

DZS - cvičení č. 1 - úvod do Matlabu

Obsah

- | [Úvod](#)
- | [Matice v Matlabu](#)
- | [Prostředky pro tvorbu programu/skriptu](#)
 - | [Výrazy](#)
 - | [Konstanty](#)
 - | [Řídící struktury](#)
 - | [Elementární funkce](#)
 - | [Elementární fce pro manipulaci s maticemi](#)
 - | [Datové analýzy, FFT](#)
 - | [Grafy](#)

Úvod

Matlab je matematický programovací jazyk a interaktivní prostředí optimalizované pro maticové operace. Uživatel v tomto prostředí pracuje za pomoci programovacího jazyka s poměrně jednoduchou syntaxí. Příkazy tohoto jazyka je Matlabu možno zadávat jednak skrze příkazovou řádku a nebo prostřednictvím skriptu - skupiny příkazů uložených v souboru jména {prikaz}.m. Takový skript je pak možné vyvolat z příkazové řádky Matlabu příkazem {prikaz} za předpokladu, že aktuálním adresářem Matlabu je adresář se skriptem {prikaz}.m. (V rámci příkazové řádky Matlabu je možné se po souborovém systému pohybovat příkazem `cd`, zjistit aktuální adresář příkazem `pwd` a vypsát obsah adresáře příkazem `dir`, více viz. **help matlab\general**).

Matlab disponuje kvalitní nápovědou kterou je možné vyvolat pomocí příkazu `help`.

```
help
help toolbox (např. help signal )
help prikaz
```

Pokud si nejste jisti co hledáte, pomůže vám příkaz `lookfor`

Základním datovým prvkem Matlabu je matice. Jak skaláry tak vektory jsou matice(1x1, 1xn, nx1 prvkové matice).

Matice v Matlabu

Definice

Základními způsoby pro definici matice jsou v Matlabu

- | výčet - jednotlivé prvky matice jsou zadány - `A=[1 2 3;3 4 5]`
- | generování - `rand(m,n)`, `zeros(m,n)`, `ones(m,n)`; m- počet řádků, n - počet sloupců
`a = [1:2:10;1:1:5]` - 2 řádková matice, 1. řádka od 1 do 10 krok 2, 2. řádka od 1 do 5 krok 1.
- | načtení z externího datového souboru - příkazy `save`, `load`

Výběr skaláru, vektoru či submatice z matice

K indexaci prvků matice se využívají kulaté závorky

- | `b = A(2,1)` proměnné `b` je přiřazena hodnota pole matice `A` na řádce 2, sloupci 1
- | `B = A(1:2, 2:3)` proměnné `B` je přiřazena submatice matice `A` sestávající z jejího prvního a druhého řádku omezeného na druhý a třetí sloupec
- | `B = A(:, 2:3)` proměnné `B` je přiřazena submatice matice `A` sestávající ze všech jejích řádků omezených na sloupce 2 a 3.

Prostředky pro tvorbu programu/skriptu

Výrazy

Výrazem se rozumí kombinace proměnných, konstant, operátorů a funkcí vracející hodnotu - matici.

```
A = rand(2,2); B = rand(2,2);  
C = A + B; C = A * B; C = A.^2; C = inv(A);
```

Základními operátory pro tvorbu výrazů jsou: `+` ; `-` ; `*` ; `/` ; `^` ;
' (transpozice) ; a jejich ekvivalenty znamenající operaci prvek po prvku - `.*` ;
`./` ; `.^`.

Seznam operátorů je možné v matlabu získat pomocí příkazu **help matlab\ops**.

Konstanty

<code>i, j</code>	komplexní jednotka
<code>inf</code>	nekonečno
<code>NaN</code>	nečíselná hodnota

Řídící struktury

Základními řídicími strukturami jsou cykly a podmínky.

- | **Cyklus for** (help for)

Cyklus slouží k opakovanému vykonávání části skriptu. Počet iterací je dán.

```
for I = 1:N, %pro I od jedné do N  
    for J = 1:N, %pro J od jedné do N  
        A(I,J) = 1/(I+J-1); %přiřad' prvku matice A na  
                            řádce I a sloupci J  
                            hodnotu 1/(I+J-1)  
    end  
end
```

Přerušit provádění příkazu FOR lze provést zevnitř iterovaného bloku skriptu příkazem `break`.

Vynutit přechod na další iteraci příkazu FOR lze provést zevnitř iterovaného bloku příkazem `continue`.

┆ **Cyklus while** (`help while`)

Cyklus slouží k opakovanému vykonávání části skriptu do doby kdy platí podmínka - počet iterací není ve skriptu definován.

```
while a>b,                                %dokud je a větší než b
    prováděj b = b + c^2
    b = b + c^2;
end
```

┆ **Podmíněné vykonání - if** (`help if`)

```
if I == J                                %pokud I je rovné J, A
    (I,J) = 2
    A(I,J) = 2;
elseif abs(I-J) == 1                    %pokud I není rovné J ale
    abs (I-J)=1 => A(I,J) = -1
    A(I,J) = -1;
else                                    %pokud nebyly předchozí
    podmínky splněny A(I,J)=0
    A(I,J) = 0;
end
```

┆ **Větvení na základě hodnoty výrazu - switch** (`help switch`)

```
switch lower(METHOD)                  %větvit se bude na
    základě hodnoty proměnné
    METHOD
    case
    {'linear','bilinear'}               %METHOD=='linear' ||
    METHOD=='bilinear' =>
    disp('method is linear')
    disp('Method is
linear')
    case 'cubic'                        %analogicky s předchozím
    disp('Method is cubic')
    case 'nearest'
    disp('Method is
nearest')
    otherwise                           %pokud hodnota METHOD
    neodpovídá žádnému case
    => disp('unknown method')
    disp('Unknown method.')
end
```

Více k řídicím strukturám se lze dovědět v nápovědě Matlabu - **help matlab\lang**

Elementární funkce (help elfun)

Níže je uveden nepřeložený seznam elementárních funkcí, bližší informace k jednotlivým funkcím lze získat pomocí `help {funkce}`

Trigonometric. `sin` - Sine.
`sinh` - Hyperbolic sine.
`asin` - Inverse sine. `asinh` - Inverse hyperbolic sine.
`cos` - Cosine.
`cosh` - Hyperbolic cosine.
`acos` - Inverse cosine.
`acosh` - Inverse hyperbolic cosine.
`tan` - Tangent.
`tanh` - Hyperbolic tangent.
`atan` - Inverse tangent.
`atan2` - Four quadrant inverse tangent.
`atanh` - Inverse hyperbolic tangent.
`sec` - Secant.
`sech` - Hyperbolic secant.
`asec` - Inverse secant.
`asech` - Inverse hyperbolic secant.
`csc` - Cosecant.
`csch` - Hyperbolic cosecant.
`acsc` - Inverse cosecant.
`acsch` - Inverse hyperbolic cosecant.
`cot` - Cotangent.
`coth` - Hyperbolic cotangent.
`acot` - Inverse cotangent.
`acoth` - Inverse hyperbolic cotangent.

Exponential.
`exp` - Exponential.
`log` - Natural logarithm.
`log10` - Common (base 10) logarithm.
`log2` - Base 2 logarithm and dissect floating point number.
`pow2` - Base 2 power and scale floating point number.
`realpow` - Power that will error out on complex result.
`reallog` - Natural logarithm of real number.
`realsqrt` - Square root of number greater than or equal to zero.
`sqrt` - Square root.
`nextpow2` - Next higher power of 2.

Complex.
`abs` - Absolute value.
`angle` - Phase angle.
`complex` - Construct complex data from real and imaginary parts.
`conj` - Complex conjugate.
`imag` - Complex imaginary part.
`real` - Complex real part.
`unwrap` - Unwrap phase angle.
`isreal` - True for real array.

cplxpair - Sort numbers into complex conjugate pairs.

Rounding and remainder.

fix - Round towards zero.

floor - Round towards minus infinity.

ceil - Round towards plus infinity.

round - Round towards nearest integer.

mod - Modulus (signed remainder after division).

rem - Remainder after division.

sign - Signum.

Elementární fce pro manipulaci s maticemi

Níže uvedený seznam obsahuje elementární funkce pro práci s maticemi - jejich generování apod. Nápovědu v jednotlivém funkčním lze opět získat pomocí příkazu `help {funkce}`

<code>zeros</code>	Fce generuje matici daných rozměrů tvořenou nulovými prvky
<code>ones</code>	analogicky k předchozí fci, ale s jednotkami
<code>eye</code>	jednotková matice
<code>rand</code>	generuje matici požadovaných rozměrů; hodnoty prvků jsou náhodné hodnoty z intervalu (0,1) vybrané dle rovnoměrného rozdělení
<code>randn</code>	analogicky s předchozím bodem, ale normální rozdělení
<code>linspace</code>	generuje vektor s konstantním krokem
<code>logspace</code>	generuje vektor s logaritmickým krokem
<code>size</code>	rozměry matice
<code>length</code>	délka vektoru
<code>ndims</code>	počet dimenzí
<code>cat</code>	spojování matic
<code>find</code>	fce vrací indexy nenulových prvků
<code>end</code>	poslední index
<code>magic</code>	magický čtverec - součty sloupců a řádek jsou konst.

Fce pro datové analýzy, FFT

<code>max</code>	maximální prvek z vektoru/
------------------	----------------------------

	vektor maximálních hodnot sloupců matice
min	analogicky k předchozímu
mean	Střední hodnota prvků vektoru/vektor středních hodnot jednotlivých sloupců matice
var	rozptyl
sort	setřídění elementů ve vzestupném pořadí
sum	suma elementů
nonzeros	z vektoru vrátí subvektor nenulových prvků
diff	pro vektor je výsledkem vektor diferencí po sobě jdoucích prvků - vektor o 1 kratší, pro matici vrací matici tvořenou diferencemi hodnot po sobě jdoucích prvků bráno po sloupcích.
filter	digitální filter
conv	konvoluce
fft	diskrétní Fourierova transformace
fft2	2 rozměrná DFT
ifft	inversní DFT

Grafy

plot	graf v lineárních osách <code>t=0:0.05:10; plot(t,sin(t));</code>
loglog	graf v logaritmických osách
semilogx	graf s logaritmicky vynesenou nezávisle proměnnou
semilogy	analogicky k předch.
grid	zobrazení mřížky v grafu
subplot	možnost zobrazení více grafů v jednom okně
hold	možnost vynesení více grafů přes sebe