


Sveučilište u Zagrebu
Fakultet prometnih znanosti
Zavod za inteligentne transportne sustave
Katedra za primijenjeno računarstvo

	Vježba:	#1
	Kolegij:	Umjetna inteligencija
	Tema:	Uvod u Matlab
	Vježbu pripremili:	Doc. dr. sc. Edouard Ivanjko Martin Gregurić, mag. ing. Mario Buntić, mag. ing.

Upute za izradu vježbi/zadataka

Prije dolaska na vježbu potrebno je proučiti pripremu za vježbu. Vježbe je potrebno izraditi pomoću alata MATLAB. Vrijeme za izradu zadanih zadataka iznosi 90 minuta. Na kraju vježbi riješene zadatke (📁) je potrebno predati preko Merlin sustava.

Cilj vježbe

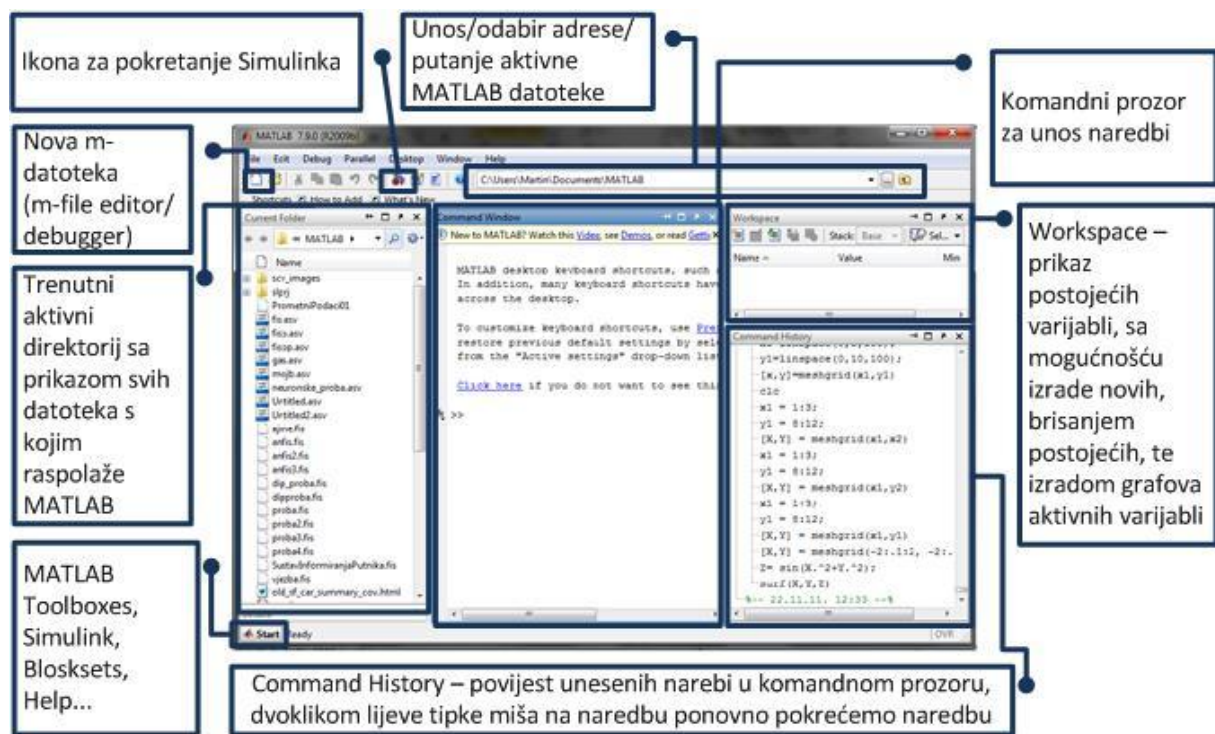
Upoznavanje s osnovama rada u MATLAB okruženju. Organizacija i struktura podataka. Aritmetičke, relacijske i logičke operacije. Grafovi i prikazi krivulja. Osnovna sintaksa programskog jezika MATLAB.

Opis vježbe

Ime programskog sustava MATLAB dolazi od skraćenice njegovog primarnog svojstva - matričnog kalkulatora, odnosno laboratorija (engl. *MATrix LABoratory*). Općenito MATLAB programski sustav možemo definirati kao okruženje sa programskim jezikom namijenjeno tehničkim izračunima. Obuhvaća izračune, vizualizaciju i programiranje u jednostavnoj okolini. Osnovna svojstva MATLAB-a:

- Matrični kalkulator interpreterskog tipa;
- Grupiranje naredbi u skripte i funkcije;
- Otvorenost – razvoj skupova alata za rješavanje specifičnog tipa problema (engl. „toolbox“);
- Testiranje algoritama u stvarnom vremenu.

Osnovna razvojna okolina MATLAB-a, odnosno početni prozor prikazan je na slici 1.



Slika 1. Osnovni elementi razvojne okoline MATLAB-a

Pokretanje MATLAB-a

MATLAB se može pokrenuti na tri načina:

- Dvostrukim klikom lijeve tipke miša na ikonu MATLAB-a smještenoj na radnoj površini (engl. „desktop“) ukoliko postoji;
- Jednostrukim klikom tipke miša na glavni izbornik – „Start“, te odabirom sljedećih opcija: „Programs“ (ili „All Programs“ u novijim verzijama Windows operativnog sustava) -> MATLAB -> R2009b -> MATLAB R2009b (oznaka inačice npr. „R2009b“ ovisi o trenutnoj instaliranoj inačici programskog paketa Matlab);

- Upisom naredbe „matlab“ u „run“ prozor unutar „Start“ izbornika ili u novijim inačicama Windows operativnog sustava u donjem dijelu „Start“ izbornika u polje u kojem piše „*Search programs and files*“.

Iz MATLAB programskog okruženja se izlazi također na tri moguća načina:

- Unosom naredbe „quit“ u komandni prozor i potvrdnom unosa tipkom Enter;
- Odabirom opcija: „File“ -> „Exit“;
- Zatvaranje osnovnog prozora MATLAB-a pritiskom lijeve tipke miša na ikonicu za zatvaranje aplikacije.

Uvod u osnovno korištenje MATLAB programskog sustava

Za početak korištenja MATLAB okruženja potrebno je naučiti stvoriti varijablu, kao osnovno polazište rješavanja bilo kojeg problema. Ime varijable može biti bilo koji alfanumerički znak, te znak „_“ (engl. „underscore“), a ime varijable mora početi slovnim znakom. Pri imenovanju varijable napomenimo kako MATLAB razlikuje velika i mala slova (engl. „case sensitivity“), pa su primjerice varijable „Broj“ i „broj“ potpuno različite i međusobno nezavisne.

Primjerice stvorimo varijablu koja sadrži matricu „b“:

$$b = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

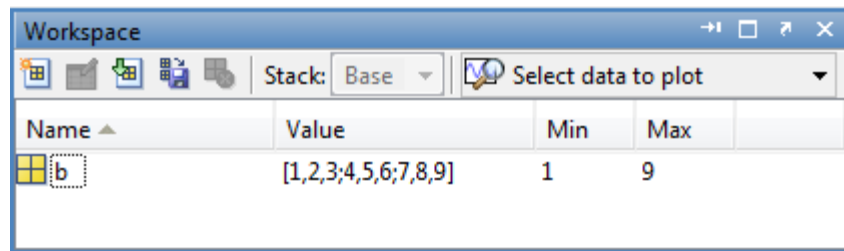
U MATLAB kodu izradit ćemo ovakvu matricu i spremiti njene vrijednosti u varijablu pod imenom „b“ na sljedeći način:

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

Prazan razmak između brojeva označava elemente retka matrice, a znak „;“ označava kraj jednog retka i početak drugog retka. Nakon pritiske tipke Enter za potvrdu unosa, dobit ćemo sljedeći ispis u komandnom prostoru:

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

Varijabla pod imenom „b“ prikazat će se u Workspace-u sa njezinim imenom (engl. „Name“), vrijednostima (engl. „Value“), te minimalnoj (engl. „Min“) i maksimalnoj (engl. „Max“) vrijednosti. Kao što je to prikazano na slici 2.

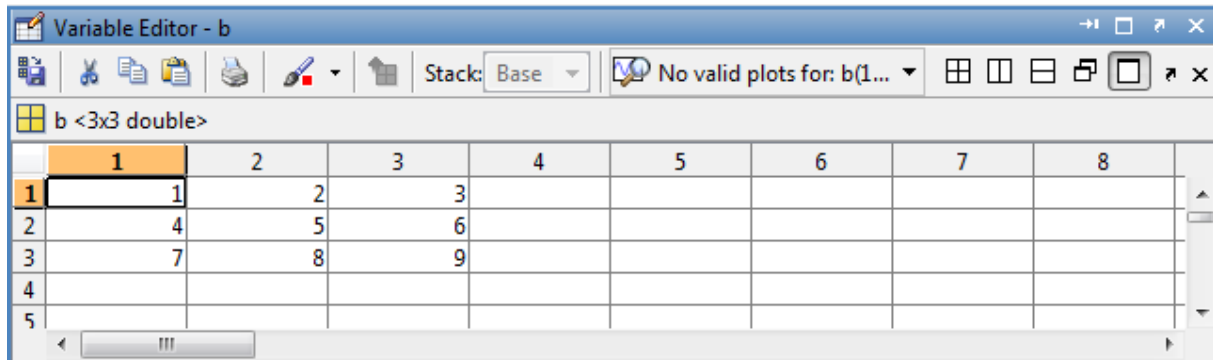


Slika 2. Prikaz varijable „b“ u Workspace-u

Ukoliko želimo stvaranje varijable samo u Workspace-u, bez ispisa njenih vrijednosti u komandnom prostoru unese se linija koda sa završnim znakom „;“.

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

Ukoliko želimo dodatno promijeniti ili dodati neke vrijednosti u novo stvorenu varijablu „b“, dvostrukim klikom lijeve tipke miša na ime varijable u Workspace-u otvorit će se alat „Variable Editor“ kako je to prikazano na slici 3.



Slika 3. Prikaz varijable „b“ u alatu „Variable Editor“

Ukoliko se želi obrisati jedna ili više varijabli iz Workspace-a koriste se iduće naredbe koje se unose u MATLAB radni prozor:

```
>>clear all      - briše sve varijable iz Workspace-a;
>>clear a b c   - briše varijable a, b i c iz Workspace-a.
```

Varijablu je moguće izbrisati i u Workspace-u, jednostrukim klikom desne tipke miša na ime varijable koja se želi obrisati, te odabirom opcijom „Delete“ sa padajućeg izbornika. Komandni prostor se briše unosom naredbe „clc“ u komandni prostor i potvrdom unosa tipkom Enter.

Ukoliko želimo primjerice odrediti dimenziju matrice varijable „b“ učinit će se to sa idućom naredbom u komandnom prostoru:

- $[m,n] = \text{size}(b)$ - rezultat provedbe ove linije koda su dvije varijable „m“ i „n“. Prva varijabla „m“ označava broj redaka, a varijabla „n“ označava broj stupaca. Argument funkcije „size“ -„b“ označava ime varijablu u kojoj je spremljena

matrica čije dimenzije želimo znati. Nakon potvrde unosa tipkom <enter> u komandnom prostoru dobit ćemo rezultat:

```
m =  
    3  
n =  
    3
```

Što znači kako varijabla „b“ sadržava matricu 3x3 što je i istina. Ekstrakcija dijela matrice provodi se sa naredbama:

- *ime_var(n1:n2,m1:m2)* – gdje je „ime_var“ – ime varijable u kojoj je spremljena matrica. Argument „n1“ označava prvi broj (indeks) elementa u retku, „n2“ zadnji broj elementa u retku. Znak „:“ između „n1“ i „n2“ označava označavanje svih elemenata između prvog i zadnjeg elementa u retku. Dok „m1“ označava prvi broj elementa u stupcu, a „m2“ zadnji broj elementa u stupcu. Znak „“ označava odvajanje argumenta koji se odnose na retke i stupce.

Pa primjerice ukoliko želimo iz matrice u varijabli „b“ ekstrahirati prvi stupac:

$$b = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 4 \\ 7 \end{bmatrix}$$

Koristit ćemo naredbu:

```
>> b(:,1)
```

Ukoliko u argumentu za broj redaka i/ili stupaca nema vrijednosti već samo stoji znak „:“, to znači da su odabrani svi redci i/ili stupci. Želimo li iz matrice „b“ ekstrahirati vrijednosti brojeva 5, 6, 8, 9:

$$b = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$$

Koristit ćemo naredbe:

```
>> b(2:3,2:3)
```

ili

```
>>b (2:m,2:n)
```

Dok ćemo zadnji redak u matrici „b“:

$$b = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow [7 \ 8 \ 9]$$

Ekstraktirat uz pomoć iduće naredbe:

```
>> b(end,1:end)
```

Funkcija „end“ poprima vrijednost zadnjeg elementa u retku i/ili stupcu. MATLAB okruženje omogućuje grupiranje podataka različitih tipova i dimenzija u jednu cjelinu. Takve varijable sa nazivaju polja elementa. Primjerice stvorimo jednu varijablu polja elementa „T“ u koju ćemo spremiti varijablu u kojoj su spremljeni znakovi – „Tok“, cjelobrojna vrijednost – „2“, te polje cjelobrojnih vrijednosti (jednodimenzionalna matrica).

```
>> T = {'Tok',2,[1.2 1.25 1.1 1 1.2]}
```

Nakon potvrde unosa tipkom Enter, dobivamo sljedeći prikaz:

```
T =
      'Tok'      [2]      [1x5 double]
```

Vidi se da varijabla „T“ sadržava tri polja sa različitim tipovima i dimenzijama podataka u njoj. Ukoliko se želi pristupiti određenom elementu u polju, primjerice polju tri u kojem se nalazi polje elemenata 1 x 5 unesimo iduću naredbu:

```
>> T{3}
```

Rezultat će biti:

```
ans =
1.20 1.25 1.10 1.00 1.20
```

Uviđamo iz rješenja kako ukoliko ne stvorimo vlastitu varijablu u kojoj će biti spremljeno rješenje gornjeg koda, MATLAB će stvoriti novu privremenu varijablu „ans“ u kojoj će biti spremljen rezultat. Pristup primjerice drugom elementu u polju izvodi se idućom naredbom:

```
>> T{3}(2)
```

Rezultat će biti:

```
ans =
1.2500
```

Slični učinak možemo postići sa strukturnom varijablom koju imenujemo nazivom „Mjerenje“:

```
Mjerenje = struct('Name','Tok','Ts',2,'data',[1.2 1.25 1.1 1 1.2])
```

Nakon pritiske tipke Enter:

```
Mjerenje =  
Name: 'Tok'  
Ts: 2  
data: [1.2000 1.2500 1.1000 1 1.2000]
```

Pristup elementima trećeg polja u slučaju primjene strukturne varijable omogućuje se idućom naredbom:

```
>> Mjerenje.data  
  
ans =  
  
1.2000 1.2500 1.1000 1.0000 1.2000
```

Dok se pristup drugom elementu u polju izvodi se idućom naredbom:

```
>> Mjerenje.data(2)  
  
ans =  
  
1.2500
```

MATLAB podržava aritmetičke operatore koji su prikazani u tablici 1.

Tablica 1. Aritmetički operatori u MATLAB-u sa opisom i prioritetom izvođenja

Prioritet	Operator	Opis
1.	()	Grupiranje i prioritet
2.	'	Konjugiranje i transponiranje
	.'	Transponiranje
3.	^	Potenciranje
	.^	Pojedinačno potenciranje elemenata
4.	*	Matrično množenje
	.*	Množenje među elementima matrica
	/	Desno dijeljenje ($X/Y=X*Y^{-1}$)
	\	Lijevo dijeljenje ($X\Y=X^{-1}*Y$)
	./	Dijeljenje između pojedinih elemenata matrica
5.	+	Zbrajanje
	-	Oduzimanje

Primjerice ukoliko želimo transponirati matricu u varijabli „b“, te ju spremiti u varijablu „a“ koristimo idući liniju koda:

```
a = b.'
```

```
a =
```

```
    1    4    7
```

```
    2    5    8
```

```
    3    6    9
```

Ili primjerice pomnožiti elemente matrice pohranjenoj u varijabli „b“ sa elementima transponirane matrice u varijabli „b“, te rezultat spremiti u varijablu „c“:

```
>> c = b .* a
```

```
c =
```

```
    1    8   21
```

```
    8   25   48
```

```
   21   48   81
```

Također u MATLAB programskom jeziku postoje i relacijski operatori koji kao rezultat operacije vraćaju logičku vrijednost (0 ili 1). Tablica 2. prikazuje podržane relacijske operatore.

Tablica. 2. Relacijski operatori u MATLAB-u sa opisom i primjerom

Operator	Opis	Primjer
<	Manje	a<b
<=	Manje ili jednako	a<=b
>	Veće	a>b
>=	Veće ili jednako	a>=b
==	Jednako	a==b
~=	Različito	a~=b

Provedite negaciju matrice u varijabli „a“ sa matricom u varijabli „b“, brojevi u matrici koji nisu isti bit će označeni s „1“ što znači da je negacija istinita, a sa „0“ ako su isti što znači da negacija nije istinita:

```
a ~= b
```

```
ans =
```

```
    0    1    1
```

```
    1    0    1
```

```
    1    1    0
```

Sad usporedimo koji su elementi matrice „a“ veći od elemenata matrice „b“ idućom linijom koda:

```
a > b
```



```
ans =
     0     1     1
     0     0     1
     0     0     0
```

Uz relacijske operatore koriste se često i logički operatori. Popis, primjer i opis pojedinih logičkih operatora u MATLAB okruženju može se vidjeti u tablici 1.3.

Tablica. 1.3. Primjer i opis logičkih operatora koje podržava MATLAB

Operator	Opis	Primjer
&	Logičko „I“	a & b
	Logičko „ILI“	a b
~	Logičko „NE“	~ b
xor	Logičko isključivo „ILI“	xor(a,b)

Primjerice pronađimo elemente matrice spremljene u varijablu „a“ koji su veći od vrijednosti „2“, ali i koji su manji od vrijednosti „3“ u matrici spremljenoj u varijabli „b“:

```
>> a>2 & b<3
ans =
     0     1     0
     0     0     0
     0     0     0
```

MATLAB podržava naredbe odluka i ponavljanja. Prvo ćemo obraditi AKO („If“) naredbu. Sintaksa AKO naredbe je:

```
if logicki_izraz
    naredbe;
elseif logicki_izraz
    naredbe;
else
    naredbe;
end
```

Za primjer imamo iduću AKO naredbu:

```
a = 2;
b = 0;

if a>3 & a<5
    b=0;
else
    b=5;
end;
```

Rezultat „ans“ će imati vrijednost „5“. MATLAB, programsko okruženje podržava i ZA „for“ naredbu čija je sintaksa sljedeća:

```

for varijabla=pocetna_vrijednost:krajnja_vrijednost
    naredbe;
end;
ili
for varijabla=pocetna_vrijednost:uvecanje:krajnja_vrijednost
    naredbe;
end;

```

Napišimo programski kod sa ZA petljom koji će izraditi matricu čiji će elementi redaka biti za vrijednost „2“ veći od prethodnog elementa:

```

for i=1:10
    for j=1:2:6
        a(i,j)=i+j;
    end;
end;

```

Rezultat je matrica:

```

a =
     2     0     4     0     6
     3     0     5     0     7
     4     0     6     0     8
     5     0     7     0     9
     6     0     8     0    10
     7     0     9     0    11
     8     0    10     0    12
     9     0    11     0    13
    10     0    12     0    14
    11     0    13     0    15

```

Zaključimo kako su u drugoj ZA petlji sa uvećanjem vrijednosti za „2“ brojevi koji nisu obuhvaćeni uvećanjem u rješenju, a manji su od broja „6“ pišu u matrici kao „0“ bez obzira na trenutnu vrijednost prve ZA petlje.

Napravljenoj matrici možemo pridodati DOK naredbu („While“) koja ima sljedeću sintaksu:

```

while logicki_izraz
    naredbe;
end

```

Te za primjer, neka se vrijednost varijable prve ZA petlje ukoliko je veća od vrijednosti „1“ poveća za pet puta i spremi u varijablu matrice „a“ na početku „i“-tog reda, pa se potom vrijednost i umanj za vrijednost „2“:

```

while i>1
    a(i)=5*i
    i=i-2;
end;

```

Rezultat je matrica:

```

a =

```

2	0	4	0	6
10	0	5	0	7
4	0	6	0	8
20	0	7	0	9
6	0	8	0	10
30	0	9	0	11
8	0	10	0	12
40	0	11	0	13
10	0	12	0	14
50	0	13	0	15

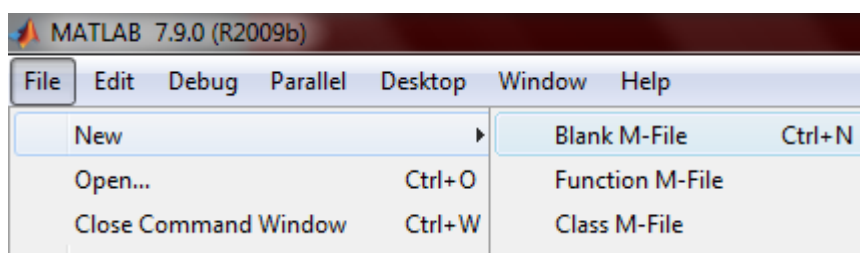
MATLAB također podržava izradu i korištenje postojećih funkcija. Postoje tri glavne vrste funkcija:

- Interne funkcije;
- Funkcije unutar skupa alata (engl. „Toolboxa“);
- Funkcije izrađene od strane korisnika.

Interne funkcije se dijele još i na:

- Elementarne matematičke funkcije;
 - Trigonometrijske i ciklotometrijske funkcije;
 - Logaritamske i hiperbolne funkcije;
 - Ostale funkcije;
- Funkcije za obradu vektora i matrica;
- Funkcije za rad s polinomima;
- M-funkcije.

U ovim vježbama koristit ćemo neke interne funkcije prilikom izrada vlastitih M-funkcija. M-funkcija pripada u skup M-datoteka koje definiramo kao tekstualne datoteke koje sadrže MATLAB naredbe definirane od strane korisnika. U M-datoteke uz M-funkcije pripadaju i M-skripte. Obje vrste M datoteka stvaraju se u M-file editor-u koji se može pokrenuti ikonicom objašnjenom na slici 1.1. ili odabirom opcija: „File“ -> „New“ -> „Blank M-File“ kako je prikazano na slici 4.:

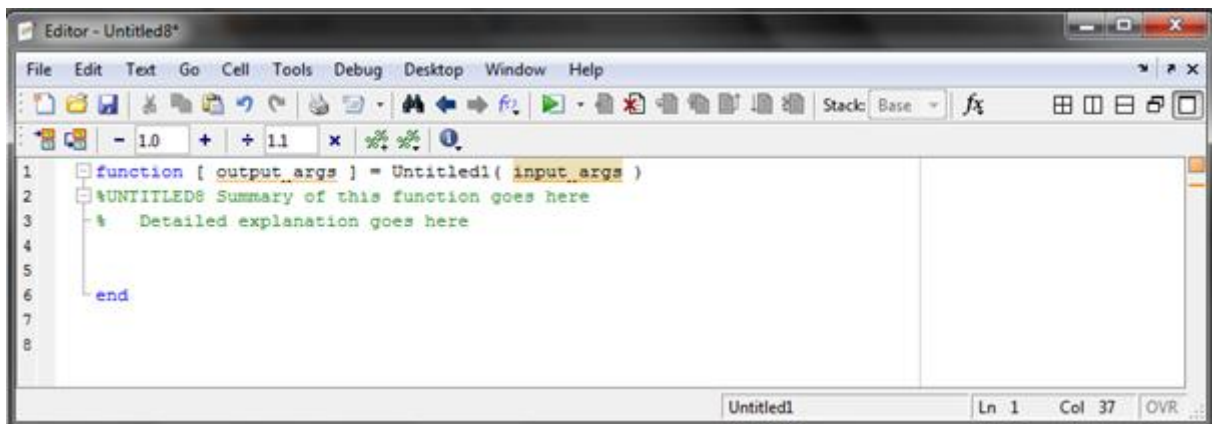


Slika 4. Prikaz pristupa M-file editor-u

Potrebno je napomenuti kako je pri pisanju M-datoteka moguće komentirati određenu liniju koda radi olakšanja budućih ispravaka ili dorada koda. Komentari počinju sa znakom „%“. M-funkcije u M-file editor-u počinju sa idućom sintaksom:

```
function izlaz=ime_funkcije (ulaz1,ulaz2,...,ulazn)
%komentar koje se ispisuje na help ime_funkcije
Naredbe i pozivi drugih funkcija
izlaz=...
```

Ili umjesto odabira opcija: „File“ -> „New“ -> „Blank M-File“ kako je prikazano na slici 1.4., odaberimo: „File“ -> „New“ -> „Function M-File“. U tom slučaju otvoriće se M-file editor, ali sa početnom sintaksom za izradu funkcije kao što je to prikazano na slici 1.5.:



Slika 1.5. Prikaz M-file editora kao „Function M-File“

Otvorimo M-file editor kao „Function M-File“ i izradimo jednostavnu funkciju za izračun prosječnog broja nekog vektora sa ispitivanjem prihvatljivog ulaza:

```
function y = prosjecan(x)
% Komenatr funkcije koji se ispisuje kada utipkamo help prosjecan
[m,n] = size(x);
if ~(m == 1) | (n == 1) | (m == 1 & n == 1)
    error('Ulaz mora biti vektor!')
end
y = sum(x)/length(x);      % Stvarno racunjanje
```

Pozivanje funkcije ima iduću sintaksu:

```
ime_fun(ulaz1,ulaz2,...,ulazn)
ili
feval('ime_fun', ulaz1,ulaz2,...,ulazn)
```

U komandnom prostoru definirajmo varijablu „z“ sa vektorom koji će sadržavati sve brojeve od „1“ do „99“:

```
z = 1:99;
```

Te pozovimo funkciju „prosjecan“ sa argumentom u kojoj će biti vrijednosti vektora varijable „z“:

```
prosjecan(z)
```

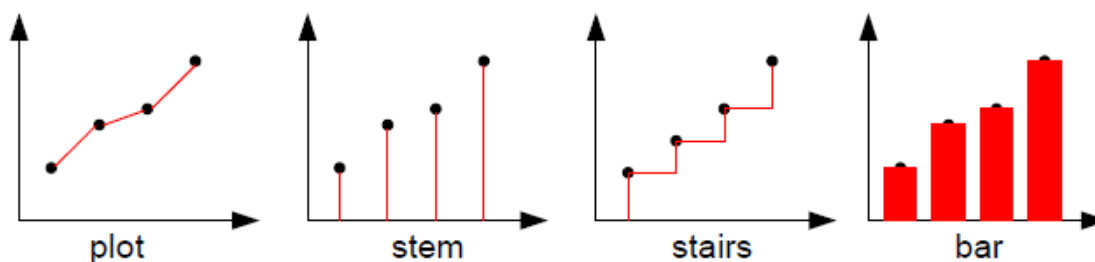
Rezultat će biti:

```
ans =
    50
```

MATLAB omogućuje grafički prikaz rezultata, te sadrži niz funkcija za opis grafova, njegovih osi i pripadajućih krivulja. Graf pojedinog rješenja prikazuje se u slici

(eng. "Figure"), tj. zasebnom prozoru. Grafičke funkcije MATLAB-a dijelimo na dvodimenzionalne i trodimenzionalne prikaze.

Za iscrtavanje dvodimenzionalne slike koristi se četiri osnovne funkcije: „plot“, „bar“, „stem“ i „stairs“. Naredba „plot“ spaja susjedne točke grafičkog prikaza ravnom linijom, „bar“ daje stupčasti prikaz, dok se kod „stairs“ prikaza dobiva grafički prikaz u kojem se susjedne točke povezuje stepenastim načinom. Nareda „Stem“ svaki podatak prikazuje vertikalnom linijom čija dužina odgovara iznosu podataka. Sam podatak označen je kružićem na vrhu linije. Njihove osobine grafički su prikazane na slici 1.6.



Slika 1.6. Prikaz osnovnih naredbi za izradu slika u MATLAB-u

Sintaksa naredbi je:

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

Ukoliko se vektor „x“ i opcije izostave, tada se prikazuju točke vektora „y“ u ovisnosti o njihovom rednom broju. Postoji li i vektor „x“ i „vektor „y“, tada ti vektori tvore niz uređenih parova točaka. Vektori „x“ i „y“ moraju imati isti broj redaka. Ako postavimo „y“ kao matricu umjesto vektora, koja ima isti broj redaka kao vektor „x“, naredba „plot“ crta po jednu krivulju za svaki stupac vektora „y“. Opcije su znakovi ili nizovi znakova koji određuju boju i tip linije kako je prikazano na tablici 4.

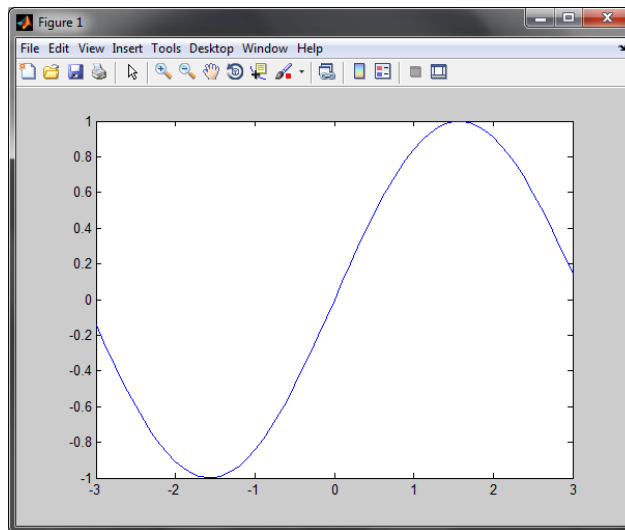
Tab 4. Opcije za određivanje boje i tipa linije

Oznaka	Boja	Oznaka	Tip linije
y	Žuta	-	Puna
m	Ljubičasta	:	Točkasta linija
c	Svijetloplava	-.	Crta točaka
r	Crvena	--	Isprekidana linija
g	Zelena		
b	Plava		
w	Bijela		
k	Crna		

Kao primjer za izradu grafa pomoću naredbe „plot“ može se uzeti sinusnu krivulju u intervalu vrijednosti od „-3“ do „3“. Razmak između brojeva u intervalu neka bude „0,1“. Kod za izradu takvog grafa je sljedeći:

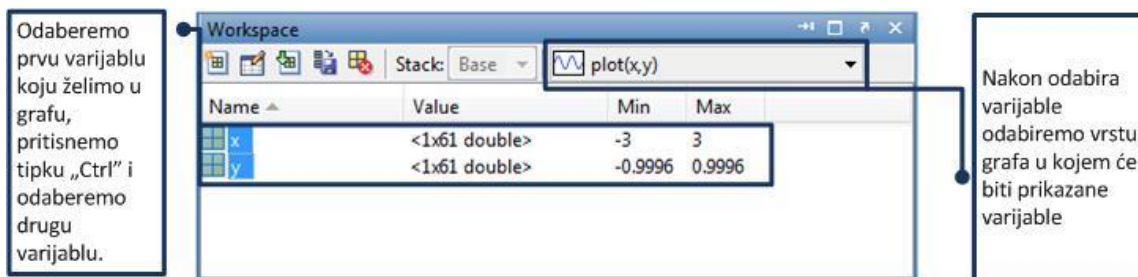
```
x = -3:.1:3;
y = sin(x);
plot(x,y)
```

A rezultat će biti idući graf prikazan na slici 7.



Slika 7. Prikaz sinusne krivulje

Isti graf možemo stvoriti pomoću Workspace-a odabirom varijabli koje želimo prikazati u grafu, te potom izaberemo opciju vrste izrade grafa kako je prikazano na slici 8.

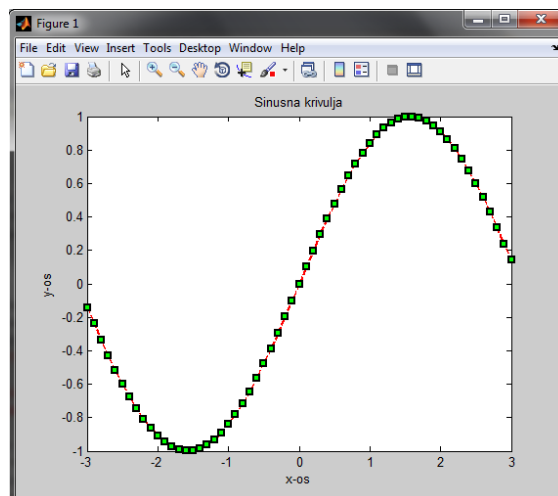


Slika 8. Izrada grafa pomoću Workspace-a

Ukoliko želimo stvoriti isti graf, ali s točkama označenim zelenim pravokutnicima sa crnim rubom i crvenom isprekidanom krivuljom. Također, želimo i ispisati ime grafa (engl. „title“), sa imenom x-osi (engl. „xlabel“) i y-osi (engl. „ylabel“) unesimo u komandni prozor idući kod:

```
x = -3:.1:3; %vektor s brojevima između broja -3 i 3 za 0.1
povećanje
y = sin(x) ; %funkcija sin za izračun sinusa iz vektora x
plot(x,y, '--
rs', 'LineWidth',2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'Marke
rSize',5)
% '--rs'      iscrtavanje crtkane crvene linije s kvadratima,
'LineWidth',2 debljina linije je 2,
'MarkerEdgeColor','k' rub kvadrata markera je crne boje,
'MarkerFaceColor','g', boja markera je zelena, 'MarkerSize',5
veličina markera je 5.
xlabel('x-os') %x-os će biti ispisana kao „x-os“
ylabel('y-os') %y-os će biti ispisana kao „y-os“
title('Sinusna krivulja') %naslov grafa će biti „Sinusna krivulja“
```

Dobiva se slika kao što je prikazana na slici 9.

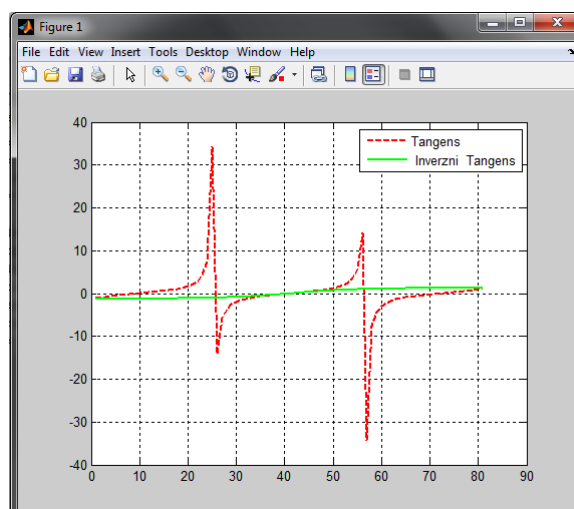


Slika 9. Prikaz grafički uređene sinusne krivulje

Ukoliko se želi izraditi graf na kojem će biti prikazane dvije krivulje, primjerice inverz funkcije tangensa i tangensa, sa njihovom legendom i pomoćnom mrežom, koristit će se sljedeće linije koda:

```
z = -4:.1:4; %vektor s brojevima između broja -4 i 4 za 0.1 povećanje  
y =atan(z); %funkcija atan za izračun inverznog tangensa iz vektora z  
x= tan(z); %funkcija tan za izračun tangensa iz vektora z
```

```
plot(x, '--r','DisplayName','Tangens','LineWidth',2);  
%prikazi varijablu x , '--r' crtkanom crvenom bojom,  
'DisplayName','Tangens' u legendi će biti predstavljena imenom  
„Tangens“, 'LineWidth',2, debljine linije 2  
hold all; %zadržavanje karakteristika prethodne linije grafa kako bi  
se mogla u istim koordinatama iscrtati druga linija grafa  
plot(y, 'g','DisplayName',' Inverzni Tangens', 'LineWidth',2);  
%prikazi varijablu y , 'g' zelenom bojom, 'DisplayName','Inverzni  
Tangens' u legendi će biti predstavljena imenom „Inverzni Tangens“,  
'LineWidth',2, debljine linije 2  
hold off;%isključuje zadržavanje karakteristika linija grafa  
legend('show'); %prikazi legendu  
grid on; %prikazi pomoćnu mrežu
```

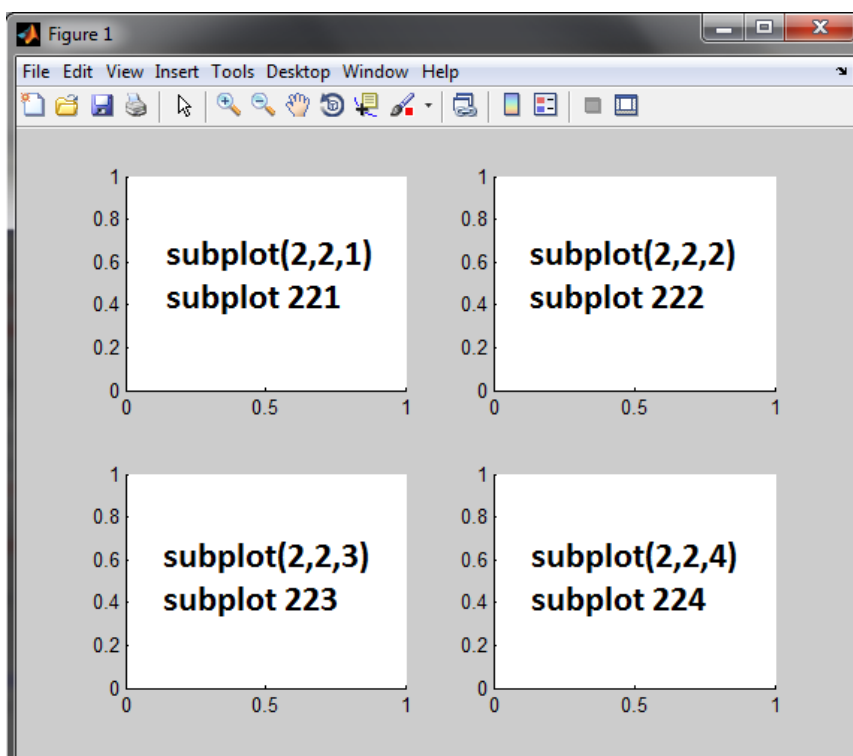


Slika 10. Prikaz dviju odvojenih linija na istom grafu

MATLAB omogućuje izradu više odvojenih grafova na istoj slici. Za izradu takvih prikaza koristimo funkciju „subplot“. Sintaksa funkcije „subplot“ je sljedeća:

[1] *subplot(m,n,p)* – argument „m“ i „n“ predstavljaju matricu m x n, tj. koliko će biti grafova na jednoj slici koji će biti raspoređeni u obliku matrice m x n. Dok argument „p“ predstavlja redni broj pojedinog grafa na slici, s time da se prvi graf u gornjem lijevom kutu slike nalazi pod rednim brojem jedan, te se redni brojevi grafova kreću po redovima s lijeva na desno.

Na idućoj slici 11. vidimo raspored grafova na slici i njihovu sintaksu pri određivanju parametra pojedinog grafa.



Slika 1.11. Pregled sintakse za pristup pojedinim grafovima na slici

Moguće je još dodatno urediti grafove na slici, tako da se primjerice graf na prvom („subplot 221“) i trećem („subplot 223“) mjestu na slici spoje u jedan, tada je naredba za izradu i pristup tom grafu sljedeća:

```
subplot(2,2,[1 3])
```

Ako želimo primjerice spojiti graf na prvom („subplot 221“) i drugom („subplot 222“) mjestu na slici u jedan, koristit ćemo iduću naredbu za izradu i pristup tom grafu:

```
subplot(2,2,1:2)
```

MATLAB podržava i izradu trodimenzionalnih (3D) grafova. Slično kao u 2D slučaju korisnik mora sam kreirati vrijednosti nezavisnih varijabli u kojima će se funkcija izračunavati. Glavni elementi za crtanje 3D grafa su „mesh“ i „surface“ funkcija.

U izradi 3D grafa koristit ćemo funkciju „meshgrid“ koja ima sljedeću sintaksu i primjenu:

[2] $[X,Y] = \text{meshgrid}(x,y)$ – funkcija pretvara vektore argumenta „x“ i „y“ u polja „X“ i „Y“ kojim će se evaluirati funkcija sa dvije varijable („x“ i „y“) u „mesh“ ili „surface“ trodimenzionalnom tipu grafa. Vrijednosti redova izlaznog polja „X“ su kopirane iz vrijednosti vektora „x“, a vrijednosti stupaca izlaznog polja „Y“ su kopirane iz vrijednosti vektora „y“.

Za primjer napravimo dvije nezavisne varijable: „x1“ koja će sadržavati vektor vrijednosti brojeva između vrijednosti „1“ i „3“, te varijablu „y1“ koja će sadržavati vrijednosti brojeva između vrijednosti „8“ i „12“. Nad tim varijablama pomoću funkcije „meshgrid“ kreirajmo polja „X“ i „Y“ za izradu 3D grafa:

```
x1 = 1:3;
y1 = 8:12;
[X,Y] = meshgrid(x1,y1)
```

Rezultat je:

```
X =
     1     2     3
     1     2     3
     1     2     3
     1     2     3
     1     2     3

Y =
     8     8     8
     9     9     9
    10    10    10
    11    11    11
    12    12    12
```

Preklapanjem ovih dvaju polju se dobiva „X“ i „Y“ koordinatni sustav ili 2D mreža svih mogućih kombinacija vrijednosti ulaznih varijabli „X“ i „Y“ u jednadžbu, kao što je prikazano u tablici 5.

Tablica 5. Prikaz mreže polja „X“ i „Y“ (koordinatnog sustava) izrađene pomoću funkcije „meshgrid“

		X koordinata		
Y koordinata	1,8	1,8	2,8	3,8
	1,9	1,9	2,9	3,9
	1,10	1,10	2,10	3,10
	1,11	1,11	2,11	3,11
	1,12	1,12	2,12	3,12

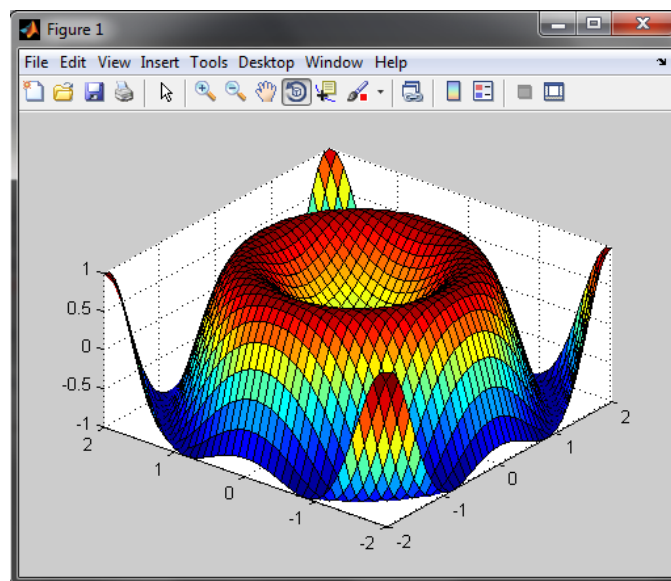
2D mreži polja „X“ i „Y“ će se dodati „Z“ koordinata, koja će predstavljati rješenja jednadžbe s određenom kombinacijom ulaznih varijabli „X“ i „Y“. Primjerice prikažimo 3D graf jednadžbe:

$$Z = \sin(X^2 + Y^2)$$

Dok će ulazne varijable „X“ i „Y“ imati raspon vrijednosti između brojeva „-2“ i „2“ uvećanih za vrijednost „0,1“. Traženi 3D graf će se ostvariti sljedećim linijama koda:

```
[X,Y] = meshgrid(-2:.1:2, -2:.1:2);  
Z= sin(X.^2+Y.^2);  
surf(X,Y,Z)
```

Rezultat je prikazan na slici 12.



Slika 12. Prikaz 3D grafa jednadžbe $Z = \sin(X^2 + Y^2)$

Zadaci (obični)

Urađene zadatke je potrebno u .zip ili .rar datoteci predati preko Merlina „Predaja 1. auditorne/laboratorijske vježbe“. Naredbe rješenja zapišite u word dokumente rezultata, te ispisa (npr. transponiranje matrice ili crtanje slike).

1.1. Zadatak

Napravite dvije proizvoljne matrice veličine 3×3 sa proizvoljnim vrijednostima i nad njima napravite sve operacije iz tablice 1 „Aritmetički operatori u MATLAB-u sa opisom i prioritetom izvođenja“.

1.2. zadatak

Napravite dvije proizvoljne matrice veličine 3×3 sa proizvoljnim vrijednostima i nad njima napravite sve operacije iz tablice 2 „Relacijski operatori u MATLAB-u sa opisom i primjerom“.

1.3. zadatak

Napravite dvije proizvoljne matrice veličine 2×2 i primijenite operatore iz tablica. 3. „Primjer i opis logičkih operatora koje podržava MATLAB“.

1.4. zadatak

Korištenjem naredbi grananja u MATLAB-u `for` i `if` kreirajte matricu veličine 2×100 na način da u prvom stupcu budu brojevi od 1 do 100, a u drugom stupcu budu vrijednosti 1 ili 0 ovisno o tome da li je broj paran ili ne.

1.5. zadatak

Sa stranice [1] pokrenite primjere za iscrtavanje 2D plotova i pokušajte ih obogatiti sa dodatnim opcijama. Nakon toga u komandnoj liniji pokrenite sljedeću naredbu `doc graph2d` i potrošite 7-8 minuta na čitanje dokumentacije.

1.6. zadatak

Korištenjem subplot-a iscrtajte sljedeće krivulje:

```
x=0:.1:2*pi;           % x vektor od 0 do 2*pi, dx = 0.1
plot(x,sin(x));        % plot za funkciju sinusa
plot(x,cos(x));        % plot za funkciju cosinusa
plot(x,exp(-x));       % plot negativne ekponencijalne funkcije
plot(x, x.^3);         % plot x^3
```

Za naredbe iznad potrebno je dodati subplot dio da bi se slike iscrtale na slici koja je podijeljena na 4 dijela i to 2×2 .

1.7. zadatak

Pročitajte 7-8 minuta dokumentaciju za iscrtavanje grafova u 3D prostoru (`doc graph3d`) i pokrenite min 3 primjera.

Zadaci (izazov)

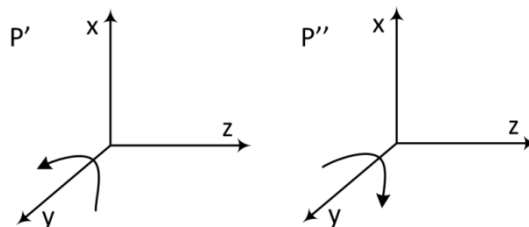
Ovi zadaci su izazov i nisu obavezni.

2.1. Zadatak

Rotirajte točku $P(2, 0, 0)$ u pozitivnom smjeru oko osi Y u 3D prostoru za $\theta = 90^\circ$ te skicirajte točku prije i poslije rotacije.

Objašnjenje: Rotacija oko osi Y u pozitivnom smjeru prikazana je matricom dok je rotacija oko osi Y u suprotnom smjeru prikazana drugom slikom P'' .

$$P' = PR_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Postupak množenja matrice 4×4 sa matricom 1×4 što je rješenje ovog zadatka dan je ispod.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} n_{00} & n_{01} & n_{02} & n_{03} \\ n_{10} & n_{11} & n_{12} & n_{13} \\ n_{20} & n_{21} & n_{22} & n_{23} \\ n_{30} & n_{31} & n_{32} & n_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} n_{00}x + n_{01}y + n_{02}z + n_{03} \\ n_{10}x + n_{11}y + n_{12}z + n_{13} \\ n_{20}x + n_{21}y + n_{22}z + n_{23} \\ n_{30}x + n_{31}y + n_{32}z + n_{33} \end{bmatrix}$$

2.2. Zadatak

Ako imamo zadan poligon sa tri točke koje predstavljaju njegove vrhove i čije su koordinate sljedeće $P_1(4, 3, 0), P_2(1, 2, 0), P_3(2, 2, 0)$ rotirajte poligon oko osi Z u pozitivnom smjeru za $\varphi = 45^\circ$ te skicirajte poligon prije i poslije rotacije. Da bi napravili rotaciju poligona potrebno je za svaku njegovu točku izračunati novu poziciju. Matrica za rotaciju oko osi Z prikazana je ispod:

$$P' = PR_z(\varphi) = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 & 0 \\ -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Skicirajte poziciju poligona prije i poslije rotacije nekim od alata u MATLAB-u.

2.3. Zadatak

Ako su zadana dva kvaterniona $q_1 = 10 + 11i + 23j + 6k$ i $q_2 = 1 + 3i + 6j + 1k$ izračunajte vrijednost kvaterniona q_3 ako izvršite sljedeću operaciju $q_3 = q_1 + q_2$. Naš zadatak je napraviti MATLAB funkciju koja radi zbrajanje kvaterniona.

Objašnjenje: Kvaternioni predstavljaju veličinu rotacije tijela u prostoru. Više o njima i njihovoj primjeni na <http://3dgep.com/?p=1815> (Jeremiah van Oosten).

Binarna operacija zbrajanja ili oduzimanja dva kvaterniona $q_1 = a_1 + b_1i + c_1j + d_1k$ i $q_2 = a_2 + b_2i + c_2j + d_2k$ koja pripadaju skupu H definira se na sljedeći način:

$$\begin{aligned}q_1 \pm q_2 &= (a_1 + b_1i + c_1j + d_1k) \pm (a_2 + b_2i + c_2j + d_2k) \\ &= (a_1 \pm a_2) + (b_1 \pm b_2)i + (c_1 \pm c_2)j + (d_1 \pm d_2)k\end{aligned}$$

2.4. Zadatak

Ako su zadana dva kvaterniona $q_1 = 10 + 11i + 23j + 6k$ i $q_2 = 1 + 3i + 6j + 1k$ izračunajte vrijednost kvaterniona q_3 ako izvršite sljedeću operaciju $q_3 = q_1 \cdot q_2$. Naš zadatak je napraviti MATLAB funkciju koja radi zbrajanje kvaterniona.

Objašnjenje: binarna operacija koja se naziva kvaternionsko množenje i označava se sa \cdot , za nju vrijedi:

$$\begin{aligned}q_1 \cdot q_2 &= (a_1 + b_1i + c_1j + d_1k) \cdot (a_2 + b_2i + c_2j + d_2k) \\ &= a_1 \cdot (a_2 + b_2i + c_2j + d_2k) + b_1i \cdot (a_2 + b_2i + c_2j + d_2k) \\ &\quad + c_1j \cdot (a_2 + b_2i + c_2j + d_2k) + d_1k \cdot (a_2 + b_2i + c_2j + d_2k) \\ &= a_1a_2 + a_1b_2i + a_1c_2j + a_1d_2k + b_1ia_2 + b_1ib_2i + b_1ic_2j + b_1id_2k \\ &\quad + c_1ja_2 + c_1jb_2i + c_1jc_2j + c_1jd_2k + d_1ka_2 + d_1kb_2i + d_1kc_2j + d_1kd_2k\end{aligned}$$

Ako na rezultat primijenimo sljedeća pravila $i^2 = j^2 = k^2 = ijk = -1$ iz čega proizlaze sljedeća pravila $ij = k$, $ji = -k$, $jk = i$, $kj = -i$, $ki = j$, $ik = -j$ koja su prikazana Cayleyovom tablicom (ispod) dobije se sljedeća formula za kvaternion q_R koji je rezultat operacije kvaternionskog množenja $q_1 \cdot q_2$:

$$\begin{aligned}q_R &= a_R + b_Ri + c_Rj + d_Rk \\ a_R &= a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2 \\ b_R &= a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2 \\ c_R &= a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2 \\ d_R &= a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2\end{aligned}$$

Tablica 1. Cayleyeva tablica

\cdot	i	j	k
i	-1	k	-j
j	-k	-1	i
k	j	-i	-1

Literatura

- [1] <http://www.mathworks.com/products/matlab/examples.html?file=/products/demos/shipping/matlab/graf2d.html>