

# Osnovi računarstva II

## Čas 11

Miloš Daković

Elektrotehnički fakultet – Podgorica

5. maj 2025.

Kroz predmet Osnovi računarstva II upoznali ste se sa algoritmima (koji su osnova programiranja), i sa softverskim okruženjima Octave, MATLAB i Maxima. Izučavanje svih mogućnosti ovih okruženja daleko prevazilazi okvire jednog predmeta. Sa druge strane navedena okruženja se stalno unapređuju i proširuju novim mogućnostima.

Na kraju nastave upoznaćemo se sa nekim naprednim Octave/MATLAB tehnikama kao što su:

- Grafički korisnički interfejs
- Rad sa fajlovima
- Napredna podešavanja grafika.

# Grafički korisnički interfejs (GUI)

- Dobar je kada programska rješenja do kojih smo došli treba proslijediti korisnicima koji se ne snalaze sa komandnom linijom i sintaksom Octave/MATLAB komandi.
- Možemo ga tumačiti kao okvir koji obavlja komunikaciju sa korisnikom, odnosno preuzimanje ulaznih podataka i prezentaciju dobijenih rezultata – ulazni i izlazni koraci u našim algoritmima.
- Sastoji se od grafičkih kontrola specifičnih za većinu modernih operativnih sistema (dugmad, polja za unos podataka, odabir opcija i sl.) Sve se to kombinuje sa naprednim mogućnostima MATLAB-a za grafičko predstavljanje rezultata.
- Osnovna funkcija je **uicontrol** koja kao argumente uzima parove „ključ“, „vrijednost“ kojima definišemo osobine i ponašanje kontrole.

# GUI – uicontrol primjeri

```
h1 = uicontrol('style','pushbutton',           ← Dugme  
              'string','Nacrtaj',             ← Tekst na dugmetu  
              'Position',[10,10,100,30],       ← Pozicija i dimenzije  
              'callback',                     ← Šta uraditi kada se klikne na dugme  
              'x=0:0.1:10;plot(x,randn*sin(x)+randn*cos(2*x))')
```

```
h2 = uicontrol('style','edit',                 ← Polje za unos teksta  
              'string','Oznaka y-ose',         ← Ponuđeni tekst  
              'Position',[120,10,100,30],      ← Pozicija i dimenzije  
              'callback',                     ← Šta uraditi kada se unos završi  
              'ylabel(get(h2,'string'))')
```

```
h3 = uicontrol('style','text',                 ← Tekst  
              'string','Primjer GUI-a',        ← Šta napisati  
              'Position',[230,10,100,30])      ← Pozicija i dimenzije
```

# Rad sa fajlovima

- Otvaranje fajla (za čitanje): `fp = fopen('abc.txt', 'r');`
- Čitanje jednog reda: `s = fgets(fp);`
- Čitanje jednog reda: `s = fgetl(fp);` U čemu je razlika?
- Provjera da li smo došli do kraja fajla: `feof(fp);`
- Zatvaranje fajla: `fclose(fp);`
  
- Otvaranje fajla (za upis): `f2 = fopen('abc2.txt', 'w');`
- Upisivanje: `fprintf(f2, 'Prva linija\n');`
- `fprintf(f2, ' N=%d, P= %g\n', N, sqrt(2));`
- `fprintf(f2, ' N=%-8d, %s\n', 15, st);`
- Zatvaranje fajla: `fclose(f2);`

# Rad sa fajlovima – primjer 1

Potrebno je matricu  $A$  upisati u tekstualni fajl `matrica.txt`, tako da se elementi svake vrste upisuju u jednoj liniji fajla odvojeni zarezima. Svi elementi matrice su prirodni brojevi sa najviše 4 cifre. Na početku fajla upisati dimenzije matrice.

```
[N,M] = size(A);  
format_string = '%4d';  
for k=2:M  
    format_string = [format_string, ', %4d'];  
end  
format_string = [format_string, '\n'];  
fp = fopen('matrica.txt','w');  
fprintf(fp,'Dimenzije matrice su %dx%d\n',N,M);  
fprintf(fp,format_string,A);  
fclose(fp)
```

← Zašto transponovanje?

# Rad sa fajlovima – primjer 1 – alternativno rješenje

Prethodni primjer se može riješiti i ovako:

```
[N,M] = size(A);  
fp = fopen('matrica.txt','w');  
fprintf(fp,'Dimenzije matrice su %dx%d\n',N,M);  
for n = 1:N                                ← petlja po vrstama  
    for m = 1:M                              ← petlja po kolonama  
        if (m == 1)                          ← prvom elementu ne treba zarez ispred  
            fprintf(fp,'%4d',A(n,m));  
        else                                  ← naredni elementi se odvajaju zapetom od prethodnog  
            fprintf(fp,', %4d',A(n,m));  
        end  
    end  
    fprintf(fp,'\n');                        ← novi red na kraju svake vrste  
end  
fclose(fp)
```

# Rad sa fajlovima – primjer 2

- Otvoriti fajl **Primjer2.m**. Za svaki karakter prebrojati koliko se puta ponavlja.
- Ispisati sortiranu listu tipa: Karakter: *c*, Broj ponavljanja: *b*.
- Ako je sadržaj fajla **Primjer2.m** ovakav:

**aaa=4411**

**B=4141414**

- Očekivani rezultat je:

**Karakter: 4 , Broj ponavljanja: 6**

**Karakter: 1 , Broj ponavljanja: 5**

**Karakter: a , Broj ponavljanja: 3**

**Karakter: = , Broj ponavljanja: 2**

**Karakter: B , Broj ponavljanja: 1**

# Rad sa fajlovima – primjer 2 – moguće rješenje

```
fp = fopen('Primjer2.m', 'r');           ← Otvaranje fajla
B = zeros(255,1);                         ← Anuliranje brojača
while ~feof(fp)                           ← Dok ne dođemo do kraja fajla
    s = fgetl(fp);                        ← čitamo liniju
    for k=1:length(s)                     ← prolazimo kroz karaktere linije
        B(abs(s(k))) = B(abs(s(k)))+1;   ← brojimo
    end
end
fclose(fp);                               ← Zatvaramo fajl
[Bs, Ind] = sort(B);                      ← Sortiramo brojače
for k=255:-1:1                            ← Idemo od najvećeg brojača naniže
    if Bs(k)==0                           ← Došli smo do karaktera koji se ne pojavljuje
        break;                           ← prekidamo ispis
    end
    fprintf('Kar.: %s  B.p.: %d\n', char(Ind(k)), Bs(k))
end
```

# Izuzeci (exceptions)

Ranije su pomenuti, kada smo govorili o naredbama za kontrolu toka programa. Možemo ih koristiti u radu sa fajlovima, na primjer: Pretpostavimo da našem programu treba informacija o vrijednosti broja  $N$  koji je zapisan u tekstualnom fajlu **nekifajl.txt**. Ukoliko ne uspije očitati  $N$  iz fajla program treba da nastavi rad podrazumijevajući da je  $N = 0$ .

```
try
    fp = fopen('nekifajl.txt', 'r');
    s = fgetl(fp);
    N = str2num(s);
catch
    warning('Problem sa čitanjem fajla. Koristim N=0')
    N=0;
end
```

# Napredna podešavanja grafika

Matlab i Octave generišu grafike visokog kvaliteta. Nekad nam je potrebno interveisati u načinu prikaza objekata na grafiku kako bi dobili zadovoljavajući rezultat. Na primjer:

- Podesiti debljinu linija na grafiku
- Promijeniti font koji se koristi na osama grafika
- Kontrolisati veličinu markera
- Postavljati poravnanje za tekstualne podatke na grafiku
- Kontrolisati numeričke vrijednosti prikazane na osama grafika ( na primjer na x-osi stoje oznake 0,2,4,6 i 8, a željeli bismo da budu prikazani svi brojevi od 0 do 8).
- ...

# Napredna podešavanja grafika

- Grafičkom objektu se pristupa preko njegovog „handle“-a.
- Handle trenutno aktivne figure dobijamo komandom **gcf**
- Handle trenutno aktivnog grafika dobijamo komandom **gca**
- Handle nacrtanih linija vraća naredba **plot** **`h = plot(x, y)`**
- Handle ispisanog teksta daje naredba **`H = text(0, 0, 'abc')`**
- Funkcije **subplot**, **axis**, **xlabel**, **ylabel**, **title** takođe vraćaju handle kreiranih objekata.
- Osobine postavljamo sa **`set(h, 'osobina', vrijednost)`** pri čemu možemo navesti više parova osobina – vrijednost.
- Osobine očitavamo sa **`get(h, 'osobina')`**
- Sve osobine objekta sa handle-om ***h*** dobijamo sa **`get(h)`**.
- Osnovni tipovi grafičkih objekata su figure, axis, line, text

# Neke osobine figura, osa, linija i teksta

- **'units'** – jedinice mjere (inči, pikseli, centimetri)
- **'position'** – vektor, 4 elementa, **[x0, y0, širina, visina]**
- **'paperposition'**, **'paperunits'** – za figure objekat
- **'children'** – handle-ovi pod-objekata
- **'color'** – boja objekta, **[1, 0, 0]** ili **'b'** ili **'blue'**
- **'xgrid'**, **'ygrid'** – grid (mreža) na osama (**'on'** ili **'off'**)
- **'linewidth'** – debljina linije (u pt)
- **'markersize'**, **'markerfacecolor'**, **'markeredgecolor'**, **'marker'** – markeri
- **'fontname'**, **'fontsize'** – font teksta
- **'horizontalalignment'** – poravnanje teksta po horizontali:  
**'left'**, **'center'**, **'right'**
- **'verticalalignment'** – poravnanje teksta po vertikali:  
**'top'**, **'middle'**, **'bottom'**
- **'rotation'** – ugao ispisa teksta (u stepenima)

# Napredna podešavanja grafika – primjer

```
x = 0:0.1:10; y1 = sin(x); y2 = cos(x)+cos(3*x);  
h = plot(x,y1,x,y2,'-o')  
set(h(1),'linewidth',4)  
set(h(2),'linewidth',2  
    ,'markerfacecolor',[1,0,0]  
    ,'markersize',3)  
xlabel('x-osa','fontsize',18)  
set(gca,'fontsize',18,'fontname','Times')  
set(gca,'xtick',0:10)  
set(gca,'ytick',[-1,0,1])  
text(pi/2,1,'maksimum'  
    ,'horizontalalignment','center'  
    ,'verticalalignment','bottom'  
    ,'fontsize',18)  
set(gca,'xgrid','on','linewidth',1.5)  
set(gca,'Position',[0.2,0.2,0.7,0.7])
```

← Nacrtajmo dvije linije  
← Debljina prve 4pt  
← Debljina druge 2pt  
← Crvena boja markera  
← Veličina markera  
← Veći font  
← Oznake na x-osi  
← Oznake na y-osi  
← Centrirano  
← Iznad date tačke  
← Grid po x-osi  
← Pozicija osa