

# MATLAB HRAVĚ

Zdeněk Jančík, FIT VUT Brno

**MATLAB** (MATrix LABoratory) – software pro vědecké výpočty a zobrazování.

## 1 Několik praktických rad

- Windows: spusťte Matlab z adresáře Q:\MATLAB dvojitým kliknutím na jeho ikonu. Chybová hlášení ignorujte.
- Linux: spusťte Matlab z terminálového okna pomocí `matlab`. Pokud preferujete “command-line styl” bez mohutného grafického rozhraní, použijte `matlab -nodesktop`.
- v Matlabu můžete zadávat příkazy pouze, je-li kurzor na příkazové řádce `>>`
- ve vedlejší okně si otevřete v libovolném editoru (můžete použít i vestavěný editor Matlabu, který umí “obarvovat”) **textový soubor**, kam si **vše zapisujete** (příkazy, výsledky, Vaše poznámky). Příkazy můžete do Matlabu kopírovat pomocí copy (`Ctrl-c`) and paste (`Ctrl-v`) a to i několik příkazů najednou. Po ukončení cvičení Vám pak zbyde přesný záznam toho, co jste dělali, a co (snad) chodilo.
- Když budete copy-pastovat příkazy ze zadání (a to asi budete hodně), doporučujeme spíše používat zdrojové kódy v Latexu. Z PDFka se Bohu žel špatně kopírují apostrofy a ty jsou pro Matlab poměrně důležité (načítání souborů, transpozice vektorů).
- Obsahuje-li řádek znak `%`, bere se jeho část za tímto znakem jako *poznámka* a není Matlabem prováděna.
- v Matlabu je možné se k předchozím příkazům vracet pomocí `↑`. Návrat může být i “selektivní”, např. k již použitým příkazům `plot` se můžete vrátit pomocí:

`pl ↑ ↑ ↑ ...`

## 2 Velmi stručné základy Matlabu

1. **Matlab rozlišuje v názvech proměnných malá a velká písmena**, takže `mujvektor`  $\neq$  `MujVektor`  $\neq$  `MUJVEKTOR`.
2. **Pomoc !!!**
  - v Matlabu: `help jméno_funkce` a `lookfor klíčové_slovo`
  - na webu: <http://www.mathworks.com/access/helpdesk/help/techdoc/>
3. **Matlab indexuje prvky od 1!!**

### 2.1 Proměnné

1. **Typy proměnných:**
  - skalární: `a=1;`
  - sloupcový vektor: `vecsl=[1; 2; 3]`
  - řádkový vektor: `vecrad=[2.5 16e-3 85]`
  - matice: `mat=[1 2 3; 4 5 6; 7 8 9]`
2. **Zobrazení obsahu proměnné:**
  - textové: jméno proměnné bez středníku.
  - grafické s x-ovými souřadnicemi bodů 1,2,3,...: `plot(vec)`
  - grafické se zadáním x-ové i y-ové souřadnice: `plot(vecx,vecy)`
  - zjištění velikosti proměnné:  
`length(vec)` – délka vektoru  
`size(mat)` – počet řádků a sloupců matice (dvouprvkový vektor)

### Příklady

1. Naplňte skalární, vektorovou a maticovou proměnnou. Zobrazte jejich obsah jako text a graficky.
2. Jak zadáte komplexní číslo? Co je výsledkem grafického zobrazení vektoru komplexních čísel?

## 2.2 Základní operace s vektory a maticemi

- přičtení/odečtení/násobení/dělení skalárem vektoru nebo matice:  $v+s$ ,  $v-s$ ,  $v*s$ ,  $v/s$
- skalární součin dvou vektorů: `rad_v*sl_v`
- součin dvou vektorů nebo matic prvek po prvku: `rad_v.*rad_v`, `sl_v.*sl_v`, `mat.*mat`
- generování aritmetické posloupnosti:  
s krokem 1: `min_hodnota:max_hodnota`  
s volitelným krokem: `min_hodnota:krok:max_hodnota`
- transpozice vektoru/matice: `vektor'`, `matice'`
- funkce (abs, sin, cos, log, ...) pracují většinou i s vektory či s maticemi prvek po prvku.
- vytvoření matice nul či jedniček:  
`zeros(počet_řádků,počet_sloupců)`, `ones(počet_řádků,počet_sloupců)`
- přístup k prvkům vektorů a matic: `vec(1)`, `vec(2:5)`, `mat(3,4)`, `mat(3,:)`, `mat(1:5,4)`

## Příklady

1. Vytvořte matici o rozměrech  $4 \times 5$  se všemi prvky rovnými jedné. Přičtěte ke všem prvkům 3.25, pak vynásobte všechny prvky -1.5. Zobrazte třetí a čtvrtý sloupec matice.
2. Vygenerujte funkci  $\cos(x)$  v intervalu  $[0, 2\pi]$  a zobrazte se slušnou osou.

## 3 Načítání, ukládání a zobrazování dat

### 3.1 Textová data

Pro načítání textových dat v Matlabu existuje celá řada funkcí. My si vyzkoušíme dvě - `load` a `textread`.

- `load` - načítá pouze čísla oddělená mezerami
- `textread` - umožňuje načítat data v obecném formátu (řetězce, celá čísla, desetinná čísla), dále je možno definovat odělovače, komentáře apod.

```
% soubor my_data.txt:  
% 1 2 3 4 5  
% 6 7 8 9 10  
A = load('my_data.txt');
```

```
% soubor my_data.txt:  
% Sally      Level1 12.34 45 Yes  
% Tommy     Level1 15.29 18 No  
% nacteme  
[names, types, x, y, answer] = textread('my_data.txt', '%s %s %f %d %s')
```

Ukládání textových dat funguje obdobně. Nejjednodušší funkcí je `save`, která slouží k uložení matice čísel do souboru.

```
save -ASCII data.txt A;
```

## Příklady

1. Načtěte soubor s kurzem koruny k Euru `kurz.dat`, kde v prvních třech sloupcích je datum (23 9 2008) a v dalších sloupcích je cena při nákupu a prodeji. Zobrazte průběh obou cen do jednoho grafu.
2. Vygenerujte gaussovku  $\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$  na intervalu  $[-5, 5]$  s parametry  $\mu = 0$  a  $\sigma = 1$ , výsledek zobrazte a uložte do textového souboru.

## 3.2 Binární data

Přístup k binárním datům je podobný jako v jazyce C. Pro otevření souboru se používá `fopen`, pro čtení `fread` a zápis `fwrite`.

```
ff = fopen('test.raw','r');
s = fread(ff, [1 inf], 'short');
plot(s);
fclose(ff);
```

```
fid = fopen('out.bin','wb');
y = sin(0:pi/100:2*pi);
fwrite(fid,y,'float');
fclose(fid);
```

## Příklady

1. Gaussovku, kterou jste si vygenerovali v minulém příkladu uložte do binárního souboru. Zvolte jiné parametry ( $\mu$  a  $\sigma$ ). Soubor načtete a zobrazte.

## 3.3 Zvuk

Matlab umí načítat jednokanálové zvukové soubory ve formátu wav pomocí funkce `audioread`, zápis obstarává analogicky funkce `wavwrite`. Zvuk můžete přehrát přímo v Matlabu pomocí `soundsc`.

```
[Y,Fs] = audioread('test.wav');
% neco udelame
audiowrite('out.wav',Y,Fs,'BitsPerSample',16);
soundsc(Y,Fs);
```

## Příklady

1. Načtete signál “létající prase” a zkuste si jej o 0,5 s zpozdít.
2. Na signál “létající prase” aplikujte efekt echo (nápopověda: sečtete původní signál se zeslabeným posunutým signálem).

## 3.4 Obraz

Matlab umí pracovat se všemi základními obrazovými formáty (JPG, TIFF, PNG, GIF). Pro načítání se používá funkce `imread`, pro zápis `imwrite`. Obrazová data zobrazujeme pomocí funkce `imagesc`.

```
img = imread('lena_cb.tif');
imagesc(img); colorbar;
% neco udelame
imwrite(img,'out.png');
```

## Příklady

1. Načtete obrázek a zobrazte jej (nápopověda: správného zobrazení barev dosáhnete pomocí funkce `colormap(gray)`).
2. Obrázek zesvětlete a ztmavte. Po každé úpravě výsledek zobrazte.

## 4 Další využití Matlabu

Matlab umí vytisknout veškerou grafiku do souboru (nebo na tiskárně). Tisk do souboru je zajímavější, protože výsledky můžeme vkládat do článků (slajdů, diplomky, apod.) různé grafy a obrázky z Matlabu. Základem je funkce `print`.

```
x = 0:pi/100:2*pi;
y = cos(x);
plot(x,y);
print -dpng cos.png;
```

Pro uživatele systému  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  je jistě zajímavé, že Matlab umí grafiku vytisknout i do vektorového formátu eps.

```
print -deps cos-bw.eps % cernobíle EPS
print -depsc cos-c.eps % barevné EPS
```

Popisky k datům se tvoří funkcí `legend`, k osám pomocí `xlabel`, `ylabel` a `zlabel` a k celému grafu pomocí `title`.

```
plot(x,y); legend('cos(x)'); xlabel('x'); ylabel('y'); title('kosinusovka');
```

Více os  $y$  dostaneme, pokud místo `plot` použijeme `plotyy`.

```
plotyy(x,y1,x,y2);
```

Pro zobrazení dat s velkým dynamickým rozsahem se používají logaritmické osy. Graf s logaritmickými osami se potom vykresluje pomocí funkce `loglog`. Funkce `logspace(x,y)` generuje vektor s logaritmickým rozložením hodnot od  $10^x$  do  $10^y$ .

```
x = logspace(-1,2);  
loglog(x,exp(x)); grid on;
```

Často také potřebujeme do jednoho obrázku vygenerovat větší množství grafů. K tomu skouží funkce `subplot(m,n,p)`, kde  $m$  je počet grafů na výšku,  $n$  na šířku a  $p$  je pořadí grafu.

```
x = 0:pi/100:2*pi;  
y1 = sin(x); y2 = cos(x);  
subplot(211); plot(x,y1);  
subplot(212); plot(x,y2);
```

Naopak někdy chceme generovat každý obrázek do zvláštního okna. To otevřeme pomocí funkce `figure`.

```
x = 0:pi/100:2*pi;  
y1 = sin(x); y2 = cos(x);  
figure(1); plot(x,y1);  
figure(2); plot(x,y2);
```

... a to zdaleka není vše. Další zajímavé funkce si zájemci jistě najdou v nápovědě.

## Příklady

1. Zobrazte legendu ke grafu někde jinde, než vpravo nahoře.
2. Zobrazte obrázek s osami v milimetrech (předpokládejte hodnotu  $dpi = 100$ ) a vhodnými popisky.
3. Zobrazte zvukovou nahrávku z osami v sekundách a vhodnými popisky.

## Řešení příkladů

- Velmi stručné základy Matlabu

1. `A=ones(5,4); A=A+3.25; A=A.*(-1.5); A(:,3:4)`
2. `x=0:pi/100:2*pi; plot(x,cos(x));`

- Načítání a ukládání textových dat

1. `load kurz.dat; s=size(kurz); t=1:s(1,1); plot(t,kurz(:,4),t,kurz(:,5))`
2. `mu=0; sigma=1; x=-5:0.01:5;  
y=1/(sigma*sqrt(2*pi))*exp(-(x-mu).^2)/(2*sigma^2);  
plot(x,y); save -ASCII gauss.txt y;`

- Načítání a ukládání binárních dat

1. `fid = fopen('gauss.bin','wb'); mu=1; sigma=0.5; x=-5:0.01:5;  
y=1/(sigma*sqrt(2*pi))*exp(-(x-mu).^2)/(2*sigma^2);  
fwrite(fid,y,'float'); fclose(fid);  
ff = fopen('gauss.bin','r'); s = fread(ff, [1 inf], 'float');  
plot(s);fclose(ff);`

- Načítání a ukládání zvukových dat

1. `[Y,Fs] = audioread('test.wav');  
z=zeros(length(Y)+Fs/2,1);  
z(Fs/2+1:Fs/2+length(Y)) = Y;`
2. `[Y,Fs] = audioread('test.wav');  
z1=zeros(length(Y)+Fs/2,1); z2=zeros(length(Y)+Fs/2,1);  
z1(1:length(Y)) = Y; z2(Fs/2+1:Fs/2+length(Y)) = Y;  
ech = z1 + 0.5*z2;  
soundsc(ech,Fs);`

- Načítání a ukládání obrazových dat

```
1. img = imread('lena_cb.tif'); imagesc(img);  
   colormap(gray); imagesc(img);  
   imagesc(img*1.5); colormap(gray);  
   imagesc(img*0.1); colormap(gray);
```

- Další využití Matlabu

```
1. x=0:pi/100:2*pi; y=cos(x); plot(x,y); legend('cos(x)', 'Location', 'EastOutside');  
2. img = imread('lena_cb.tif');  
   dpi=100; s=size(img); x=1:50:s(1,1); x=x/dpi*25.4;  
   imagesc(img); axis image; colormap(gray); title('Lena');  
   set(gca, 'XTickLabel', x); set(gca, 'YTickLabel', x);  
3. [Y, Fs] = audioread('test.wav'); t=1:length(Y); t=t/Fs;  
   plot(t, Y); title('létající prase'); xlabel('s');
```