

GNU Octave

- ▶ vrlo često u tehnicima
- ▶ retko stvarni problemi imaju closed-form solution
- ▶ čak i kad imaju, pitanje upotrebljivosti
- ▶ mnogo detalja — numerički pristup
- ▶ u početku tretirano kao numerologija ...
- ▶ kolika je „saznajna vrednost“ nekog rešenja i kako se meri?

© Predrag Pejović, 

Numerical Computation, Tools #1: Matlab

- ▶ diminiira(o) Matlab
- ▶ Wikipedia, http://en.wikipedia.org/wiki/Comparison_of_numerical_analysis_software, 26.12.2020:
 - ▶ “Numerical computation and simulation with extended 2D/3D visualization. Emphasis on vectorised processing.”
 - ▶ “Proprietary”
 - ▶ 2014: “\$2150 (commercial) / \$99 (student)”
 - ▶ 2014: \$300 manje nego pre tri godine, isto kao prošle i prethodne godine
 - ▶ 2015: “\$2650 (standard) / \$625 (education)”
 - ▶ 2017: “\$2650 (standard) / \$625 (education)”
 - ▶ 2018: “\$2150 (standard) / \$500 (education) / \$149 (home) / \$49 (student)”
 - ▶ 2020: “\$2150 (standard) / \$500 (education) / \$149 (home) / \$49 (student)”
- ▶ osnovni tip podatka: n-dimenziona matrica
- ▶ extended 2D/3D visualization ...

red je da pomenemo (i ulinkujemo) i neke alternative ...

- ▶ Scilab, još jedan Matlab-like, French ...
- ▶ FreeMat (prilično neaktivan poslednjih godina)
- ▶ Python, PyLab ... prednosti modernog jezika
- ▶ R, statistika
- ▶ Julia, jako brzo, vrlo budućnost ...
- ▶ ...

okruženja

da navedem tri:

1. `octave-cli`, command line interface
2. `octave`, GUI, od verzije 4
3. `jupyter-notebook`, mora da se instalira `octave_kernel` koristeći `pip3` sa komandne linije

zbog izgrađenih navika na predavanjima uglavnom `octave-cli` pokrenemo sva tri okruženja i vidimo verzije

`ver`

Numerical Computation, Tools #2: GNU Octave

- ▶ GNU Octave
 - ▶ “General numerical computing package **with lots of extension modules**. Syntax mostly compatible with MATLAB”
 - ▶ “GPL”
 - ▶ “Free”
 - ▶ John W. Eaton
 - ▶ 1988
- ▶ <https://www.gnu.org/software/octave/>
- ▶ uglavnom sintaksa kao za Matlab, ali ima i nadgradnje i „podgradnje“
- ▶ nadgradnje: `i++`, `++i`, `i--`, `#`, do petlja, ...
- ▶ najveća razlika packages (toolboxes), mada rastu brzo!

Odakle se nabavlja GNU Octave?

- ▶ GNU/Linux, Ubuntu:
 - ▶ repository, Ubuntu Software Center
 - ▶ GNU Octave
 - ▶ Synaptic Package Manager
 - ▶ search “octave”
 - ▶ toolboxes, dodatna dokumentacija, ...
- ▶ win:
 - ▶ nije **bio** kao pod GNU/Linux
 - ▶ ima toolboxes
 - ▶ sada: [link](#)
- ▶ Da li vam je potreban GUI? Ima sada i to, by default! (na moju žalost)

literatura, dokumentacija

- ▶ dokumentacija koja ide uz GNU Octave, odlična, tu je sve što je realno potrebno
- ▶ <https://www.gnu.org/software/octave/octave.pdf>, **1121 strana!**
- ▶ tutorials na [www](#), mnogo
- ▶ izbor linkova, <http://tnt.etf.rs/~oe4sae/>
- ▶ napomena 2018: **GNU Octave 4.4.1, 1060 strana!!!**
- ▶ napomena 2020: **GNU Octave 6.1.0, 1121 strana!!!**

pocinjemo, GNU Octave kao kalkulator

```
2+2
2 + 2 # može space okolo, preporuka
2-3
2*3
3/4
3.0/4.0
10/3
-10/3
2^3
3^2
3^3
10^10 % sve su to floats!
3^64
3^640 # još nije Inf
```

Inf i NaN

```
3^6400 # Infinity, Inf
5 / Inf
5 * Inf
0 / 3
3 / 0
0 / 0 # Not a Number, NaN
4 * NaN
5 / NaN
```

promenljive, ; i ans

```
a = 3
a = 4; # odziv suppressed
a
disp(a) # uočite razliku, nema a = . . .
b = 5
a + b
ans # kod Python-a je ovo _

clear all
a

3 + ...
2
```

kompleksni brojevi

Čist run, dok niste predefinisali i i j

```
i^2
j^2
1 + j
3 + 4j
abs(3 + 4j)
arg(1 + j)
180 / pi * arg(1 + j)
```

strukture podataka

```
tacka1.x = 5
tacka1.y = 7
tacka2.x = 1
tacka2.y = -1
tacka1 + tacka2
tacka1.x + tacka2.x
# slab overloading

ima i podršku za objektno orijentisano programiranje
```

stringovi

```
a = 'Pera'
b = "Mika"
a + b # čudan string?
strcat(a, b)
a == b
a(1)
a(3)
b(1:3)
b(1 : 3)
b ( 1 : 3)
```

help!

```
help(floor)
help floor # short description
doc floor # documentation, long description
# izlazak sa q
help ceil
doc ceil
# izlazak sa q
help
doc
```

relacioni operatori

```
2 < 3
3 < 2
2 <= 2
2 >= 3
2 ~ = 2
2 != 2 % malo proširenje u odnosu na Matlab
2 == 2 % pazite se, često pravi bugs!!!
```

logički operatori

```
1 & 0
1 & 1
0 & 0
0 | 0
1 | 0
1 | 1
~1
!1
~0
!0
```

logički operatori, nad nizovima

```
a = [0 1 0 1]
b = [0, 0, 1, 1]
a & b
a | b
!a
~b
a' & b'
a' | b'
[a' !a']
[a' b' a' & b']
[a' b' a' | b']
```

operatori inkrementiranja

```
c = 0
++c
c++
c
c--
c
--c
c
```

matrice

```
a = [1, 2; 3, 4]
a = [1 2; 3 4] # ovo ne može u Python-u
a = [1 2
3 4] # ni ovo

b = inv(a)
a * b
b * a
a .* b
b .* a
a ./ a
a / a
inv(a) * a
a / b
a * a
```

matrice, indeksiranje i transpose, 1

```
a = 1 : 5
a(1) # razlika u odnosu na Python!!!
a(5)
a(2 : 4)
a(-1)
length(a)
size(a)
b = a'
b(4)
length(b)
size(b)
```

matrice, indeksiranje i transpose, 2

```
c = [1 2 3; 4 5 6]
length(c)
size(c, 1)
size(c, 2)
c(2, 2)
c(2, :)
c(:, 2)
c(2 : 4) # pazite se, bugs!!!
c
```

neke posebne matrice

```
eye(3)
zeros(4)
zeros(4, 1)
zeros(1, 4)
ones(5)
ones(1, 5)
```

scripts, Gauss na loš način

```
gedit gauss.m

View / Highlight Mode / Octave

# Gauss, na los nacín

n = input('do koliko? ');

s = 0;

for i = 1 : n
    s += i;
end

printf('sum = %d\n', s)
```

scripts, Gauss na loš način, run

```
ls
ls gauss.m
help gauss
gauss
100
```

moglo je i ovako

```
a = 1 : 100 # ili a = 1 : 100;
sum(a)
# ili
a = ones(1, 100)
b = cumsum(a)
sum(b)
mean(a)
mean(b)
```

save, prvi deo

```
clear all
a = 3
b = 5
string = 'string'
x = [1 2; 3 4]
save ws # ceo workspace!
clear all
load ws
```

a sada uradite

```
gedit ws
```

sa komandne linije, van octave, ili double click na ws

load, prvi deo

```
a
b
string
clear all
a
b
string
load ws
a
b
string
```

funkcije sa nizovima

```
deg = 0 : 360;
length(deg)
wt = deg * pi / 180;
s = sin(wt);
s2 = s .* s;
rmss = sqrt(mean(s2))
rmss - 1 / sqrt(2)
c = cos(wt);
c2 = c.^2;
rmcs = sqrt(mean(c2(1 : length(c2) - 1)))
rmcs - 1 / sqrt(2)
rmcs = sqrt(mean(c2(1 : end - 1)))
rmcs - 1 / sqrt(2)
```

plot, 2d

```
plot(deg, s)
plot(deg, c)
hold on
plot(deg, s, 'r')
close all
plot(deg, s, deg, c)
hold on
plot(deg, s2, 'm', deg, c2, 'c')
axis([0 360 -1 1]) # mora , in Python
set(gca, 'XTick', 0 : 90 : 360)
xlabel('wt [deg]')
ylabel('sin, cos, sin^2, cos^2')
title('funkcije')
```

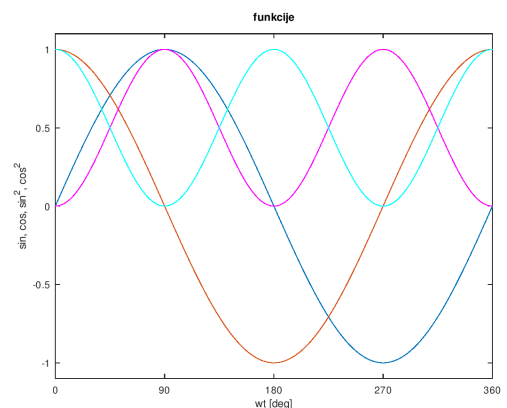
plot, 2d

```
help figure
doc figure
help print
doc print
print('slika.png', '-dpng')
print('slika.pdf', '-dpdf')
print('slika.eps', '-deps')
print('slika.jpg', '-djpg')
print('slika.svg', '-dsvg')
```

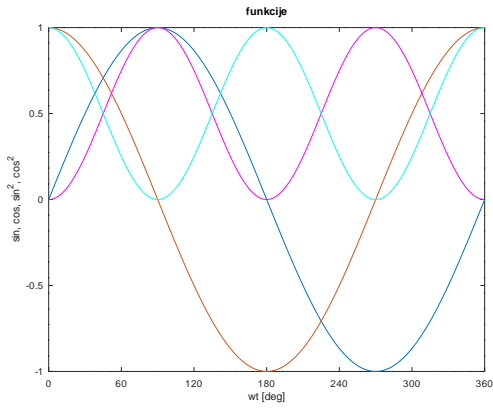
pogledajte png, pdf, eps, jpg i svg sliku, različite u odnosu na ekran, a i međusobno!

svaki drajver specifičan, kao kod gnuplot-a, što je do verzije 4.0 bio default graphics renderer, a sada optional

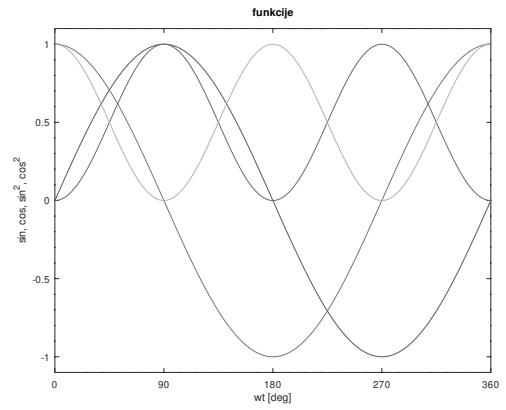
slika.png



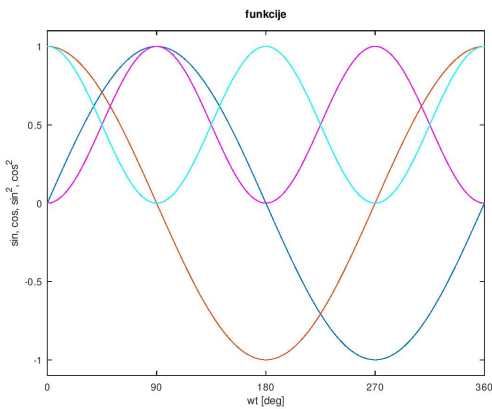
slika.pdf, cropped



slika.eps



slika.jpg



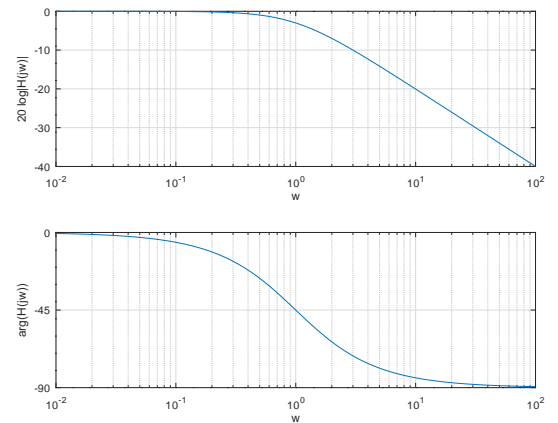
save i load, drugi deo

```
whos
data = [deg' wt' s' c' s2' c2'];
data
size(data)
save data data
clear all
data
load data
data
gedit data
```

subplots

```
# frekvencijske karakteristike, realan pol u 1
close all
w = logspace(-2, 2, 401);
s = j * w;
H = 1 ./ (1 + s);
h = 20 * log10(abs(H));
phi = 180 / pi * arg(H);
figure(1)
subplot(2, 1, 1)
semilogx(w, h)
axis([0.01, 100, -40, 0])
xlabel('w')
ylabel('20 log|H(jw)|')
grid()
subplot(2, 1, 2)
semilogx(w, phi)
axis([0.01, 100, -90, 0])
set(gca, 'YTick', [-90, -45, 0])
xlabel('w')
ylabel('arg(H(jw))')
grid()
print('rpi.pdf', '-dpdf')
```

rp.pdf, cropped



packages

ono što Matlab zove toolboxes
rastu i napreduju!
pravi trenutak je ovde!

```
pkg load control
H = tf([1], [1 1])
bode(H)
```

ima još packages!
menjaju se!
... synaptic

linearne jednačine

hoću da rešim sistem jednačina:

$$\begin{aligned}x + y &= 3 \\ x - y &= 1\end{aligned}$$

```
a = [1, 1; 1, -1]
b = [3; 1]
b = [3 1]'
a \ b # linear least squares
inv(a) * b
cond(a) # condition number
rcond(a)
rref(a) # reduced row echelon form
```

if

```
i = 1;
if i == 1
    disp(1)
endif

if i == 1
    disp(1)
else
    disp(0)
endif
```

switch

```
i = 2;
if i == 1
    disp(1)
elseif i == 2
    disp(2)
endif

switch i
case 1
    disp(1)
case 2
    disp(2)
case 3
    disp(3)
otherwise
    disp(0)
endswitch
```

for petlja

```
for i = 1 : 100
    disp(i)
endfor

for i = [1,2,3,5,3,2,6,7,9,1,10]
    disp(i)
endfor

for i = 'pera mika laza zika'
    disp(i)
endfor
```

while petlja

```
i = 0;
while (i <= 100)
    disp(i++)
endwhile
```

if-elseif-else

```
i = 2;
if i == 1
    disp(1)
elseif i == 2
    disp(2)
else
    disp(0)
endif

i = 2;
if i == 1
    disp(1)
elseif i == 2
    disp(2)
endif
```

funkcije

napravimo pdv.m, a tribute to a colleague

```
function y = pdv(x)
    #ovo je jednostavna funkcija koja računa pdv
    #
    #poziva se sa pdv(x) i vraća 1.2 * x

    y = x * 1.20
endfunction

pdv(100)
pdv(0)
pdv(200.0)
pdv([1 2 3 4 5])
pdv('wrong?')
help pdv
```

while petlja

```
i = 0;
while (i <= 100)
    disp(i++)
endwhile
```

break

```
i = 0;
while (i <= 100)
    if i > 73
        break
    endif
    disp(i++)
endwhile
```

continue

```
# los program!
i = 0;
while (i <= 100)
    i++;
    if rem(i, 2) == 0
        continue
    endif
    disp(i)
endwhile

# bolje je ovako:
i = 0;
while (i <= 100)
    i++;
    if rem(i, 2) != 0
        disp(i)
    endif
endwhile
```

do petlja

```
i = 0;
do
    i++;
    disp(i)
until i == 100

i = 0;
do
    i++;
    disp(i)
until (i == 100)
```

i još mnogo toga ...

- ▶ GNU Octave je jako bogat funkcijama!
- ▶ sjajan reference manual, 1121 strana!
- ▶ ovo uputstvo će biti dopunjavano, proširivano ...
- ▶ toolboxes, packages ...
- ▶ jezik raste, mogućnosti rastu!
- ▶ a sada? koristi se, ali Python, Julia, ...
- ▶ mora da se predaje, Matlab je uzrok
- ▶ mnogo toga ima, prioritet?
- ▶ od kada je ovaj predmet započet, 2010. godine, mnogo toga je nastalo!