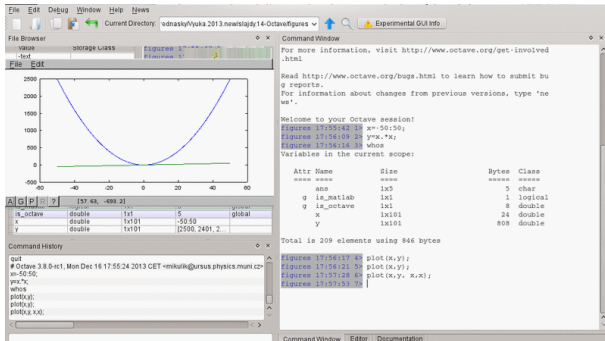
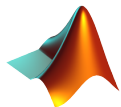


Lineární algebra v GNU Octave a Matlab

Programování F1400 + F1400a doc. RNDr. Petr Mikulík, Ph.D.

podzimní semestr 2020



Programy pro numerickou a symbolickou matematiku

- **Svobodné programy pro numerickou matematiku:** GNU Octave, SciLab, R (statistika), ...
- **Komerční programy pro numerickou matematiku:** Matlab, IDL, Mathematica, MathCAD, ...
- **Pro symbolické výpočty:** svobodná Maxima, komerční Maple
- **Matlab:** cca od roku 1983, komerční software firmy MathWorks.
Web: <http://www.mathworks.com/products/matlab/>
Platba nejen za základní program, ale též za různé balíčky (toolboxy), např. pro ODE, splajny, ...
MU: síťová multilicence
- **GNU Octave:** cca od roku 1993, svobodný software; postupná tendence k plné kompatibilitě s Matlabem (ne vždy ideální)
Web: <http://www.octave.org>
K dispozici (zdarma a legálně) pro všechny operační systémy (Unix, Linux, Mac, MS Windows, OS/2, Android, ...).
Též online webová verze, např. <http://octave-online.net> nebo http://rextester.com/1/octave_online_compiler

Spuštění GNU Octave a Matlab: ikona či příkazová řádka

- **Spuštění** ikonou programu, nebo z příkazové řádky:
 - Spuštění s GUI: `octave --gui`, bez GUI: `octave`
 - Spuštění s GUI: `matlab`, bez GUI: `matlab -nodesktop`
 - Jenom spuštění výpočtu (programu): `octave soubor.m`
- **GUI**: terminál s příkazy, okna s grafem, seznam proměnných, aj.

The screenshot displays the GNU Octave graphical user interface. At the top, there is a menu bar with 'File', 'Edit', 'Debug', 'Window', and 'Help'. Below the menu is a 'File Browser' window showing a directory structure with files like 'vauze', 'storage', 'class', and 'figures'. The main area is divided into three panes:

- Figure Window:** Shows a 2D plot with a blue parabolic curve and a green horizontal line. The x-axis ranges from -60 to 60, and the y-axis ranges from -500 to 2500.
- Command Window:** Contains the following text:

```
For more information, visit http://www.octave.org/get-involved.html
Read http://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

Welcome to your Octave session!
figures 17:59:42 1> x=-50:50;
figures 17:56:09 2> y=x.*x;
figures 17:56:16 3> whos
Variables in the current scope:

Attr Name      Size      Bytes  Class
---- ----      -
ans            1x5        5      char
g is_matlab    1x1        1      logical
g is_octave    1x1        8      double
x              1x101     24     double
y              1x101     808    double

Total is 209 elements using 846 bytes
```
- Variable List:** A table showing the current state of variables:

Variable	Type	Size	Value	Class
is_octave	double	1x1	5	global
x	double	1x101	-50:50	
y	double	1x101	[2500, 2401, 2...	

At the bottom, there is a 'Command History' window showing the sequence of commands entered:

```
quit
# Octave 3.8.0-rc1, Mon Dec 16 17:55:24 2013 CET <mikulik@ursus.physics.muni.cz>
x=-50:50;
y=x.*x;
whos
plot(x,y);
plot(x,y);
plot(x,y, x,x);
```

- **Výpočty výrazů:**

```
octave> 1+2*pi
ans = 7.2832
octave> format long; 1+2*pi
ans = 7.28318530717959
octave> format short; 1+2*pi
ans = 7.2832
octave> format rat; 1+2*pi
ans = 823/113
```

- **Formátování výstupu jako v C** – Matlab pouze `fprintf()`, Octave `fprintf()` i `printf()`

```
octave> fprintf('Eulerovo cislo %.16f, odmocnina %g\n', exp(1.0), sqrt(6.25))
Eulerovo cislo je 2.7182818284590451, odmocnina je 2.5
```

- Použití **automatické proměnné** `ans` s poslední vypočtenou hodnotou:

```
octave> 8*64
ans = 512
octave> ans*16
ans = 8192
```

- Výrazy s proměnnými, **čárka vs středník** pro potlačení výstupu:

```
octave> a=sin(pi/3), b=cos(pi/3); c=a/b-tan(pi/3)
a = 0.866025403784439
c = 0
```

- Výrazy s **komplexními čísly** – zadání pomocí znaku i nebo j za číslem:

```
octave> format short
octave> 1+2*pi + 3i + sqrt(-1)
ans = 7.2832 + 4.0000i
octave> 1+2*pi + 3j + sqrt(-1)
ans = 7.2832 + 4.0000i
octave> ans*2
ans = 14.5664 + 8.0000i

octave> exp(1j*pi), exp(1j*pi/2)
ans = -1.0000e+00 + 1.2246e-16i
ans = 6.1232e-17 + 1.0000e+00i
```

- Reálná a imaginární složka, komplexně sdružené číslo:

```
octave> a=1+2j, real(a), imag(a)
a = 1 + 2i
ans = 1
ans = 2
octave> conj(a), a'
ans = 1 - 2i
ans = 1 - 2i
```

GNU Octave jako kalkulačka pro vektory

- Zadání **řádkových vektorů** – výčtem prvků (oddělovačem je mezera nebo čárka), rozsahem s krokem nebo počtem dělení:

```
octave> a=[1 2 3 4], b=[10:0.5:11.5]
```

```
a =
```

```
1 2 3 4
```

```
b =
```

```
10.000 10.500 11.000 11.500
```

```
octave> c = linspace(1.0, 2.0, 5)
```

```
c =
```

```
1.0000 1.2500 1.5000 1.7500 2.0000
```

```
octave> d = logspace(-2,+2,5)
```

```
d =
```

```
1.0000e-02 1.0000e-01 1.0000e+00 1.0000e+01 1.0000e+02
```

- Vektorový a skalární součin, úhel mezi vektory, délka vektoru:

```
octave> cross(1:3, -3:-1:-5) % vektorovy soucin - cross product
```

```
ans =
```

```
2.0000e+00 -4.0000e+00 2.0000e+00
```

```
octave> dot(a,b) % skalarni soucin - dot product
```

```
ans = 110
```

```
octave> acos( dot(a,b) / (norm(a) * norm(b)) ) * 180/pi
```

```
ans = 21.118
```

```
octave> a=[1 2 3]; norm(a), norm(a)**2 % norma vektoru je jeho delka
```

```
ans = 3.7417
```

Řádkové vs sloupcové vektory

- Zadání **sloupcového vektoru** – výčtem s odřádkováním středníky nebo transpozicí řádkového vektoru:

```
octave> a = [1 2 3 4]      % nebo a = [1, 2, 3, 4]
```

```
a =  
    1    2    3    4
```

```
octave> a1 = [1; 2; 3; 4]
```

```
a1 =  
    1  
    2  
    3  
    4
```

```
octave> a2 = a'           % nebo a2 = [1 2 3 4]'
```

```
a2 =  
    1  
    2  
    3  
    4
```

- Poznámka: **transpozice** vektoru s komplexními čísly též provede komplexní sdružení, bez něj je třeba použít funkci `transpose()`.

Vytváření různých matic

```
octave> a = [1 2; 3 4] ... zadaná
a =          výčtem
          1  2          prvků
          3  4
```

```
octave> b = hilb(4) ... Hilbertova
b =          matice
  1.00000  0.50000  0.33333  0.25000
  0.50000  0.33333  0.25000  0.20000
  0.33333  0.25000  0.20000  0.16667
  0.25000  0.20000  0.16667  0.14286
```

```
octave> c = magic(4) ... magický
c =          čtverec
  16   2   3  13
   5  11  10   8
   9   7   6  12
   4  14  15   1
```

```
octave> c = pascal(4) ... Pascalova
c =          matice
   1   1   1   1
   1   2   3   4
   1   3   6  10
   1   4  10  20
```

```
octave> c1 = zeros(2,4), c2 = ones(1,5)
c1 =
   0   0   0   0 ... nulová matice
   0   0   0   0
c2 =
   1   1   1   1  1 ... pouze jedničky
```

```
octave> c1=0.5+zeros(2,4), c2=4*ones(1,5)
c1 =
   0.50000  0.50000  0.50000  0.50000
   0.50000  0.50000  0.50000  0.50000
c2 =
   4   4   4   4   4
```

```
octave> c3 = eye(3), c4 = rand(3)
c3 =
   1   0   0 ... jednotková matice
   0   1   0
   0   0   1
```

```
c4 =          ... náhodná matice
  0.0011806  0.7808150  0.3504223
  0.7956658  0.3569909  0.7034885
  0.8786757  0.7556389  0.7293366
```


Maticová algebra, základní maticové operace

```
octave> a, b=a', a+b
```

```
a =  
  1  2  
  3  4  
  
b =          % transponovaná matice  
  1  3  
  2  4  
  
ans =  
  2  5      % součet matic  
  5  8
```

```
octave> 2*a
```

```
  2  4  
  6  8
```

```
octave> a*b, a.*b
```

```
ans =  
  5  11      % maticový součin  
 11  25  
  
ans =  
  1  6      % součin po prvcích  
  6  16
```

```
octave> det(a), rank(a), inv(a), a*inv(a)
```

```
ans = -2      % determinant matice  
ans = 2      % hodnost (angl. rank) matice  
ans =  
 -2.00000  1.00000  % inverzní matice  
  1.50000 -0.50000  
ans =  
  1.00000  0.00000  % test výpočtu inv. mat.  
  0.00000  1.00000
```

```
octave> a ./ b, a/b, a*inv(b)
```

```
ans =  
  1.00000  0.66667  % podíl prvků matic  
  1.50000  1.00000  
ans =  
  0.00000  0.50000  % podíl matic a/b  
 -2.00000  2.50000  
ans =  
  0.00000  0.50000  % podíl matic a*inv(b)  
 -2.00000  2.50000
```

Soustavy rovnic, vlastní hodnoty a vlastní vektory

Řešení soustavy rovnic:

$$A \cdot \vec{x} = \vec{b}$$

```
octave> a=[1 2; 3 4], b=[1; 2]
```

```
a =
```

```
 1  2
```

```
 3  4
```

```
b =
```

```
 1
```

```
 2
```

```
octave> x = a \ b
```

```
ans =
```

```
 0.00000
```

```
 0.50000
```

```
octave> a*x - b
```

```
ans =
```

```
 0
```

```
 0
```

Vlastní hodnoty (eigenvalues) a vektory matice:

$$A\vec{v} = \lambda\vec{v}, (A - \lambda E)\vec{v} = 0, |A - \lambda E| = 0$$

```
octave> [u,v] = eig(a)
```

```
u =
```

```
-0.82456 -0.41597
```

```
 0.56577 -0.90938
```

```
v =
```

```
Diagonal Matrix
```

```
-0.37228 0
```

```
 0 5.37228
```

```
octave> det(a-v(1,1)*eye(2)), det(a-v(2,2)*eye(2))
```

```
ans = 0
```

```
ans = 0
```

```
octave> k=1; a*u(:,k) - v(k,k)*u(:,k)
```

```
ans =
```

```
 0
```

```
 0
```

```
octave> k=2; a*u(:,k) - v(k,k)*u(:,k)
```

```
ans =
```

```
 0
```

```
 0
```

- **Záznam dění** veškerých vstupů a výstupů na obrazovce lze automaticky kopírovat do souboru příkazem

```
diary
```

- Příkazy se zadávají na příkazové řádce interpretru, lze je editovat (šipky vlevo a vpravo, totéž s Ctrl). **Historii příkazů** lze procházet (šipky nahoru, dolů) a rychle vyvolávat (Ctrl-R), či ji celou vypsát příkazem

```
history
```

- Dlouhé výstupy na obrazovce je možné nechat **stránkovat** (vzpomeňte na rouru s programy more a less):

```
a=1:1000
more on; a
more off; a
```

- **Skriptování** – soubory se skripty (programy): příkazy napíšeme do souboru s příponou `.m`, např. `test1.m`, a ten pak z příkazové řádky zavoláme jménem souboru nebo příkazem `source()`:

```
test1
source('test1.m')
```

- **Načtení matice ze souboru** – nejjednodušší (pouze matice nebo sloupec čísel, řádky začínající znakem # nebo % se přeskakují):

```
load data.dat  
mat1 = load('data.dat')
```

- **Načtení matice ze souboru** – nejjednodušší a nejrychlejší a též i pro obrázky či jiné typy dat:

```
a=importdata('data.dat');  
obr=importdata('domecek.png');  
zvuk=importdata('test.mp3');
```

- **Zápis matice do souboru** – nejjednodušší:

```
save data.dat matA b c  
save -text data.dat matA b c  
save -binary data.mat matA b c
```

- *Poznámka: **Pokročilý způsob čtení souboru** – nastavení hlavičky souboru, oddělovače sloupců, typů údajů ve sloupcích:*

```
textread(), textscan(), nebo dlmread()
```

- *Poznámka: **Pokročilé způsoby čtení a zápisu** – syntaxe jako v Céčku:*

```
fopen(); fscanf(), fgetl(), fgets(), fread(); fclose()  
fopen(); fprintf(), fputs(), fwrite(); fclose()
```

Kreslení grafů ve 2D

- **Vykreslení grafu funkce** – nejdříve rozsah a krok (vzorkování) rozsahu, pak výpočet, poté vykreslení:

```
x = -20:0.5:20; % nebo linspace(-20,20,41)
y1 = x.*x;
plot(x,y1)
```

- Kreslení grafů **více funkcí** a **nastavení grafu** (popisky, osy, ...):

```
x = -20:0.5:20; % nebo linspace(-20,20,41)
y1 = x.*x;
y2 = 50-x.*x;
plot(x,y1, x,y2)
title('paraboly')
xlabel('x [mm]'); ylabel('výška [mm]')
grid on
legend('funkce x^2', 'funkce 50-x^2')
xlim([-10 10]); ylim([0 100])
```

- **Typy, barvy a velikosti čar a bodů** kreslených křivek:

```
plot(x,y1)
plot(x,y1, 'b--', x,y2, 'r-.')
plot(x,y1, 'g-', 'LineWidth',2, x,y2, 'r*', 'MarkerSize',3)
```

w	whitew	.	point	-	solid
m	magenta	o	circle	:	dotted†
c	cyan	x	x-mark	-.	dashdot†
r	red	+	plus	--	dashed†
g	green	*	star		
b	blue	s	square†		
y	yellow†	d	diamond†		
k	black†	v	triangle (down)†		
		^	triangle (up)†		
		<	triangle (left)†		
		>	triangle (right)†		
		p	pentagram†		
		h	hexagram†		

Vykreslování grafů

- **Více oken s grafy** současně:

```
figure; plot(sqrt(1:100)); figure; plot(-10:10)
figure(1); plot(1:100); figure(2); plot(-10:10)
```

- **Postupné př(e,i)kreslování** do jednoho obrázku:

```
clf(); hold on;
plot(sqrt(0:64)); plot(10-sqrt(1:81));
hold off
```

- **Tisk grafu do souboru** (formáty pdf, postscript, svg, png, ...):

```
print -dpdf fig.pdf
print -deps2 -tight fig1.eps
print -depssc2 -tight fig2.eps
print -dsvg fig.svg
print -dpng "-S800,600" fig.png
```

- *Technická poznámka – GNU Octave: lze zvolit vykreslování grafů skrze gnuplot, knihovnu qt nebo knihovnu fltk:*

```
available_graphics_toolkits          ... vypise se:  fltk  gnuplot  qt
plot(sqrt(0:81))
graphics_toolkit
... zavřít okno s grafikou ...
graphics_toolkit gnuplot            ... zmena zobrazovacího systemu
plot(sqrt(0:81))
```

- **Mapa** (image) – pohled na plochu shora:

```
sombbrero();  
peaks(20); colorbar;  
a=1./hilb(30);  
imagesc(a); colorbar  
  
colormap(autumn); colorbar  
colormap(1-gray); colorbar  
colormap(jet); colorbar  
help colormap      nebo      doc colormap
```

- **Drátový graf** (mesh), **reliéf** (surface), **vrstevnice** (contours):

```
a=sombbrero(); % nebo a=peaks();  
mesh(a)  
surf(a)  
surf(a); shading flat  
surf(a); shading interp  
axis([0 20 0 10])  
axis([0 20 0 10 -10 40])  
contour(a); meshc(a); surfc(a)
```

- *Poznámka: Pokročilé – generování matice souřadnic x, y pomocí `meshgrid()`, `repmat()`, `aj`.*