

# MATLAB I

Ing. Bohumil Kovář

5. března 2002

## 1 Úvod

MATLAB je programový systém společnosti The MathWorks, Inc. Název souboru je zkratka "MATrix LABoratory" a výstižně charakterizuje způsob práce s programem. Původně MATLAB sloužil jako interface na metody numerických knihoven LINPACK a EISPACK. Později se stal komerčním produktem a pro svou jednoduchost se stal průmyslovým standardem v oblasti softwarových řešení automatického řízení (CADCS).

### 1.1 Přednosti MATLABu

Tím, že se MATLAB stal světovým standardem v mnoha inženýrských oborech, je rozšířen po celém světě a existuje pro všechny operační systémy (Unix, Linux, Solaris, Windows, Mac OS). MATLAB je možno relativně pohodlně rozšiřovat o další funkce a na Internetu je možné získat několik desítek toolboxů. S programem je možné začít pracovat i bez speciálních znalostí programování (nicméně, elementární znalosti programování jsou výhodou). V systému MATLAB jsou zabudovány robustní numerické metody. MATLAB je možné dále rozšiřovat o externí programy (MEX-soubory) naprogramované v C/C++ nebo Fortranu.

## 2 Příkazy a proměnné

Příkazy jsou v MATLABu ve tvaru *proměnná = výraz*. Výrazy se skládají z operátorů, speciálních znaků, funkcí a proměnných. Pokud chybí přiřazení proměnné, zavede se systémová proměnná *ans*. Umístěním středníku za výrazem potlačíme výstup na obrazovku. Mezery uvnitř výrazu jsou nepodstatné. Výsledkem je obecně matice, která se zobrazí na obrazovce. Jména proměnných a funkcí musí začínat písmenem. MATLAB **rozlišuje malá**

**a velká písmena** ve jménech proměnných, funkcí a konstant.

### Příklad:

```
>> 1320 / 63
>> ans =
      20.9524
>> a = 1 + 1;
>> a = a + 1
>> a =
      3
```

## 3 Čísla a aritmetické výrazy

MATLAB provádí všechny výpočty v dvojité přesnosti (pokud není definováno jinak). Priorita operací je stejná jako v matematice. Okolo symbolů *e*, *E* nesmí být mezera (např. 3.14159, 6.6345e23, 9.12E-20). V MATLABu je možné používat následující operátory:

- + sčítání,
- odčítání,
- \* násobení,
- / pravé dělení,
- \ levé dělení,
- ^ umocňování
- () závorky upravují pořadí provádění operací.

## 4 Manipulace s maticemi a vektory

### 4.1 Generování vektoru

Pro generování vektoru se používá notace s dvojtečkou.

#### Příklad:

```
>> x = 1:5
>> x =
     1     2     3     4     5
>> y = 0:pi/4:pi
>> y =
    0.0000    0.7854    1.5708    2.3562    3.1416
```

## 4.2 Generování tabulky

#### Příklad:

```
>> x = [0.0:0.1:0.5]';
>> y = exp(-x).*cos(x);
>> [x y]
>> ans =
         0    1.0000
    0.1000    0.9003
    0.2000    0.8024
    0.3000    0.7077
    0.4000    0.6174
    0.5000    0.5323
```

## 4.3 Generování matice

1. Odkaz na konkrétní prvek matice

```
>> a(3,3) = a(1,3) + a(3,1);
```

2. Specifikace matice (5,1), která se skládá z prvků A(1,3), A(2,3), ..., A(5,3)

```
>> A(1:5,3);
```

3. Specifikace matice (5,4), která se skládá z prvních pěti řad a sloupců 7 až 10 matice A

```
>> A(1:5,7:10);
```

4. Všechny prvky 3. sloupce matice A

```
>> A(:,3);
```

5. Nahrazení 2., 4. a 7. sloupce matice A prvními třemi sloupci matice B

```
>> A = [1 2 3 4 5 6 7 8 9; ...
        1 2 3 4 5 6 7 8 9; ...
        1 2 3 4 5 6 7 8 9];
>> B = [0 0 0 13 13; ...
        0 0 0 13 13; ...
        0 0 0 13 13];
>> A(:, [2 4 7]) = B(:, 1:3)
```

```
>> A =
```

```
     1     0     3     0     5     6     0     8     9
     1     0     3     0     5     6     0     8     9
     1     0     3     0     5     6     0     8     9
```

6. Prohození sloupců matice A

```
>> A = A(:,9:-1:1);
>> A =
     9     8     0     6     5     0     3     0     1
     9     8     0     6     5     0     3     0     1
     9     8     0     6     5     0     3     0     1
```

7. Převod matice na vektor

```
>> A = [1 2; 3 4; 5 6];
>> B = A(:)';
>> B =
     1     3     5     2     4     6
```

8. Prázdná matice. Prázdná matice  $x=[]$  má nulový rozměr. Vymazání 1. a 3. sloupce matice A se tedy provede

```
>> A(:, [1 3]) = [];
```

## 4.4 Komplexní čísla a matice

Komplexní čísla jsou povolena ve všech operacích a funkcích MATLABu. Pokud  $i$  nebo  $j$  používáme jako proměnné, je třeba nadefinovat jinou imaginární jednotku, např.

```
>> ii = sqrt(-1);
>> z = 2 + 3*ii;
```

Zápis matice s komplexními prvky se tedy provádí:

```
>> A = [1 2; 3 4] + i*[-5 6; 7 -8];
>> A = [1-5*i 2+6*i; 3+7*i 4-8*i];
```

## 4.5 Operace s maticemi

Zápis matic v MATLABu je obdobný jako v matematice. Základní operace s maticemi jsou:

1. Transpozice  $B = A'$

```
>> A = [1 2; 3 4]
>> A =
     1     2
     3     4
>> B = A'
>> B =
     1     3
     2     4
```

2. Součet a rozdíl. Matice musí mít stejný rozměr  
 $(m, n) + (m, n) = (m, n)$

```
>> C = A + B;  
>> C = A - B;
```

3. Součin. Matice musí mít příslušný rozměr  
 $(m, n) * (n, k) = (m, k)$

```
>> C = A * B;
```

4. Podíl. V MATLABu rozlišujeme pravé dělení  $C = A/B$ , které je definováno  $A * \text{inv}(B)$  a levé dělení  $C = A \backslash B$ , definované  $\text{inv}(A) * B$ . Pravé dělení je také možné definovat jako  $B/A = (A' \backslash B')'$ .

5. Umocňování  $C = A^p$  resp.  $C = p^A$  je definováno, pokud  $A$  je čtvercová matice a  $p$  skalár.

## 5 Operace s poli

Tyto operace odpovídají operacím "prvku s prvkem". Operace jsou stejné jako s maticemi. Jedná se o tyto operátory: `.' .* .\ ./ .^`.

**Příklad:**

```
>> x = [1 2 3], y = [4 5 6];  
>> z = x .\ y  
>> z =  
    4.0000  2.5000  2.0000  
>> z = x .^ y  
>> z =  
    1    32   729
```

## 6 Relace

Relace je možné provádět mezi maticemi stejných rozměrů. Porovnání je prováděno mezi odpovídajícími prvky. Výsledkem je matice samých 0 a 1. Relační operátory jsou:

`<` menší než,  
`<=` menší nebo rovno,  
`>` větší než,  
`>=` větší nebo rovno,  
`==` rovno,  
`~=` nerovno,

**Příklad:**

```
>> x = [1 4 8], y = [-4 5 0];  
>> x < y  
>> ans =  
    0    1    0
```

## 7 Logické operace

Logické operace lze provádět mezi maticemi stejných rozměrů. Logické operace jsou prováděny mezi prvky matic, které jsou obvykle 0 nebo 1. Logické operátory jsou:

`&` konjunkce (AND),  
`|` disjunkce (OR),  
`~` negace (NOT),

V MATLABu jsou zabudovány ještě další relační a logické funkce, např. `any`, `all`.

## 8 Řetězce

Řetězce jsou uzavřeny v apostrofech a ukládány jako vektory.

**Příklad:**

```
>> x = 'modelovani'  
>> x =  
    modelovani  
>> x = [x, 'systemu a procesu']  
>> x =  
    modelovani systemu a procesu  
>> size(x)  
>> ans =  
    1   27
```

## 9 Informace o proměnných

Informace o proměnných a alokované paměti udávají funkce `who` a `whos`. Alokované proměnné je možné uložit do souboru a později načíst k dalšímu zpracování pomocí funkcí `save` a `load`. Vymazání proměnné provádí příkaz `clear`.

### Příklad:

```
>> x = [1 2 3; 4 5 6; 7 8 9]
>> x =
     1     2     3
     4     5     6
     7     8     9
>> save moje_matice.mat x;
>> clear;
>> load moje_matice.mat;
>> x
>> x =
     1     2     3
     4     5     6
     7     8     9
```

### Ostatní funkce

<code>clc</code>	vymaže obrazovku
<code>echo</code>	zobrazí komentář
<code>plot</code>	grafický výstup
<code>ones(m,n)</code>	jednotková matice (m,n)
<code>zeros(m,n)</code>	nulová matice (m,n)
<code>rand(m,n)</code>	matice (m,n) náhodných čísel
<code>roots</code>	výpočet vlastních čísel

## 10 Help

MATLAB obsahuje velmi kvalitní nápovědu v HTML a PDF. Často je mnohem efektivnější použít zabudovaný help, který poskytuje dostatek informací o všech objektech MATLABu.

### Příklad:

```
>> help plot
```

## 11 Elementární funkce

MATLAB má v současné době několik stovek interních funkcí. Mezi ty nejdůležitější (a na cvičeních nepoužívanější) patří:

### Obecné funkce

<code>help</code>	on-line nápověda
<code>who</code>	seznam proměnných
<code>size</code>	dimenze matice
<code>length</code>	délka vektoru
<code>^C</code>	ukončení výpočtu
<code>chdir</code>	změna adresáře
<code>save</code>	uložení proměnné do souboru
<code>load</code>	načtení proměnné ze souboru
<code>clear</code>	de-alokace proměnných

### Matematické funkce

Goniometrické a ostatní matematické funkce jsou definovány stejně jako v matematice (např. `sin`, `cos`, `sqrt`, `exp`).