

**MATrix LABORatory**. Nejnovější je verze 6 (release 12).  
Základní internetový odkaz - <http://www.mathworks.com>.

## 1. Prostředí, stručný popis oken

Command Window – příkazové okno pro zadávání příkazů v jazyku Matlabu.

Workspace – zde se zobrazuje obsah paměti; je možné jednotlivé proměnné editovat.

Command History – dříve zadané příkazy; označení datem.

Current Directory – prohlížeč souborů a adresářů.

Launch Pad – rychlý přístup k hlavním součástem MATLABu.

Help / Matlab Help v hlavní nabídce - zobrazí nápovědu.

File / Set Path nastaví cesty k používaným adresářům.

File / Preferences – základní nastavení komponent Matlabu.

## 2. Základní vlastnosti

Základní datový typ je matice.

Za znakem % následuje komentář.

Středník za příkazem potlačí zobrazení výsledku.

Příkazy na jednom řádku se oddělují čárkami nebo středníky.

Má-li příkaz pokračovat na dalším řádku, napíše se na konci ... a stiskne Enter.

MATLAB rozlišuje velká a malá písmena (case-sensitive).

Standardně se počítá s desetinnými čísly, vyjádření výsledků ve zlomcích zajistí příkaz **format rat**. Příkaz **format long** nebo **format long e** (zobrazuje se všech 16 cifer), **format short** (4 desetinná místa).

## 3. Některé matematické funkce

<b>cos</b>	kosinus
<b>sin</b>	sinus
<b>tan</b>	tangens
<b>exp</b>	exponenciala o zakladu e
<b>log, log10, log2</b>	logaritmus přirozený, o základu 10 a 2
<b>sqrt</b>	(druhá) odmocnina
<b>abs</b>	absolutní hodnota
<b>imag, real, conj</b>	imaginární a reálná část komplexního čísla, komplexně sdružené číslo
<b>round, floor, ceil</b>	zaokrouhlení, zaokrouhlení doleva, doprava

## 4. Relační a logické operátory

<, <=, >, >=, ==, ~=

&, |, ~

## 5. Operace s maticemi

<b>u = [7 8 9]</b>	řádkový vektor
<b>v = [3; 4; 5]</b>	sloupcový vektor
<b>v = u'</b>	transpozice vektoru
<b>w = 2:8</b>	vektor [ 2 3 4 5 6 7 8]
<b>y = 2:3:20</b>	vektor [2 5 8 11 14 17 20]
<b>y = 3: end</b>	prvky vektoru y s indexy od 3. do posledního
<b>X = linspace(a, b, N)</b>	řádkový vektor N čísel ekvidistantně rozložených mezi a a b, X(1)=a, X(N)=b
<b>X = logspace(a, b, N)</b>	řádkový vektor N čísel exponenciálně rozložených mezi a a b, X(1)=a, X(N)=b
<b>A = [1 2 3; 5 6 7]</b>	matice o 2 řádcích a 3 sloupcích
<b>B = sparse(A)</b>	řádká matice
<b>eye(3)</b>	jednotková matice 3x3
<b>ones(3)</b>	matice jedniček

<b>zeros(3)</b>	matice nul
<b>diag(w)</b>	diagonální matice s vektorem w na diagonále
<b>diag(A)</b>	diagonála matice A (vektor)
<b>spdiags(B, d, m, n)</b>	řídká matice typu m x n vytvořená ze sloupců matice B umístěných na pozice určené vektorem d
<b>full(A)</b>	zobrazení řídké matice v „plném“ tvaru
<b>spy(A)</b>	schematické zobrazení „zaplnění“ matice A
<b>rand(3,4)</b>	matice typu 3 x 4 náhodných čísel mezi 0 a 1 (rovnoměrné rozdělení)
<b>rand('state',0)</b>	nastavení generátoru náhodných čísel do stavu 0
<b>randn(3)</b>	matice náhodných čísel s normálním rozdělením (střední hodnota je 0, rozptyl je 1)
<b>ones(4, 8)</b>	obdélníková matice jedniček
<b>zeros(size(B))</b>	nulová matice stejného typu jako B
<b>M = zeros(3, 4, 5)</b>	trojrozměrné pole nul
<b>B(3,1) = 8</b>	změna prvku v matici
<b>B(k)</b>	k-tý prvek v matici (počítáno po sloupcích)
<b>1:8 nebo 1:0.3:2</b>	operátor dvojtečka: vektor [1 2 3 4 5 6 7 8] nebo [1 1.3 1.6 1.9]
<b>u = A(:, 3)</b>	vektoru u se přiřadí hodnoty třetího sloupce z matice A, symbol dvojtečka se zde dosazuje místo hodnoty "všechny"
<b>A(2, :) = v</b>	druhému řádku v matici A se přidělí hodnoty vektoru v
<b>A([2 5], :)</b>	2. a 5. řádek v matici A
<b>A([2 5], :) = A([5 2], :)</b>	prohození 2. a 5. řádku v matici A
<b>A(2 : 5, 3 : 4)</b>	submatice matice A složená z průsečíků řádků 2 až 5 a sloupců 3 až 4
<b>A(:)</b>	sloupcový vektor složený ze všech sloupců matice A
<b>B = [A b]</b>	matice složená z matice A a vektoru b
<b>B = [A A'; inv(A), A]</b>	matice složená ze 4 submatic
<b>B = A([2 1 3], :)</b>	permutace řádků matice
<b>A(:, 2) = []</b>	odstranění 2. sloupce z matice
<b>v(2: 2: 10) = []</b>	odstranění všech sudých složek vektoru
Operátory: +, -, ./, \ (tzv. dělení "zleva"), ^ (umocnění), ' (vytvoření adjungované matice), .* (násobení po prvcích), ./ (dělení po prvcích), \ (dělení zleva po prvcích), .^ (umocnění po prvcích), .' (transpozice bez komplexního sdružení)	
<b>A = A + 1</b>	přičtení 1 ke všem prvkům matice A
<b>B = A + ones(3)</b>	součet dvou matic
<b>v = B * u</b>	násobení matic (vektorů)
<b>A * B</b>	násobení matic
<b>A .* B</b>	násobení matic "po prvcích"
<b>A'</b>	transpozice matice
<b>reshape(A, m, n)</b>	změna tvaru matice – sloupce se seřadí do jednoho a ten se poskládá do tvaru m x n
<b>sum(A)</b>	součet prvků ve sloupcích matice A, výsledek je řádkový vektor
<b>inv(A)</b>	inverse matice (A musí být čtvercová a regulární)
<b>pinv(A)</b>	pseudoinverse matice A
<b>x = A \ B</b>	maticové dělení - vyřešení soustavy rovnic s maticí A (čtvercová a regulární) a maticí pravé strany B (používá se zhruba řečeno Gaussova eliminace); jestliže A není regulární, je řešení získáno metodou nejmenších čtverců (pro přeурčené soustavy) nebo je výsledkem jedno řešení nehomogenní soustavy s největším počtem nulových složek (pro nedourčené soustavy)
<b>x = pinv(A) * B</b>	řešení nedourčené soustavy lineárních rovnic pomocí Moore-Penroseovy pseudoinverze (minimalizace normy vektoru řešení)
<b>null(A)</b>	báze nulového prostoru matice A
<b>A / B</b>	maticové dělení - výsledkem je B' \ A'
<b>A ./ B</b>	dělení "po prvcích a / b"
<b>A \ B</b>	dělení "po prvcích b / a"
<b>dot(x, y)</b>	skalární součin
<b>triu(A,n)</b>	horní trojúhelníková matice (od n-té diagonály) z matice A
<b>tril(A,m)</b>	dolní trojúhelníková matice (od m-té diagonály) z matice A
<b>det(A)</b>	determinant matice A (čtvercová)

<b>rank(A)</b>	hodnota matice A
<b>norm(A, p)</b>	norma matice A- sloupcová (p=1), řádková (p=Inf), největší singulární hodnota – viz. funkce <b>svd</b> (p=2), Frobeniova (p='fro')
<b>norm(v, p)</b>	$l_p$ norma vektoru v ( $1 \leq p \leq \text{Inf}$ )
<b>size(A)</b>	dvojice čísel [m n] (počet řádků a sloupců)
<b>length(A)</b>	ten větší z rozměrů matice
<b>numel(A)</b>	počet prvků v matici A
<b>trace(A)</b>	stopa
<b>[L, U, P] = lu(A)</b>	LU-rozklad matice A; platí $PA = LU$
<b>e = eig(A)</b>	vektor vlastních čísel matice A
<b>[U, S, V] = svd(A)</b>	rozklad na singulární čísla: $A = U * S * V'$ , U a V jsou unitární matice, S je diagonální matice singulárních čísel
<b>conv(u, v)</b>	konvoluce vektorů u, v
<b>isprime(A)</b>	test na prvočíslo, výsledek je 0 nebo 1
<b>finite(A)</b>	test na Inf nebo NaN, výsledek je 0 nebo 1
<b>mod(n, p)</b>	zbytek po celočíselném dělení n/p
<b>find(A &gt; 7)</b>	vyhledávání
<b>sort(A)</b>	třídění

## 6. Pole s dimenzí větší než 2

Třetí dimenzi představuje 3. index, je nazývána „page“.

**A = zeros(2, 3, 4)** trojrozměrné pole nul  
**B = zeros(7, 8);**  
**B(:, :, 2) = ones(7, 8)** přidání druhé „stránky“ k matici B

## 7. Polynomy

**p = [1 -2 8]** reprezentace polynomu pomocí vektoru jeho koeficientů (v pořadí od nejvyšší mocniny)  
**polyval(p, A)** vyčíslení polynomu p v prvcích matice A  
**polyvalm(p, A)** vyčíslení polynomu v matici A (dosazení A za nezávisle proměnnou)  
**roots(p)** kořeny polynomu p  
**p = polyfit(x, y, n)** aproximace dat [x, y] polynomem stupně n ve smyslu nejmenších čtverců

## 8. Znaky a řetězce

**a = 'slovo'** proměnná typu řetězec  
**a'** transpozice  
**a(2:4)** část řetězce  
**disp(a)** zobrazení bez označení proměnné  
**str2num(s)** převede pole řetězců na pole znaků

## 9. Některé optimalizační funkce

**p = polyfit(x, y, n)** aproximace dat [x, y] polynomem stupně n ve smyslu nejmenších čtverců  
**yk = pchip(x, y, xk)** interpolace po částech kubickým Hermiteovým polynomem  
**yk = spline(x, y, xk)** interpolace kubickou spline funkcí  
**lsqcurvefit(@F, [x0], xdata, ydata)** nelineární interpolace (metodou nejmenších čtverců); řeší úlohu najít x, pro které výraz  $\text{norm}(F(x, xdata) - ydata, 2)^2$  nabývá svého minima  
**x = fsolve(@F, x0)** řešení rovnice  $F(x) = 0$ ; počáteční přiblížení x0

## 10. Funkce pro říbližné řešení diferenciálních rovnic

**[T, Y] = ode45(@F, [0 1], [a b])** metoda Runge-Kutta 4.řádu pro soustavy diferenciálních rovnic  
**odeset('AbsTol', 1e-6)** nastavení parametrů pro řešiče diferenciálních rovnic

## 11. Některé další důležité konstanty a funkce

<b>pi</b>	číslo $\pi$
<b>i</b> nebo <b>j</b>	imaginární jednotka
<b>real(x)</b>	reálná část komplexního čísla/vektoru
<b>imag(x)</b>	imaginární část komplexního čísla/vektoru
<b>eps</b>	relativní přesnost pro operace v pohyblivé řádové čárce; např. $1 + 2^{(-53)}$ je 1
<b>realmin</b>	$2^{(-1022)}$ nejmenší nenulové reálné číslo, (ale ve skutečnosti $2^{(-1074)} \neq 0$ , $2^{(-1074)} = 0$ )
<b>realmax</b>	$(2-\text{eps})^{1023}$ největší reálné číslo menší než Inf, (ale $(2-\text{eps})^{1024} < \text{Inf}$ )
<b>Inf, inf</b>	"nekonečno", např. $1/0$ nebo hodnota větší než realmax
<b>NaN, nan</b>	nenumernický výraz (Not-a-Number)
<b>set()</b>	nastavení parametrů objektů
<b>get()</b>	parametry objektů
<b>f = inline('sin(x+a)', 'x', 'a')</b>	jednořádková definice funkce
<b>clear all, clear functions, clear JmenoFunkce</b>	vymazání z paměti

## 12. M-soubory

M-soubory jsou textové soubory s koncovkou .m, do kterých se píše funkce a skripty. Spouští se napsáním jména souboru na příkazový řádek a stisknutím Enter. Je vhodné volit název funkce stejný jako název souboru. Skript může pracovat s proměnnými v paměti MATLABu a také po skončení běhu skriptu hodnoty proměnných zůstanou v paměti. Skripty nemají vstupní ani návratové hodnoty. Funkce používá vlastní paměťový prostor, může mít vstupní argumenty a vracet hodnoty. Po definici hlavní funkce mohou v m-souboru následovat definice dalších funkcí (tzv. subfunctions). Funkce i skripty obsahují příkazy v jazyku MATLAB.

<b>if</b> ( $n > 0 \ \&\& \ \text{mod}(n, 2) == 0$ ) $m = n / 2$ <b>else</b> $m = n$ <b>end</b> ;	rozhodování
<b>if</b> ( $x < a$ ) $x = a$ , <b>elseif</b> ( $x > b$ ) $x = b$ , <b>else</b> $x = (a + b) / 2$ , <b>end</b> ;	rozhodování
<b>switch</b> ( <i>Vyraz</i> ) <b>case</b> <i>Hodnota, Prikaz, case</i> <i>Hodnota, Prikaz, ..., otherwise</i> <i>Prikaz, end</i> ;	rozhodování – příkaz switch
<b>for</b> $i = 1 : 10$ , $x(i) = i^2$ , <b>end</b> ;	cyklus
<b>for</b> $i = 1 : 2 : 10$ , $x(i) = 0$ , <b>end</b> ;	cyklus
<b>while</b> ( <i>Vyraz</i> ) <i>Prikaz</i> <b>end</b> ;	cyklus
<b>continue</b>	skok na další smyčku cyklu
<b>break</b>	ukončení cyklu
<b>disp</b> ('Chyba!')	výpis na příkazový řádek

**function** [*VystupniData*] = *JmenoFunkce*(*VstupniData*) *Prikazy* **end**      zápis funkce

**function** [maxim, minim] = *extremy* (a, b, delta)  
 $i = 1$ ;  
for  $c = a : \text{delta} : b$ ;  $x(i) = \sin(c)*c$ ;  $i = i + 1$ ; end;  
maxim = max(x);  
minim = min(x);  
plot(a : delta : b, x);

[u,v] = *extremy*(1,30,0.5)      příklad volání funkce

## 13. Soubory a adresáře

<b>!dir</b>	za znakem ! následují příkazy systému
<b>path</b>	zobrazení prohledávaných cest (lze i z hlavní nabídky)
<b>addpath</b> <i>Cesta</i>	přidání k seznamu prohledávaných cest (lze i z hlavní nabídky)
<b>cd</b> <i>JmenoAdresare</i>	změna aktuálního adresáře
<b>type</b> <i>JmenoSouboru</i>	výpis souboru
<b>whos</b>	zobrazí proměnné v paměti a jejich hodnoty
<b>save</b> ' <i>JmenoSouboru</i> '	proměnné a jejich hodnoty se uloží do souboru <i>JmenoSouboru.mat</i>
<b>load</b> ' <i>JmenoSouboru</i> '	zkopíruje se do paměti obsah souboru <i>JmenoSouboru.mat</i>
<b>diary</b>	uložení příkazů z klávesnice do souboru
<b>ver</b>	informace o Matlabu a toolboxech

**s1 = fopen('soubor', 'w')** otevření souboru pro zápis  
**fprintf(s1, '%6.2f, %12d\n', x, n)** formátovaný zápis do souboru (syntaxe dle jazyka C)  
**s2 = fopen('soubor', 'r')** otevření souboru pro čtení  
**A = fscanf(s2, '%6.2f, %12d\n')** čtení formátovaných dat ze souboru  
**fclose(s1)** uzavření souboru, uložení na disk  
**y = wavread('soubor.wav')** čtení zvukových wav-souborů  
**wavwrite(y, 'soubor.wav')** zápis zvukového wav-souboru

## 14. Kreslení

Matlab umožňuje pokročilé grafické operace využitím „ukazatelů“ na grafické objekty. Se stromovou strukturou grafických objektů se lze seznámit v nápovědě: z hlavní nabídky Help / Matlab Help / karta Contents / složka Matlab / Reference / Graphics Object Property Browser.

**fplot('sin', [-2 10])** vykreslí funkci  
**x = 0 : pi/100 : 2\*pi; y = sin(x); plot(x, y)** vykreslí body o souřadnicích xi, yi spojené lomenou čarou  
**plot(y)** vykreslí body o souřadnicích i, yi spojené lomenou čarou  
**plot(x, y, ':', x, sin(x-0.3), 'b--')** vykreslí dva grafy do stejného obrázku, první tečkovaně, druhý čárkovaně modře  
**plot(A)** vykreslí sloupce matice (několik sérií bodů spojených lomenou čarou)  
**plot(Z)** je-li Z vektor komplexních čísel, vykreslí se body o souřadnicích real(Z) a imag(Z)  
**t=0:0.2:2\*pi; plot(cos(t)+i\*sin(t), '\*')** kreslení komplexního vektoru (kružnice)  
**t = 0 : pi/50 : 10\*pi; plot3(sin(t), cos(t), t)** parametrické 3-D kreslení  
**t = 0:0.01:2\*pi; polar(t, abs(sin(2 \* t) .\* cos(2 \* t)));** parametrické kreslení  
**[x, y, z] = pol2cart(theta, r, z)** převedení z polárních nebo cylindrických souřadnic do kartézských  
**[theta, r] = cart2pol(x, y)** převedení z kartézských souřadnic do polárních nebo cylindrických  
**[x, y, z] = sph2cart(theta, phi, r)** převedení ze sférických souřadnic do kartézských  
**[theta, phi, r] = cart2sph(x, y, z)** převedení z kartézských souřadnic do sférických

**xlabel('rychlost [m/s]')** označení vodorovné osy  
**ylabel('čas [s]')** označení svislé osy  
**title('Pohyb kyvadla')** nadpis grafu  
**axis square, axis([-pi pi -1 1])** nastavení os souřadnic  
**grid on, grid off** zapnutí / vypnutí zobrazení souřadnic  
**text(1, 1.5, '\itNalezněte lokální maxima.')** umístění textu do grafu, používá se jazyka TeX  
**hold on** výstup do existujícího (posledního vytvořeného) grafu  
**hold off** překreslení předchozího grafického výstupu  
**subplot(m, n, p)** rozdělení aktuálního grafického okna na mxn částí v m řádcích a n sloupcích a nastavení grafického výstupu do p-tého z nich (počítáno po řádcích)  
**cla** vymazání aktivního grafu  
**clf** vymazání aktivního grafického okna

**[X, Y] = meshgrid(x, y)** vytvoří uspořádanou množinu všech uspořádaných dvojic ze složek vektorů x a y, např.:  
**[X, Y] = meshgrid(0:0.1:3, 0:0.4:5); surf(X, Y, sin(X .\* Y))**  
**[X, Y] = meshgrid(0:0.1:3, 0:0.1:3); surf(X, Y, sin(X .\* Y), 'FaceColor', 'red', 'EdgeColor', 'none');...**  
**camlight left; lighting phong** jiný příklad použití  
**contour(X, Y, cos(X + Y))** vrstevnice  
**contourf(X, Y, cos(X + Y))** vrstevnice (vybarvené plochy)  
**surf([0 1; 0 1; 0 1], [0 0; 1 1; 2 2], [1 1; 6 6; 1 1])** kreslení plochy po trojúhelnících (střecha)  
**surfc(X, Y, cos(X + Y))** kreslení plochy a vrstevnic  
**for k = 1: 50 ; y(k) = rand(1, 1) \* k; plot(x, y); M(k) = getframe; end; movie(M, 2, 30)** "pohyblivé" obrázky, přehrání (1x při nahrávání +) 2x, a to frekvencí 30 obr./s  
**movie2avi(M, 'soubor')** převedení sekvence obrázků na soubor formátu avi  
**disp('Slova')** výpis hlášení  
**text(x, y, 'Slova')** umístění textu do obrázku na souřadnice x, y  
**b = ginput(3)** vstup souřadnic 3 bodů vybraných pomocí myši  
**gca** "ukazatel" na aktivní graf  
**gcf** "ukazatel" na aktivní grafické okno

**set()** nastavení parametrů grafického výstupu, např.:  
**set(gca, 'DataAspectRatio', [1 1 1])** nastavení poměrů měřítek os souřadnic v aktivním grafu  
**get()** odečtení parametrů grafického výstupu, např.:  
**get(gcf, 'Color')** barva pozadí aktivního grafického okna