

Začínáme (help general)

help nápověda, pro konkrétní fci **help sin**
doc dokumentace, pro konkrétní fci **doc sin**
lookfor vyhledání funkce (**lookfor cosine**)
clc vymazání Command Window

Proměnné

MATLAB rozlišuje velikost znaků. Název proměnné musí začínat písmenem. Základní proměnnou je dvourozměrné pole reálných čísel (64 bitů).

Skalár je pole 1×1 .

Řádkový vektor délky n je pole $1 \times n$.

Sloupcový vektor délky m je pole $m \times 1$.

Matice s m řádky a n sloupci je pole $m \times n$.

Řetězec obsahuje text. (**help strfun**)

clear a smazání proměnné **a**

clear smazání všech proměnných

Předdefinované konstanty

ans výsledek předchozího výpočtu
eps tzv. strojová přesnost
Inf nekonečno, např. $1/0 = \text{Inf}$
NaN nedefinovaný výraz, např. $0/0 = \text{NaN}$
pi hodnota π (3.1415...)
1i, 1j komplexní jednotka

Vytvoření matice/vektoru

Prvky na řádku jsou odděleny mezerou nebo čárkou, středník ukončuje řádek.

v = [1 2 3 4 5]	řádkový vektor
w = [5; 2; 3; 4]	sloupcový vektor
A = [1,2,3; 4,5,6; 7,8,9]	matice

Nebo též takto

```
A = [1,2,3  
     4,5,6  
     7,8,9]
```

length délka vektoru nebo největší rozměr pole
size rozměry pole, **[m,n] = size(A)**

Vytvoření speciálních matic/vektorů

(help elmat)

<code>linspace(a,b,N)</code>	pravidelné dělení intervalu $[a, b]$ obsahující N bodů (včetně a, b)
<code>i:j:k</code>	aritmetická posloupnost, první prvek i , diference j , horní mez k (resp. dolní mez pro záporné j)
<code>i:k</code>	aritmetická posloupnost s diferencí 1
<code>zeros(m,n)</code>	pole nul $m \times n$
<code>zeros(n)</code>	pole nul $n \times n$
<code>ones(m,n)</code>	pole jedniček $m \times n$
<code>ones(n)</code>	pole jedniček $n \times n$
<code>eye(m,n)</code>	pole nul $m \times n$ s jedničkami na diagonále
<code>eye(n)</code>	jednotková matice $n \times n$
<code>rand(m,n)</code>	pole $m \times n$ náhodných čísel
<code>rand(n)</code>	pole $n \times n$ náhodných čísel

Indexování proměnných

Číslování začíná od jedničky!

<code>v(1)</code>	první prvek vektoru v
<code>v(end)</code>	poslední prvek vektoru v
<code>v(2:3:9)</code>	druhý, pátý a osmý prvek vektoru v
<code>A(2,3)</code>	prvek matice A ve druhém řádku, třetím sloupci
<code>A(:,3)</code>	třetí sloupec matice A
<code>A(1,:)</code>	první řádek matice A
<code>A(1:2:end,:)</code>	matice obsahující liché řádky matice A
<code>A(1:2,2:4)</code>	podmatice A obsahující 1.–2. řádek, 2.–4. sloupec
<code>A(1,end)</code>	poslední prvek prvního řádku matice A

Operace s maticemi/vektory

(help arith, help ops)

<code>+</code>	sčítání		
<code>-</code>	odčítání		
<code>*</code>	násobení	<code>.*</code>	násobení po složkách
<code>/</code>	dělení	<code>./</code>	dělení po složkách
<code>^</code>	mocnění	<code>.^</code>	mocnění jednotlivých složek
<code>'</code>	komplexní sdružení	<code>.'</code>	transpozice

Matematické funkce

(help elfun, help matfun)

Následující funkce mají očekávaný význam:

`abs`, `exp`, `log`, `log10`, `log2`, `sqrt`, `sin`, `asin`, `cos`, `acos`, `tan`, `atan`,
`floor`, `ceil`, `round`, `max`, `min`, `norm`, `rank`, `det`, `inv`, `sort`, `sum`.

Vstupem těchto funkcí mohou být i vektory a matice.

Skripty (“m-files”)

Posloupnost příkazů, tzv. skript je uložen v textovém souboru s příponou `.m`. Skript lze spustit zadáním jména souboru (bez přípony) v příkazovém řádku, nebo tlačítkem “Run” v okně editoru.

`;` potlačení výstupu, umísťuje se na konec příkazu
`%` začátek komentáře (celý zbytek řádku)
`...` pokračování příkazu/výrazu na dalším řádku

Funkce

Funkce jsou definovány v jednotlivých souborech pojmenovaných stejně jako funkce `NazevFunkce.m` následovně:

```
function [out1,...,outN] = NazevFunkce(in1,...inM)
% NazevFunkce: Stručný popis (volitelně)
% ...
příkazy;
```

Funkci poté spustíme takto

```
[výstup1,...,výstupN] = NazevFunkce(vstup1,...vstupM)
```

Logické operátory (help relop)

`<` menší než
`<=` menší nebo rovný
`>` větší než
`>=` větší nebo rovný
`==` rovný
`~=` různý
`&` logické AND (“a zároveň”)
`|` logické OR (“nebo”)
`~` negace

Cykly (help lang)

```
for k = Vektor
    příkazy;
end
Obvykle: for k = 1:n

while LogickýVýraz
    příkazy;
end
```

Pro náročnější výpočty je žádoucí se cyklům vyhýbat.

Větvení kódu (help lang)

```
if LogickýVýraz
    příkazy;
elseif LogickýVýraz      % Volitelně
    příkazy;
else                      % Volitelně
    příkazy;
end
```

Vykreslování výsledků (help graph2d, help graph3d)

figure	vytvoření nového okna pro graf
close	zavření aktuálního okna
close all	zavření všech oken
plot	základní 2D graf
semilogy	2D graf s logaritmicky škálovanou osou y
imagesc	zobrazení obrázku, vhodné pro zobrazení prvků matice
plot3	zobrazení křivky/bodů ve 3D
surf	zobrazení plochy ve 3D
legend	legenda ke grafu
title	název grafu
xlabel	popisek osy x
ylabel	popisek osy y