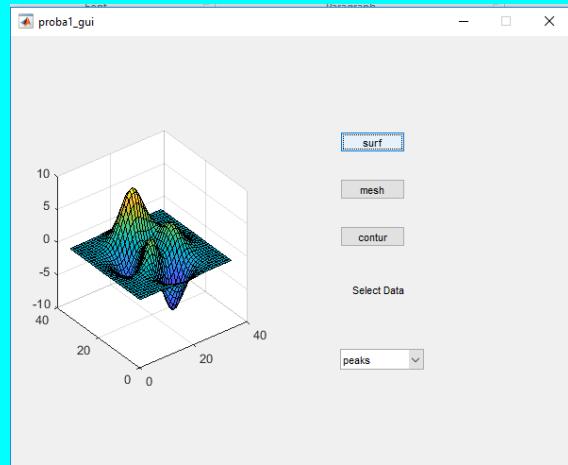
***** ISTO PONAVLJAMO i za function pushbutton2_Callback

```
% Display surf plot of the currently selected data.
mesh (handles.current_data);
```

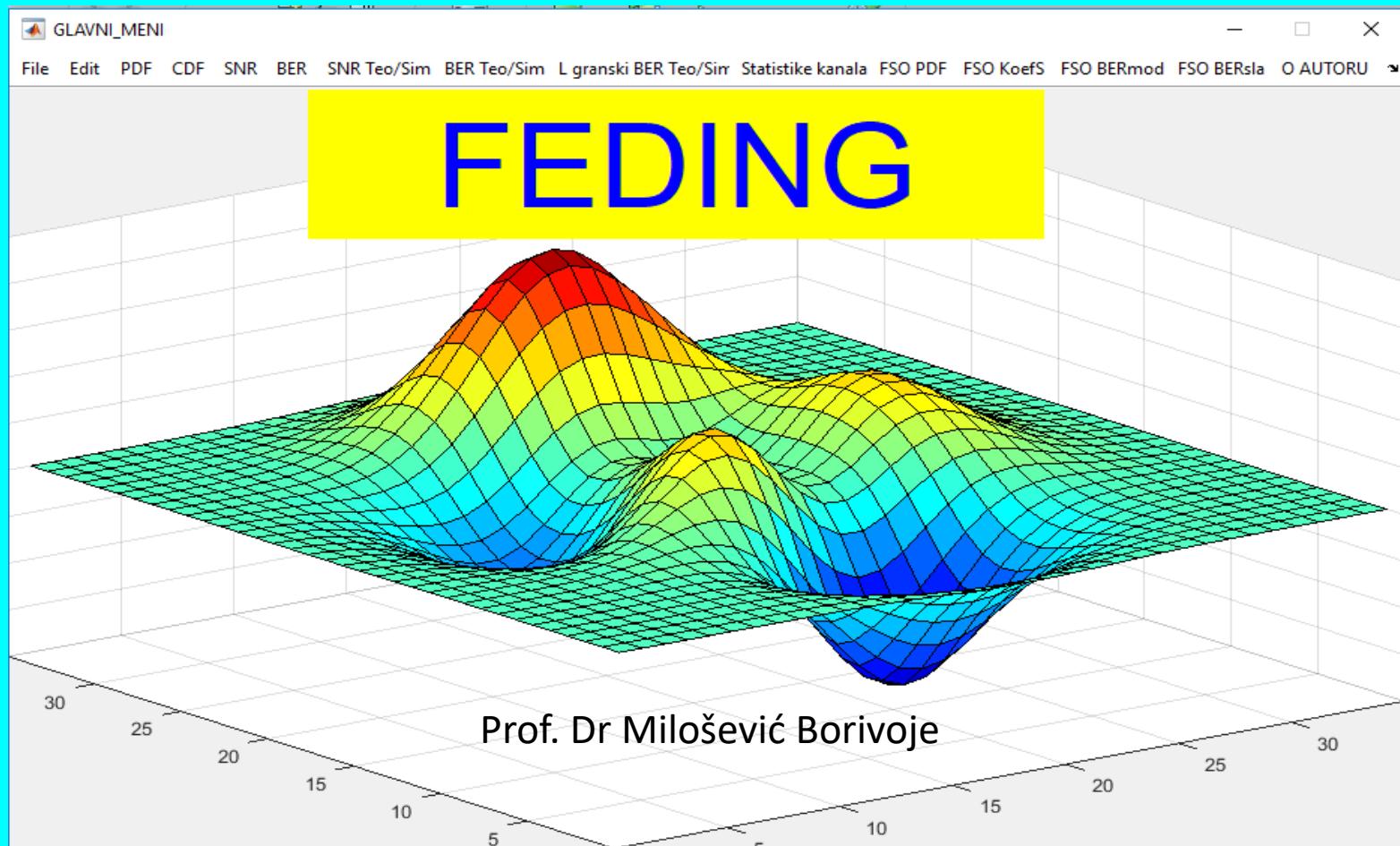
***** ISTO PONAVLJAMO i za function pushbutton3_Callback

```
% Display surf plot of the currently selected data.
contour
(handles.current_data);
```

REZULTATI GUI



Izgled jednog gotovog GUI uradjenog za Windows okruženje



Izgled jednog gotovog GUI uradjenog za Windows okruženje

