Príklady na operácie s prvkami matíc

- druhy matíc a ich reprezentácia v programovom prostredí MATLAB

- operácie s maticami

1. Matice

Matica je určitá množina čísel alebo iných matematických objektov (tzv. prvkov matice) usporiadaných do pravidelných riadkov a stĺpcov.

MATLAB vždy počíta s maticami (aj skalár je len matica typu 1x1)

1.1. Druhy matíc

Skalár

- je matica typu **1** x **1**
- Pre vytvorenie skaláru použijeme priradenie pomocou " =" . Ak nezadáme názov premennej automaticky sa vytvorí premenná ans (od slovička answer), kde sa uloží číslo.

Pozor, táto premenná bude stále prepisovaná, ak neurčime inú. V takom prípade prídeme o naše hodnoty uložené v tejto premennej!

```
>> a = 5

a =

  5
```

alebo

```
>> 5

ans =

  5
```

Matica m x n

- Maticu väčšiu ako 1x1 vždy zapisujeme do hranatých zátvoriek [ ]
- Zapisuje sa po riadkoch pričom jednotlivé elementy (prvky) matice sú oddelené čiarkou alebo medzerou
- Nový riadok vytvoríme zadaním bodkočiarky alebo stlačením klávesy ENTER pričom v tomto prípade ak nie sú zátvorky ukončené enter neodošle príkaz, iba nás posunie na ďalší riadok

```
>> A=[1,0
2,5
4,3]

A =

   1   0
   2   5
   4   3
```

alebo

```
>> B=[1 0; 2 5; 4 3]

B =

   1   0
   2   5
   4   3
```

alebo

```
>> C=[1 0
2 5
4 3]

C =

   1   0
   2   5
   4   3
```

alebo

```
>> D=[1, 0; 2, 5; 4, 3]

D =

   1   0
   2   5
   4   3
```

Riadkový vektor

- je matica typu **1** x **n**
- Vytvára sa podobne ako by sme vytvorili maticu n x m , ale použijeme len medzery alebo čiarky medzi jednotlivými číslicami
- Vytvorenie riadkového vektoru je možné aj dvojbodkovou konvenciou a to tak, že do hranatých zátvoriek zapíšeme 3 hodnoty, prvá je začiatočná hodnota, druhá hodnota kroku a tretia je konečná hodnota. Túto možnosť je výhodné využívať pri väčších vektoroch, ktoré majú byť rovnomerne rozdelené.

```
>> E = [1 2 3]

E =

     1     2     3
```

alebo

```
>> E = [1, 2, 3]

E =

     1     2     3
```

alebo

```
>> E=[0:1.2:5]

E =

         0    1.2000    2.4000    3.6000    4.8000
```

- V prípade ak neurčíme hodnotu kroku, bude automaticky rovný 1.

```
>> E=[0:5]

E =

     0     1     2     3     4     5
```

Stĺpcový vektor

- matica typu **m** x **1**
- stĺpcový vektor vytvoríme použitím bodkočiarky “;” alebo klávesy ENTER za každou číslicou

```
>> F = [1;2;3]

F =

     1
     2
     3
```

alebo

```
>> F = [1
2
3]

F =

     1
     2
     3
```

Jednotková matica

- je štvorcová matica, pričom na hlavnej diagonále má samé jednotky a zvyšné prvky matice tvoria nuly
- takúto maticu vieme vytvoriť dvoma spôsobmi a to tak ako by sme vytvárali maticu n x n, a po jednom vypisovali každý prvok matice, čo je podstatne prácnejšie a náchylnejšie na pomýlenie sa ako druhá možnosť a to je vytvorenie pomocou príkazu :

  G = ***eye*** *(rozmer_matice)*

```
>> G = eye(3)

G =

     1     0     0
     0     1     0
     0     0     1
```

Nulová matica

- všetky prvky matice sú nulové
- takúto maticu je tiež možné vytvoriť dvoma spôsobmi a to buď prácnym vypisovaním pre každý prvok nulu, alebo použitím príkazu pre štvorcovú maticu *H* = ***zeros****(n)*, alebo pre maticu typu m x n

H = ***zeros(****m,n****)***

```
>> H = zeros(3)

H =

  0   0   0
  0   0   0
  0   0   0
```

alebo

```
>> H = zeros(3,2)

H =

  0   0
  0   0
  0   0
```

Matica jednotiek

- všetky prvky matice sú jednotky
- tak ako predošle dve matice je možné vytvoriť dvoma spôsobmi, prvý je už asi zrejmý, ten druhý je príkaz *I* = ***ones(****n****)*** pre štvorcové matice a *I* = ***ones(****m,n****)*** pre matice typu **m** x **n**

```
>> I = ones (3)

I =

  1   1   1
  1   1   1
  1   1   1
```

alebo

```
>> I = ones (3,2)

I =

  1   1
  1   1
  1   1
```

Generovanie vektora s ekvidistantným krokom

- pre vytvorenie môžeme použiť príkaz *J* = ***linspace(****poč_hodnota, koneč_hodnota****)*** pričom tento príkaz vytvorí vektor o 100 hodnotách v intervale <poč_hodnota, koneč_hodnota>, alebo ak chceme vytvoriť n- hodnotový vektor za konečnú hodnotu napíšeme počet prvkov *J* = ***linspace(****poč_hodnota, koneč_hodnota, počet_prvkov* ***)***

```
>> J = linspace(1,15)

J =

  Columns 1 through 6

   1.0000   1.1414   1.2828   1.4242   1.5657   1.7071

  Columns 7 through 12

   1.8485   1.9899   2.1313   2.2727   2.4141   2.5556

  Columns 13 through 18
```

```
>> J = linspace(1,15,10)

J =

  Columns 1 through 6

   1.0000   2.5556   4.1111   5.6667   7.2222   8.7778

  Columns 7 through 10

  10.3333  11.8889  13.4444  15.0000
```

Matica náhodných čísel

- V programovom prostredí MATLAB existuje príkaz *K*=***rand(****veľkosť_matice****)***. Tento príkaz vygeneruje maticu náhodných čísel z intervalu <0,1>

```
>> K = rand(2,3)

K =

   0.9572   0.8003   0.4218
   0.4854   0.1419   0.9157
```

⇨ Druhou alternatívou je použiť príkaz *K=**randn(**veľkosť_matice**)***, ktorý použije čísla z normálneho rozdelenia

```
>> K = randn(3,2)

K =

  -0.4336   2.7694
   0.3426  -1.3499
   3.5784   3.0349
```

1.2. Základné operácie s maticami

Sčítanie a odčítanie matíc

⇨ tieto operácie sú definované len pre matice rovnakých rozmerov, inak nám MATLAB vypíše chybu

```
>> A=[1 0;2 5;4 3];
>> B=[2 3;5 8];
>> A-B
??? Error using ==> minus
Matrix dimensions must agree.

>> A+B
??? Error using ==> plus
Matrix dimensions must agree.
```

⇨ toto však neplatí pre sčítanie a odčítanie skaláru od matice

```
>> A=[2 3;5 8];
>> B= A+2

B =

    4    5
    7   10

>> C=A-2

C =

    0    1
    3    6
```

Sčítanie dvoch matíc:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \cdots & a_{1n}+b_{1n} \\ a_{21}+b_{11} & a_{22}+b_{22} & \cdots & a_{2n}+b_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \cdots & a_{mn}+b_{mn} \end{pmatrix}$$

Riešenie v programovom prostredí MATLAB:

```
>> A = [1,2,3;4,5,6;7,8,9];
>> B=[7,8,9;4,5,6;1,2,3];
>> C=A+B

C =

    8   10   12
    8   10   12
    8   10   12
```

inicializácia
výpočet

výsledok

- ⇨ odčítavanie funguje na tom istom spôsobe

Násobenie matíc

- ⇨ dve matice typu $m\ x\ n$ a $p\ x\ r$ môžeme násobiť iba za predpokladu, že $n = p$ a ako výsledok dostaneme maticu veľkosti $m\ x\ r$
- ⇨ V prípade že $n \neq p$ vypíše MATLAB chybu výnimku opäť tvorí skalár, ktorým vieme násobiť matice.

```
>> A=[1 0;2 5;4 3];
>> B=[2 3;5 8];
>> C=B*A
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

- ⇨ Matice násobíme takým spôsobom, že v prvej matici postupujeme po riadku tak, že 1. prvok vynásobíme z 1.prvkom z 1. stĺpca 2.matice, ďalej vynásobíme 2.prvok v riadku 1.matice s 2. prvkom v stĺpci z 2.matice ... . Takto pokračujeme až prídeme po posledné prvky, ktoré tiež vynásobíme a všetky nové hodnoty spočítame a tým vznikne prvok a_{11} z výslednej matice.
- ⇨ Následne pokračujeme 2.riadkom matice a opäť 1.stĺpcom. z tejto kombinácie dostaneme 2. prvok 1. stĺpca výslednej matice
- ⇨ Násobenie 2 matíc si ukážeme na príklade

PRÍKLAD 1

Vypočítajte a následne overte v programovom prostredí MATLAB súčin matíc $A = \begin{pmatrix} 1 & 0 \\ 2 & 5 \\ 4 & 3 \end{pmatrix}$ a $B = \begin{pmatrix} 2 & 3 \\ 5 & 8 \end{pmatrix}$.

$$\begin{pmatrix} 1 & 0 \\ 2 & 5 \\ 4 & 3 \end{pmatrix} * \begin{pmatrix} 2 & 3 \\ 5 & 8 \end{pmatrix} = \begin{pmatrix} 2+0 & 3+0 \\ 4+25 & 6+40 \\ 8+15 & 12+24 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 29 & 46 \\ 23 & 36 \end{pmatrix}$$

Riešenie súčinu matíc v programovom prostredí MATLAB:

```
>> A=[1 0;2 5;4 3]

A =

   1   0
   2   5
   4   3

>> B=[2 3;5 8]

B =

   2   3
   5   8
```

...

```
>> C=A*B

C =

   2   3
  29  46
  23  36
```

Transponová matica

- ⇨ je matica, ktorá vznikne výmenou jednotlivých riadkov za stĺpce a naopak.

PRÍKLAD 2

Vytvorte transponovanú maticu k matici $A = \begin{pmatrix} 1 & 0 \\ 2 & 5 \\ 4 & 3 \end{pmatrix}$ a následne overte svoje riešenie v programovom prostredí MATLAB.

Transponovaná matica A:

$$A' = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 5 & 3 \end{pmatrix}$$

Riešenie v programovom prostredí MATLAB:

```
>> A=[1 0;2 5;4 3]

A =

    1    0
    2    5
    4    3

>> D=A'

D =

    1    2    4
    0    5    3
```