

- programový systém pro vědecko - technické a inženýrské výpočty, snadná vizualizace dat (vytváření grafů);
- jeho název - zkratka z anglických slov **MAT**rix **LAB**oratory (maticová laboratoř), vyvinut firmou MathWorks Inc. (viz www.mathworks.com);
- je **maticově orientován**, tj. pracuje obecně s obdélníkovými komplexními maticemi, což umožňuje pracovat i s vektory (1-řádková či 1-sloupcová matice) a čísla (matice typu (1/1)), i s reálnými čísly ($R \subset C$);
- nabízí velké množství standardních funkcí, i pro řešení různých úloh numerické matematiky a další toolboxy (balíky) funkcí, zaměřených na řešení specializovaných úloh numerické matematiky;
- je velmi jednoduchý na ovládání
interaktivní způsob práce: příkaz zadaný po odezvě (>>) se ihned vykoná, k vyřešení dané úlohy zadáme posloupnost příkazů, které se hned realizují, až dojdeme k požadovanému výsledku.
- součástí MATLABu je jednoduchý programovací jazyk – možno vytvářet tzv. M – soubory:
 - 1) Posloupnost příkazů vedoucí k vyřešení dané úlohy lze zapsat do souboru (**příkazový M-soubor** = program), kdykoliv potřebujeme řešit danou úlohu, tento soubor spustíme uvedením jeho jména na příkazovém řádku a posloupnost zapsaných příkazů se automaticky realizuje.
 - 2) Možno vytvořit (definovat) další uživatelské funkce, rozšiřující množinu standardních funkcí (**funkční M-soubory**).

Využití MATLABu pro nás: k řešení úloh z lineární algebry (vektory, matice, soustavy lineárních rovnic), grafy funkcí.

Spuštění MATLABu: Dvojklik na ikonu zástupce MATLABu (MATLAB verze 6, jen anglická verze)

Ukončení práce:

- a) zavřením okna MATLABu
- b) příkaz Menu: *File/Exit Matlab*
- c) zadáním příkazu `exit` nebo `quit` (po odezvě >>)

Základní informace

Okna MATLABu:	Command Window Možno zobrazit ještě okno Workspace Command History Current Directory	(Příkazové okno, zde zadáváme příkazy, vypisují se výsledky (viz příkaz menu <i>View</i>): (Pracovní prostor – přehled platných proměnných) (zde záznam zadaných příkazů) (aktuální adresář)
----------------------	---	--

DOS commands	<u>V MATLABu jsou platné tyto příkazy DOSu:</u> <code>cd</code> – nastavení aktuálního adresáře (nebo v okně Current Directory) <code>dir</code> – zobrazí obsah aktuálního adresáře (nebo v okně Current Directory) <code>type</code> – vypíše obsah udaného textového souboru <code>delete</code> – zruší udaný soubor (možno vybrat a odstranit i v okně Current Directory)	
---------------------	--	--

Nápověda (v angličtině)	<code>>>help</code> <code>>>help téma</code> <code>>>help elfun</code> <code>>>help jméno funkce</code> <code>>>help sin</code>	vypíše seznam témat (help topics) vypíše nápovědu ke zvolenému tématu (skupinu funkcí) vypíše nápovědu k elementárním matematickým funkcím (jejich seznam) vypíše nápovědu k dané funkci
-----------------------------------	---	---

Zadávání příkazů

`>> proměnná = výraz` proměnné zvoleného jména je přiřazena hodnota výrazu, vypíše se její hodnota:

`proměnná =`
`<výsledek>`

`>> výraz` hodnota výrazu je přiřazena do standardní proměnné *ans* (answer = odpověď), vypíše se její hodnota:

`ans =`
`<výsledek>`

i Pozn.:
Znak “=” v Matlabu znamená přiřazení.

Pravidla pro výběr jmen proměnných

Jména proměnných mohou obsahovat písmena a-z, A-Z, číslice 0-9 a znak podtržítka “_“, ale prvním znakem jména proměnné musí být písmeno. Rozlišují se malá a velká písmena (a, A – dvě různé proměnné; case sensitive). Maximální délka proměnné je 31 znaků.

Postupně vytvářené proměnné jsou uloženy v paměti počítače, v tzv. pracovním prostoru (Workspace), jsou v platnosti do ukončení práce v Matlabu a je možno je zrušit (vymazat) příkazem:

```
>> clear           maže všechny proměnné v pracovní paměti
>> clear a b u     maže jen uvedené proměnné
```

Na jednom řádku možno více příkazů, oddělených čárkou nebo středníkem.

Středník za příkazem potlačuje výpis výsledku.

Znak “%“ uvozuje komentář (doprovodný text), klávesa ↵ ho ukončuje:

```
>> % Výpočet objemu koule
>> r=4; %poloměr koule
>> V=4/3*pi*r^3
```

```
V =
    113.0973
```

Aritmetické operátory (v pořadí podle priority):

maticové

+ součet
 - rozdíl
 * maticový součin
 / \ dělení zprava, zleva (alt+Q) maticově
 A\B = inv(A)*B
 B/A = B*inv(A)
 ^ mocnina matice (alt+š)

() předeepisují pořadí prováděných operací

s jednotlivými prvky matic

+ součet
 - rozdíl
 .* součin jednotlivých prvků matic
 ./ podíl jednotlivých prvků matic
 .^ umocnění jednotlivých prvků matic

Př.: možno zadat:

u * v maticový součin dvou vektorů (nutno u-řádkový, v-sloupcový, oba stejný počet složek)
 3 * u každý prvek vektoru u vynásoben třemi
 u .* v u, v stejného typu, složky výsled. vektoru rovny součinům odpovídajících složek u, v
 A*B součin matic A, B (musí být vhodného typu)
 3*A každý prvek matice A vynásoben třemi
 A .* B A, B stejného typu, prvky výsledné matice rovny součinům odpovídajících prvků A, B

Logické operátory

< menší
 > větší
 < = menší rovno
 > = větší rovno
 = = rovno
 ~ = nerovno

Zápis čísel – povolené tvary:

např.: -21 -43.25 0.001
 3.5+4i 1j 3e5i

exponentový formát: 6.025e23 (=6.025·10²³)
 -1.602E-15 (= -1.602·10⁻¹⁵)

Standardní proměnné:

eps eps = 2⁻⁵² = 2.22·10⁻¹⁶ ...vnitřní přesnost Matlabu
 pi pi = 3.14159...
 Inf inf = 1/0 - nekonečno
 i (j) imaginární jednotka (0+1i)
 realmax realmax = 2¹⁰²⁴ ~1.7977·10³⁰⁸ ...největší kladné číslo
 realmin realmin = 2⁻¹⁰²² ~ 2.2251·10⁻³⁰⁸ ...nejmenší kladné číslo

Každé číslo je uloženo v paměti na 8 bytech s přesností přibl. na 16 platných dekadických cifer.

Výstupní zobrazovací formát čísla je možno zvolit – příkaz *format (help format)*.

Vytváření vektorů a matic

Vektory

a) prvky zadáváme do hranatých závorek, oddělovač mezi prvky: mezera, čárka nebo středník (ukončuje řádek, vytvoří se sloupcový vektor)

```
>> u=[3 5 -7], v=[2.5,4,-5.2,10], w=[4;8;3;9] % ...sloupcový vektor
```

b) pomocí dvojtečkové notace – není nutno do hranatých závorek:

<počáteční hodnota>:<krok>:<koncová hodnota>

<počáteční hodnota>:<koncová hodnota>

implicitní krok=1

```
>> u = [2:2:10], v=[1:-2:-6], w=1:4
```

```
u =
    2    4    6    8   10
```

```
v =
    1   -1   -3   -5
```

```
w =
    1    2    3    4
```

možno i:

```
>> t = -pi : pi/6 : 2*pi
```

př.:

$a_1 : d : a_n$ – aritmetická posloupnost

```
>> wt = w'    transponování vektoru
```

```
wt =
    1
    2
    3
    4
```

```
>>u(2)        vypíše hodnotu druhé složky vektoru u
```

```
ans =
    4
```

Matice

a) **Zadáním prvků matice**

Prvky zadáváme do hranatých závorek, oddělovač mezi prvky daného řádku: mezera nebo čárka, oddělovač řádků: středník, případně klávesa ENTER.

Počet prvků musí být v každém řádku matice stejný.

Př.:

```
>> A = [1 3 7; 2 0 9; 5 1 -2]
```

```
A =
    1    3    7
    2    0    9
    5    1   -2
```

možno využít dvojtečkové notace:

```
>> B = [1:2:7, 10; -10:2:-2; 2 9 -1 : 4 : 8]
```

```
B =
    1    3    5    7   10
   -10   -8   -6   -4   -2
    2    9   -1    3    7
```

```
>> B(2,3)    vypíše hodnotu prvku z druhého řádku a třetího sloupce matice B
```

```
ans =
   -6
```

b) **Řídkou matici zadáme zadáním jen jejích nenulových prvků:**

```
>> C(1,4) = 1; C(2,1) = 5; C(3,2) = -2; C(5,4) = 10
```

```
C =
    0     0     0     1
    5     0     0     0
    0    -2     0     0
    0     0     0     0
    0     0     0    10
```

c) **Zadání matice extrakcí prvků z větší matice nebo skládáním z menších celků:**

Dána

```
B =
    1     3     5     7    10
   -10    -8    -6    -4    -2
    2     9    -1     3     7
```

```
>> D = B([1,3], 2:4)           matice D vytvořena z prvků 1. a 3. řádku a 2. až 4. sloupce matice B
```

```
D =
     3     5     7
     9    -1     3
```

```
>> F = B( : , [1:3, 5] )      matice F vytvořena ze všech řádků matice B, sloupců 1 až 3 a 5
```

```
F =
    1     3     5    10
   -10    -8    -6    -2
    2     9    -1     7
```

```
r1 = B(1, : )                 první řádek matice B
```

```
s2 = B( : , 2)                druhý sloupec matice B
```

Skládáním:

```
>> M = [D; [4 7 0]]          k matici D přidán další řádek
```

```
M =
     3     5     7
     9    -1     3
     4     7     0
```

```
>> N = [M F(:,1)]           k matici M přidán jako další sloupec první sloupec matice F
```

```
N =
     3     5     7     1
     9    -1     3    -10
     4     7     0     2
```

Elementární matematické funkce (nejpoužívanější) – viz help elfun

sin, cos, tan	goniometrické funkce
asin, acos, atan	cyklometrické funkce
exp	exponenciální funkce (e^x)
log, log10	přirozený, dekadický logaritmus
abs	absolutní hodnota
sqrt	druhá odmocnina
round	zaokrouhluje na nejbližší celé číslo
fix	“ směrem k nule (celá část argumentu)
floor	směrem k minus nekonečnu (nejbližší celé menší)
ceil	směrem k plus nekonečnu (nejbližší celé větší)

- jeli argument funkce matice (vektor), realizuje se funkce s každým prvkem matice (vektoru).

Př.: Tabelece funkce $y=x \cdot \sin(x)$:

```
>> x = [0 : 0.2 : 2]';
>> y = x .* sin(x);
>> [x,y]
```

```
ans =
      0      0
  0.2000  0.0397
  0.4000  0.1558
  0.6000  0.3388
  0.8000  0.5739
  1.0000  0.8415
  1.2000  1.1184
  1.4000  1.3796
  1.6000  1.5993
  1.8000  1.7529
  2.0000  1.8186
```

Funkce pro tvorbu elementárních matic (help elmat)

zeros vytvoří matici s nulovými prvky:

<code>zeros(N)</code>	matici typu (N/N) s nulovými prvky
<code>zeros(M,N)</code>	matici typu (M/N) s nulovými prvky
<code>zeros(size(A))</code>	matici stejného rozměru jako A s nulovými prvky

eye vytvoří jednotkovou matici

ones vytvoří matici s prvky rovnými jedné

rand vytvoří matici náhodných čísel z intervalu < 0;1). (Např `A=100*rand(3,4)` - prvky A z < 0;100)

magic magický čtverec

diag diagonální matice, diagonály matic

Manipulace s maticemi (help elmat)

fliplr překlopí matici ve směru zleva-doprava

flipud překlopí matici ve směru shora-dolů

rot90 otočí matici o 90 stupňů

size rozměr matice (počet řádků a sloupců)

length délku vektoru (počet složek)

Maticové funkce (help matfun)

det výpočet determinantu matice

inv určení inverzní matice

trace určení stopy matice (součet prvků hlavní diagonály)

rank určení hodnosti matice

rref odevzdá matici po Gaussově eliminaci

Funkce pro matice i vektory

sum(X) pro X – vektor: součet jeho složek
pro X – matice: řádkový vektor součtů prvků jednotlivých sloupců

max(X) pro X – vektor: největší složku X
pro X – matice: řádkový vektor největších prvků v jednotlivých sloupcích

max(X,Y) pole stejného rozměru jako X, Y, každý jeho prvek je roven většímu z odpovídajících prvků v X, Y.

min(X) analogická s funkcí max

sort(X) pro X – vektor: setřídí prvky vektoru vzestupně
pro X – matice: setřídí prvky v jednotlivých sloupcích vzestupně

product(X) pro X – vektor: součin jeho složek
pro X – matice: řádkový vektor součinů prvků jednotlivých sloupců

Transponování matice (vektoru)

```
>> XT = X'
```

Příkazy pro výpis proměnných

- >> `a` vypíše hodnotu proměnné `a`, nebo zprávu, že není definována
- >> `who` vypíše seznam všech aktuálních proměnných
- >> `whos` totéž co `who`, navíc o každé proměnné detailní informace (totéž viz okno Workspace)

Příkazy pro uložení proměnných z pracovního prostoru do souboru

- >> `save` uloží všechny proměnné z pracovního prostoru do souboru `MATLAB.mat` v aktuálním adresáři
- >> `save jméno` totéž – do souboru `jméno.mat`
- >> `save jméno a b c` -totéž, uloží jen uvedené proměnné
- >> `save jméno.ext a b c - ascii` uloží uvedené proměnné do znakového souboru `jméno.ext` bez ohledu na uvedenou příponu

Příkazy pro načtení proměnných ze souboru do pracovního prostoru

- >> `load` načte ze souboru `MATLAB.mat` všechny proměnné v něm uložené zpět do pracovního prostoru
- >> `load jméno` načte ze souboru `jméno.mat` všechny proměnné ...
- >> `load jméno.ext` načte matici čísel z tohoto textového souboru do proměnné `jméno` (je-li `ext` \neq `mat`, je uvedený soubor chápán jako textový).