

Osnovi računarstva II

Čas 6

Miloš Daković

Elektrotehnički fakultet – Podgorica

23. mart 2020.

- Za niz X dužine N funkcija $S = \text{cumsum}(X)$ će vratiti niz parcijalnih suma $S_n = \sum_{k=1}^n X_k$.
- Primjer: $\text{cumsum}([1, 4, 7, 2])$ vraća $[1, 5, 12, 14]$.
- Funkcija cumprod umjesto sabiranja množi odgovarajuće elemente. Na primjer $\text{cumprod}(1:10)$ vraća niz faktoriijela brojeva od 1 do 10.
- Funkcija $D = \text{diff}(X)$ vraća niz takav da je $D_n = X_{n+1} - X_n$ za $n = 1, 2, \dots, N - 1$ gdje je N dužina niza X . Uočite da dobijeni niz ima jedan element manje od polaznog.
- Primjer: $\text{diff}([2, 4, 4, 7, 2])$ vraća $[2, 0, 3, -5]$.

Numeričko računanje određenih integrala

- Iz matematike je poznato pravougaono pravilo integracije

$$\int_a^b f(x) dx \approx \sum_{k=0}^{N-1} f(a + k\Delta_x) \Delta_x, \quad \Delta_x = \frac{b-a}{N}$$

Možemo ga tumačiti ovako: Izračunajmo vrijednosti funkcije na zadanom intervalu od a do b u N ravnomjerno raspoređenih tačaka. Saberimo dobijene vrijednosti i pomnožimo ih sa korakom.

- Primjer: Nađimo $I = \int_0^{\pi} \frac{x^2 + \sin(x)}{1 + e^{-x}} dx$

$dx = \text{pi}/200;$

$x = 0:dx:\text{pi}-dx;$

$f = (x.^2 + \sin(x)) ./ (1 + \exp(-x));$

$I = \text{sum}(f) * dx$

'Koliko tačaka uzeti?'

'Zašto pi-dx?'

Numeričko računanje integrala

- Posmatrajmo integrale sa promjenljivom gornjom granicom

$$\int_a^x f(t)dt \text{ gdje je } x \text{ u nekom intervalu } a \leq x \leq b.$$

- Primjer: Nađimo $I(t) = \int_0^t \frac{x^2 + \sin(x)}{1 + e^{-x}} dx$ za $0 \leq t \leq 2\pi$.

Nacrtajmo grafik funkcije $I(t)$.

```
dt = pi/200;
```

'Koliko tačaka uzeti?'

```
t = 0:dt:2*pi-dt;
```

```
f = (t.^2+sin(t))./(1+exp(-t));
```

```
I = cumsum(f)*dt;
```

```
plot(t+dt, I)
```

'Zašto t+dt?'

Numeričko računanje izvoda

- Poznato je da se izvod neke funkcije $f(t)$ može približno izračunati kao $\frac{df(t)}{dt} \approx \frac{f(t + \Delta_t) - f(t)}{\Delta_t}$.
- Pretpostavimo da smo mjerili poziciju $x(t)$ nekog tijela u vremenskim trenucima t . Neka su rezultati mjerenja smješteni u nizovima X i T (očigledno trebaju biti jednake dužine i niz T je rastući). Odredimo i nacrtajmo brzinu tijela $v(t) = \frac{dx(t)}{dt}$.
- Smatrajući da su varijable X i T definisane u radnom prostoru zadatak možemo riješiti ovako:

```
V = diff(X) ./diff(T);
```

```
plot(t(1:end-1)+dt/2, V)
```

'Zašto ovako?'

Ukoliko su svi vremenski trenuci ravnomjerno raspoređeni sa razmakom d tada umjesto **diff(T)** možemo staviti **d**.

- Octave/MATLAB komande mogu se grupisati u jedan tekstualni fajl. Takve fajlove nazivamo skript-fajlovima ili m-fajlovima jer imaju ekstenziju **.m**
- Fajlove možemo kreirati u bilo kom tekstualnom editoru.
- Sadržaj fajlova su komande koje se izvršavaju sekvencijalno (jedna po jedna redom kojim su navedene).
- Fajl „startujemo“ navođenjem njegovog imena u komandnoj liniji Octave/MATLAB-a.
- Ime fajla mora poštovati ista pravila kao ime varijable (ne smije počinjati cifrom, sastoji se od slova, cifara i podvlake). Nije dozvoljen razmak u nazivu m-fajla.
- Neki operativni sistemi prave razliku između malih i velikih slova (linux) a drugi ne (Windows). Preporuka je da se pri imenovanju i pozivanju m-fajlova vodi računa o malim i velikim slovima.

- m-fajlovi predstavljaju Octave/MATLAB programe
- Sekvenca naredbi se jednostavno realizuje navođenjem komandi koje čine sekvencu onim redoslijedom kako želimo da budu izvršene.
- Neophodne su programske naredbe za kontrolu toka programa
- Predstavnicima tih naredbi su **if**, **for**, **while**, **break**, **error**...
- m-fajlovi rade sa aktivnim radnim prostorom, odnosno u njima možemo koristiti sve varijable koje su definisane u radnom prostoru prije startovanja m-fajla.
- Sve varijable definisane unutar m-fajla ostaju u radnom prostoru nakon njegovog izvršavanja.

Naredba if (selekcija)

- Osnovi oblik **if** naredbe je:

```
if uslov  
    niz naredbi  
end
```

- Druga varijanta je

```
if uslov  
    naredbe1  
else  
    naredbe2  
end
```

- Primjeri:

```
if x<0  
    x = x+1;  
    disp('Uvećano za 1')  
end
```

```
if x<0  
    s = -1;  
    x = -x;  
else  
    s = 1;  
end  
disp(s)
```


Uslovi u IF naredbi (i WHILE petlji)

- Osnovni operatori poređenja su `==` , `>` , `<` , `>=` , `<=` , `~=`
- Ovi operatori vraćaju binarne vrijednosti koje možemo kombinovati logičkim operacijama.
- logičko „i“ se u Octave/Matlab uslovima realizuje operatorom `&`
- logičko „ili“ se u Octave/Matlab uslovima realizuje operatorom `|`
- logičko „ne“ se u Octave/Matlab uslovima realizuje operatorom `~`
- Postoje i operatori `&&` i `||` – oni provjeravaju uslove sve dok nije sigurno šta će biti konačna vrijednost logičkog izraza. Na primjer u uslovima `1==2 && x<5` i `3==3 || x<5` se nikad neće provjeravati drugi uslov `x<5` jer je rezultat poznat bez obzira na njegov ishod.

For petlja

- Osnovi oblik **for** petlje je:

```
for k = niz_vrijednosti  
    naredbe  
end
```

- Varijabla *k* redom uzima sve vrijednosti iz zadatog niza.
- Niz vrijednosti mora biti vektor vrsta.
- Ukoliko je niz vrijednosti matrica, tada varijabla *k* uzima vrijednosti pojedinačnih kolona zadate matrice.
- Čest je slučaj: **for** *k=M:N*

- Primjeri **for** petlje:

```
s=0;  
for k = 1:N  
    s = s+k;  
end  
s
```

```
A = zeros(10);  
for n = 1:10  
    for m = 1:10  
        if rem(n,m)==2  
            A(m,n) = n+m;  
        end  
    end  
end  
disp(A)
```

While petlja

- Sintaksa while petlje je:

```
while uslov
    niz naredbi
end
```

- Naredbe unutar petlje se izvršavaju sve dok je navedeni uslov tačan.
- Treba obratiti pažnju na to da ova petlja može da se izvršava neograničeno dugo.
- Tasterskom kombinacijom **Ctrl C** prekidamo izvršenje programa prije njegovog kraja.

- Primjeri while petlje:

```
N = 3567;
while N > 10
    N = floor(N/10);
end
N
```

```
n = 24; m = 5;
while n > m
    n = n-m;
end
n
```

- Šta će biti rezultat izvršenja ovih programa?

Ostale naredbe za kontrolu toka programa

- Komanda **break** prekida izvršenje programske petlje.
- Povedite računa o tome da ova komanda prekida samo jednu petlju, onu u kojoj se nalazi.
- Komandom **error('Neki tekst')** prekidamo izvršenje programa i korisniku prikazujemo tekst greške.
- Sličnu poruku, bez prekidanja izvršenja programa, možemo poslati komandom **warning**
- Komandama **try** i **catch** možemo ograničiti dio koda kod kojeg pri izvršavanju može nastupiti greška. Na primjer

```
try  
    load mjerjenja.mat  
catch  
    error('Nema fajla mjerjenja.mat!')  
end
```

- Analizirajte m-fajl:

```
x = [1,2,3,2,3,4,5,2,3]
if x<3
    disp('???')
end
if any(x<3)
    disp('Bar jedan je manji od 3')
end
if all(x<3)
    disp('Svi su manji od 3')
end
c = (x==3)
d = sum(x==2 | x>4)
e = find(x>2)
```

Zadaci

- Napišite m-fajl koji traži kvadratni korijen iz zadatog broja x Njutnovim algoritmom. (Njutnov algoritam za određivanje kvadratnog korijena je rađen na drugom času nastave.)
- Napišite program kojim se određuje najmanje N za koje je suma

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$$

veća od 11.

- Napišite program koji za zadatu matricu A čiji su elementi cijeli brojevi, određuje koji se element najviše puta ponavlja u matrici A . (Predlažem da prije pisanja programa razmislite o algoritmu.)
- Octave i MATLAB imaju komandu `input`. Ispitajte šta ona radi, i izbjegavajte da je koristite u vašim programima. (Zašto?)