Počítačová podpora v elektrotechnice

Ing. Lenka Šroubová, Ph.D.
lsroubov@kte.zcu.cz

Ing. Petr Kropík, Ph.D.
pkropik@kte.zcu.cz

Katedra teoretické elektrotechniky
Fakulta elektrotechnická
Západočeská univerzita v Plzni

Místnost: EK602

FACULTY OF ELECTRICAL ENGINEERING
UNIVERSITY OF WEST BOHEMIA

Informace

Univerzitní licence MATLABu

Pište mail na:

operator@service.zcu.cz

se žádostí o **nejnovější** licenci MATLABu.

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`*` násobení maticové

```
K =                    L =
    1   2                  5   6
    3   4                  7   8
```

Příklad:

```
M     = K   * L        N     = L   * K
```

(2,2) = (2,2) * (2,2)   (2,2) = (2,2) * (2,2) – první matice má 2 sloupce, tj. stejný počet jako má druhá matice řádků => v obou případech lze násobit maticově:

```
M =                    N =
    19   22                23   34
    43   50                31   46
```

Pro čtvercové matice `K`, `L` platí rovněž `K * L` ≠ `L * K`, pro `K * L` bude jiný výsledek než pro `L * K`.

Maticové násobení není komutativní!!!

Operátory

Operátory pro maticové operace (operace s celými maticemi)

***** násobení maticové

Pro čísla platí:

$$2 * 1 = 1 * 2 = 2$$

Pro matice - násobení jednotkovou maticí

eye – **jednotková** matice (čtvercová matice **s jednotkami na hlavní diagonále**, tj. na úhlopříčce z levého horního do pravého dolního rohu, jinde jsou nuly), nebo obdélníková matice s **jednotkovou submaticí**, např:

```
eye(2)
ans =
     1     0
     0     1
```

```
eye(2,5)
ans =
     1     0     0     0     0
     0     1     0     0     0
```

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`*` násobení maticové

```
F=[3,2,1;4,-5,6;-9,8,-7];
F =
     3     2     1
     4    -5     6
    -9     8    -7
```

```
J=eye(3)
J =
     1     0     0
     0     1     0
     0     0     1
```

```
F * J
ans =
     3     2     1
     4    -5     6
    -9     8    -7
```

```
J * F
ans =
     3     2     1
     4    -5     6
    -9     8    -7
```

Operátory

Operátory pro maticové operace (operace s celými maticemi)

* násobení maticové

```
G = [1,2,3;-4,5,-6]
G =
     1     2     3
    -4     5    -6
```

```
J1 = eye(3)
J1 =
     1     0     0
     0     1     0
     0     0     1
```

```
J2 = eye(2)
J2 =
     1     0
     0     1
```

Aby matice bylo možno násobit, musí být počet sloupců první matice stejný jako počet řádků druhé matice =>

```
G * J1
ans =
     1     2     3
    -4     5    -6
```

```
J2 * G
ans =
     1     2     3
    -4     5    -6
```

Operátory

Operátory pro maticové operace (operace s celými maticemi)

***** násobení maticové

Pro čísla platí:

$$2*\frac{1}{2}=2*2^{-1}=1$$

Pro matice - násobení inverzní maticí

`inv` – **inverzní** matice A^{-1} k dané matici A je taková matice, která po vynásobení s původní maticí dá jednotkovou matici. Inverzní matici lze vytvořit pouze ke čtvercové matici (která je tzv. regulární), např:

```
A=[1,2;3,4]
A =
     1     2
     3     4
```

```
inv(A)
ans =
   -2.00    1.00
    1.50   -0.50
```

```
A * inv(A)
ans =
     1     0
     0     1
```

Operátory

Operátory pro maticové operace (operace s celými maticemi)

* násobení maticové

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

inv(F)
   0.12500  -0.21154  -0.16346
   0.25000   0.11538   0.13462
   0.12500   0.40385   0.22115

F * inv(F)
    1.0000e+00   2.7756e-17   8.3267e-17
    1.1102e-16   1.0000e+00  -1.3878e-16
   -2.4980e-16   2.7756e-17   1.0000e+00
```

tj. jednotková matice:

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

Násobením matice a k ní inverzní matice vznikne **matice jednotková**.

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`*` násobení maticové

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

```
F.'
     3     4    -9
     2    -5     8
     1     6    -7
```

```
F * F.'
    14     8   -18
     8    77  -118
   -18  -118   194
```

```
F.'* F
   106   -86    90
   -86    93   -84
    90   -84    86
```

Násobením matice a k ní transponované matice vznikne matice **symetrická podle hlavní diagonály**.

`F * F.'` ≠ `F.'* F` Maticové násobení není komutativní !!!

Operátory

Operátory pro maticové operace (operace s celými maticemi)

/ maticové dělení (zprava) a **** maticové dělení zleva

2 / 3 - dvě děleno třemi zprava
ans = 0.66667
3 \ 2 - dvě děleno třemi zleva
ans = 0.66667

$$\frac{2}{3} = 2*\frac{1}{3} = 2*3^{-1} = 3^{-1}*2$$

U matic A^{-1} (tj. vlastně „něco jako $\frac{1}{A}$“) značí právě inverzní matici.
inv(A) – inverzní matice

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`/` maticové dělení (zprava) a `\` maticové dělení zleva

```
2 / 3
```
- dvě děleno třemi zprava
```
ans =   0.66667
3 \ 2
```
- dvě děleno třemi zleva
```
ans =   0.66667
```

$$\frac{2}{3} = 2 * \frac{1}{3} = 2 * 3^{-1} = 3^{-1} * 2$$

Pro čísla jde o stejnou operaci násobení, půjde-li však o **dělení matic**, jde o dvě **různé operace** (viz výše násobení matic)

U matic A^{-1} (tj. vlastně „něco jako $\frac{1}{A}$“) značí právě inverzní matici.

`inv(A)` – inverzní matice

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`/` maticové dělení (zpráva) a `\` maticové dělení zleva

A děleno B zprava: `A / B` „odpovídá“ `A * inv(B);`
A děleno B zleva: `B \ A` „odpovídá“ `inv(B) * A;`

B děleno A zprava: `B / A` „odpovídá“ `B * inv(A);`
B děleno A zleva: `A \ B` „odpovídá“ `inv(A) * B;`

Rovnost `A / B = A * inv(B)` nelze použít, není zcela matematicky korektní, je uvedena jen pro představu

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`/` maticové dělení (zprava) a `\` maticové dělení zleva

Příklad - jsou dány matice **U**, **T**:

```
T =                         U =
      1      2                    5      5
      3      4                    7      8
```

```
T / U                          T * inv(U)
ans =                          ans =
    3.0000   -2.0000               3.0000   -2.0000
    2.0000   -1.0000               2.0000   -1.0000
```

```
T \ U                          inv(T) * U
ans =                          ans =
     -3      -4                   -3.0000   -4.0000
      4       5                    4.0000    5.0000
```

Operátory

Operátory pro maticové operace (operace s celými maticemi)

/ maticové dělení (zprava) a **** maticové dělení zleva

Pokračování příkladu:

```
T =
      1      2
      3      4
```

```
U =
      5      5
      7      8
```

```
U / T
ans =
   -1.0000    2.0000
   -2.0000    3.0000
```

```
U * inv(T)
ans =
   -1.0000    2.0000
   -2.0000    3.0000
```

```
U \ T
ans =
    5.0000    4.0000
   -4.0000   -3.0000
```

```
inv(U) * T
ans =
    5.0000    4.0000
   -4.0000   -3.0000
```

Operátory

Operátory pro maticové operace (operace s celými maticemi)

^ mocnina maticová

```
A =
     10     5    70
      2     7     6
      8     9     1
```

```
A^2
ans =
    670    715    800
     82    113    188
    106    112    615
```

Stejný výsledek dostaneme i pomocí maticového násobení

```
A * A
ans =
    670    715    800
     82    113    188
    106    112    615
```

Operátory

Operátory pro maticové operace (operace s celými maticemi)
`^` mocnina maticová

Maticově mocnit lze jen **čtvercovou** matici – nelze obdélníkovou – probíhá stejná operace jako při maticovém násobení, první matice musí mít stejný počet sloupců jako má druhá matice řádků, tj. matice pro umocnění musí být čtvercová (musí mít stejný počet řádků a sloupců)

`M = A * A`
$(n,n) = (n,n) * (n,n)$

`A` – čtvercová matice
n –počet řádků a sloupců matice `A`, výsledná matice je stejných rozměrů

`O = B * B`
$(???) = (m,n) * (m,n)$

`B` – obdélníková matice
m,n –počet řádků a sloupců matice `B`, nelze násobit, $m \neq n$

Operátory

Operátory pro maticové operace (operace s celými maticemi)

^ mocnina maticová

Příklady:

```
A = [1 2;3 4]
A =
     1     2
     3     4
```

```
A^2
ans =
     7    10
    15    22
```

```
A.^2
ans =
     1     4
     9    16
```

```
B = [1 2;3 4;5,6]
B =
     1     2
     3     4
     5     6
```

```
B^2
```
nelze – matice musí být čtvercová!!!

```
B.^2
ans =
     1     4
     9    16
    25    36
```

Operátory

Odmocnina

`sqrt(x)` – druhá odmocnina z *x*, tj. $\sqrt{x}$

Např.

$\sqrt{4}$

```
sqrt(4)
ans =  2
```

Jak jinak na odmocninu (a nejen na druhou)? Platí: $\sqrt[n]{x} = x^{\frac{1}{n}}$

to je tedy: **`x.^(1/n)`**.

Příklad: $\sqrt{4} = 4^{\frac{1}{2}} = 2$

```
4 .^ (1/2)
ans =
      2
```

Příklad: $\sqrt[18]{237} = 237^{\frac{1}{18}} = 1{,}355$

```
237 .^ (1/18)
ans =
      1.3550
```

Operátory

Příklad: třetí odmocnina **z prvků** matice $\mathbf{V} = \begin{bmatrix} 1 & 8 & 27 \\ 64 & 100 & 1000 \end{bmatrix}$

tj. $\begin{bmatrix} \sqrt[3]{1} & \sqrt[3]{8} & \sqrt[3]{27} \\ \sqrt[3]{64} & \sqrt[3]{100} & \sqrt[3]{1000} \end{bmatrix}$

```
V = [1 8 27; 64,100,1000]
V =
      1         8        27
     64       100      1000
```

`V.^(1/3)` - operace prvek po prvku

```
ans =
    1.0000    2.0000    3.0000
    4.0000    4.6416   10.0000
```

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`'` transpozice matice a `.'` překlopení podle hlavní diagonály

Příklad:

```
W = [5,6,7;4,3,2]

W =
     5   6   7
     4   3   2
```

```
W.'
ans =
        5     4
        6     3
        7     2
```

```
W'
ans =
        5     4
        6     3
        7     2
```

Transpozice je překlopení matice kolem hlavní diagonály.
V našem příkladu se z matice o 2 řádcích a 3 sloupcích stane matice o 3 řádcích a 2 sloupcích - dojde k vzájemné výměně řádků a sloupců.
Stejný výsledek dostaneme pomocí příkazu `W.'` i pomocí příkazu `W'`, protože matice `W` je naplněna **reálnými čísly**.

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`'` transpozice matice a `.'` překlopení podle hlavní diagonály

Příklad:

```
W = [5,6,7;4,3,2]

W =
     5   6   7
     4   3   2
```

```
W.'
ans =
     5   4
     6   3
     7   2
```

```
W'
ans =
     5   4
     6   3
     7   2
```

Transpozice je překlopení matice kolem hlavní diagonály.
V našem příkladu se z matice o 2 řádcích a 3 sloupcích stane matice o 3 řádcích a 2 sloupcích - dojde k vzájemné výměně řádků a sloupců.
Stejný výsledek dostaneme pomocí příkazu `W.'` i pomocí příkazu `W'`, protože matice `W` je naplněna **reálnými čísly**.

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`'` transpozice matice a `.'` překlopení podle hlavní diagonály

Příklad:

```
K = [1+3i,6,-7+5i;4-8i,-3-9i,2i];
```

Matice `K` je naplněna **komplexními čísly**. Abychom získali transponovanou matici, musíme použít příkaz `K.'` (prostá transpozice matice).

Po provedení příkazu `K'` dostaneme tzv. adjugovanou matici, tzn. matici transponovanou, kterou tvoří ji čísla **komplexně sdružená**, tj. u **imaginárních** částí mají **opačná** znaménka.

Operátory

Operátory pro maticové operace (operace s celými maticemi)

`'` transpozice matice a `.'` překlopení podle hlavní diagonály

Pokračování příkladu:

```
K = [1+3i,6,-7+5i;4-8i,-3-9i,2i];
K =
    1 + 3i    6 + 0i   -7 + 5i
    4 - 8i   -3 - 9i    0 + 2i
```

```
K.'
ans =
    1 + 3i    4 - 8i
    6 + 0i   -3 - 9i
   -7 + 5i    0 + 2i
```

- prostá transpozice

```
K'
ans =
    1 - 3i    4 + 8i
    6 - 0i   -3 + 9i
   -7 - 5i    0 - 2i
```

- transponovaná matice s **komplexně sdruženými** čísly

Maticové operace s vektory

```
a = [23,67,56,43,31];
b = [8,3,0,23,10];
```

```
a.'
ans =
    23
    67
    56
    43
    31
```

```
b.'
ans =
     8
     3
     0
    23
    10
```

`a .* b` – násobení **prvek po prvku**

```
ans =
   184   201     0   989   310
```

`a  * b` – **maticové** násobení vektorů `a`, `b`,

(1,5) * (1,5) – počty řádků a sloupců

nelze násobit, rozměry matic musí souhlasit, 5 ≠ 1

`a.' * b.'` – **maticové** násobení transponovaných vektorů `a.',b.'`

(5,1) * (5,1)

nelze násobit, rozměry matic musí souhlasit, 1 ≠ 5

Maticové operace s vektory

```
a = [23,67,56,43,31];
b = [8,3,0,23,10];
```

```
a .* b
ans =
    184
```

– násoben

```
a.'
ans =
    23
     7
     6
    43
    31
```

```
b.'
ans =
     8
     3
     0
    23
    10
```

Při maticovém násobení „ * “ počet sloupců první matice (zde vektoru) musí souhlasit počtem řádků matice (zde vektoru).

`a * b` – **maticové** násobení vektorů **a**, **b**,

(1,5) * (1,5) – počty řádků a sloupců

nelze násobit, rozměry matic musí souhlasit, 5 ≠ 1

`a.' * b.'` – **maticové** násobení transponovaných vektorů **a.'**, **b.'**

(5,1) * (5,1)

nelze násobit, rozměry matic musí souhlasit, 1 ≠ 5

Maticové operace s vektory

```
a = [23,67,56,43,31]; b = [8,3,0,23,10];
```

```
a.'
ans =
    23
    67
    56
    43
    31
```

```
b.'
ans =
     8
     3
     0
    23
    10
```

`c = a * b.'` – **maticové** násobení

(1,1) = (1,5) * (5,1) – počty řádků a sloupců

```
c =
          1684
```

– lze násobit maticově, 5 = 5, `a` má 5 sloupců a `b.'` má 5 řádek (je to sloupec), výsledek má 1 řádek a 1 sloupec

`d = a.' * b` – **maticové** násobení

(5,5) = (5,1) * (1,5) – počty řádků a sloupců

```
d =
     184      69       0     529     230
     536     201       0    1541     670
     448     168       0    1288     560
     344     129       0     989     430
     248      93       0     713     310
```

– lze násobit maticově, 1 = 1, `a.'` má 1 sloupec a `b` má 1 řádek, výsledek má 5 řádků a 5 sloupců

Operace s vektory a maticemi

```
a = [1,2,3,4,5];
b = [6,7,8,9,10];
```

Násobení **prvek po prvku** – změny

% verze R2015a		% verze R2017b	
`a   .* b`	– lze	`a   .* b`	– lze
`a.'.* b.'`	– lze	`a.'.* b.'`	– lze
`a.'.* b`	– **nelze**	`a.'.* b`	– lze
`a   .* b.'`	– **nelze**	`a   .* b.'`	– lze

Násobení **maticové** se nemění.

`a   * b` – **maticové** násobení řádkových vektorů `a`, `b`,

(1,5) * (1,5) – počty řádků a sloupců

nelze násobit, rozměry matic musí souhlasit, 5 $\neq$ 1

`a.' * b.'` – **maticové** násobení (transponovaných) sloupcových vektorů `a.'`, `b.'`

(5,1) * (5,1)

nelze násobit, rozměry matic musí souhlasit, 1 $\neq$ 5

Operace s vektory a maticemi

```
a = [1,2,3,4,5];
b = [6,7,8,9,10];
```

Násobení **prvek po prvku** – změny

```
% verze R2015a
a   .*
a.
```

```
% verze R2017b
      b
```

– lze
– lze
– lze
– lze

Při maticovém násobení „ * “ počet sloupců první matice (zde vektoru) musí souhlasit počtem řádků matice (zde vektoru).

Násobení

`a * b` – **maticové** násobení vektorů `a`, `b`,
(1,5) * (1,5) – počty řádků a sloupců
nelze násobit, rozměry matic musí souhlasit, 5 ≠ 1

`a.' * b.'` – **maticové** násobení (transponovaných) sloupcových vektorů `a.'`,`b.'`
(5,1) * (5,1)
nelze násobit, rozměry matic musí souhlasit, 1 ≠ 5

Operace s vektory a maticemi

```
a = [1,2,3,4,5]; b = [6,7,8,9,10];
```

```
a.'
ans =
     1
     2
     3
     4
     5
```

`c = a * b.'` – **maticové** násobení

(1,1) = (1,5) * (5,1) – počty řádků a sloupců

```
c =
       130
```

– lze násobit maticově, 5 = 5,

`a` má 5 sloupců a `b.'` má 5 řádek (je to sloupec), výsledek má 1 řádek a 1 sloupec

```
b.'
ans =
     6
     7
     8
     9
    10
```

`d = a.' * b` – **maticové** násobení

(5,5) = (5,1) * (1,5) – počty řádků a sloupců

```
d =
     6     7     8     9    10
    12    14    16    18    20
    18    21    24    27    30
    24    28    32    36    40
    30    35    40    45    50
```

– lze násobit maticově, 1 = 1, `a.'` má 1 sloupec a `b` má 1 řádek, výsledek má 5 řádků a 5 sloupců

Operace s vektory a maticemi

```
M = [1,2,3;4,5,6]; n = [7;8];
```

```
M =
   1   2   3
   4   5   6

n =
       7
       8
```

`O     = n.'  * M` – **maticové** násobení

(1,3) = (1,2) * (2,3) – počty řádků a sloupců

```
O =
      39      54      69
```

– lze násobit maticově, 2 = 2, `n.'` má 2 sloupce a `M` má 2 řádky, výsledek má 1 řádek a 3 sloupce

`P     = M   * n` – **maticové** násobení

(?,?) = (2,3) * (2,1) – počty řádků a sloupců

```
Error using  *
Inner matrix dimensions must agree.
```

– **nelze** násobit, rozměry matic musí souhlasit, 3 ≠ 2

`Q     = n   * M`

(?,?) = (2,1) * (2,3)

```
Error using  *
```

`R     = M  *  n.'`

(?,?) = (2,3) * (1,2)

```
Error using  *
```

Výpočet soustavy lineárních rovnic

Příklad: Řešte soustavu lineárních algebraických rovnic

$$
\begin{aligned}
10x_1 + 5x_2 + 70x_3 + 5x_4 + 5x_5 &= 275 \\
2x_1 + 7x_2 + 6x_3 + 9x_4 + 6x_5 &= 100 \\
8x_1 + 9x_2 + x_3 + 45x_4 + 22x_5 &= 319 \\
3x_1 + 12x_2 + 6x_3 + 8x_4 + 33x_5 &= 242 \\
20x_1 + 2x_2 + 20x_3 + 96x_4 + 98x_5 &= 958
\end{aligned}
$$

$$\mathbf{A} * x = \mathbf{b}$$

```
A=[10, 5,70, 5, 5;...        x=[ x1          b=[275;...
    2, 7, 6, 9, 6;...            x2             100;...
    8, 9, 1,45,22;...            x3             319;...
    3,12, 6, 8,33;...            x4             242;...
   20, 2,20,96,98]               x5]            958]
```

kde `A` je **matice koeficientů soustavy** lineárních rovnic,
`x` je sloupcový **vektor neznámých** (tj. `[x1;x2;x3;x4;x5]`),
`b=[275;100;319;242;958]` je sloupcový **vektor pravých stran** (tj. vše, co je vpravo od =)

Výpočet soustavy lineárních rovnic

Platí:

$$A * x = b$$

$$A^{-1} * A * x = A^{-1} * b$$

$$J \quad * x = A^{-1} * b$$

$$x = A^{-1} * b\,,$$

kde J je **jednotková matice**.

- podle tohoto vztahu počítáme

```
x = inv(A) * b          % * operace maticové násobení
```

(5,1) = (5,5) * (5,1) – počty řádků a sloupců souhlasí

- nebo pomocí maticového dělení zleva

```
x = A \ b          % \ operace maticové dělení zleva
```

Výpočet soustavy lineárních rovnic

Platí:

$$A * x = b$$

$$A^{-1} * A * x = A^{-1} * b$$

$$J \quad * x = A^{-1} * b$$

$$x = A^{-1} * b,$$

Víme, že:

$$A^{-1} * A = J$$

$$J * x = x$$

kde J je **jednotková matice**.

- podle tohoto vztahu počítáme

```
x = inv(A) * b        % * operace maticové násobení
```

(5,1) = (5,5) * (5,1) – počty řádků a sloupců souhlasí

- nebo pomocí maticového dělení zleva

```
x = A \ b        % \ operace maticové dělení zleva
```

Výpočet soustavy lineárních rovnic

```
A=[10,5,70,5,5;2,7,6,9,6;8,9,1,45,22;...
    3,12,6,8,33; 20, 2,20,96,98];
b=[275;100;319;242;958];
```

Rychlejší výpočet

Řešení pomocí výpočtu inverzní matice a maticového násobení

```
x = inv(A) * b
x =
    1.0000
    2.0000
    3.0000
    4.0000
    5.0000
```

Řešení pomocí maticového dělení zleva

```
x = A \ b
x =
    1.0000
    2.0000
    3.0000
    4.0000
    5.0000
```

Výpočet soustavy lineárních rovnic

Pozor na nepřesnosti způsobené **zaokrouhlováním**:
(`round` zaokrouhluje **na nejbližší celé číslo**)

```
x = round(inv(A)) * b
x =
   361
     0
     0
     0
     0
```

A to je opravdu špatně....

```
x = round(inv(A).*1000)./1000 * b
x =
   0.32400
   2.41100
   3.03000
   3.58600
   5.20400
```

- zaokrouhlíme **na tisíciny**

Stále velmi nepřesné...

Zaokrouhlování

round – zaokrouhlení na nejbližší **celé číslo**

floor – zaokrouhlení na nejbližší **nižší** celé číslo, zaokrouhlení **dolů**

ceil – zaokrouhlení na nejbližší **vyšší** celé číslo, zaokrouhlení **nahoru**

fix – zaokrouhlení na nejbližší celá čísla **směrem k nule**.

Příklad:

```
a=[-0.954, -1.231, 5.241, 6.896];
```

```
floor(a)
ans =
    -1   -2    5    6
```

```
round(a)
ans =
    -1   -1    5    7
```

```
ceil(a)
ans =
     0   -1    6    7
```

```
fix(a)
ans =
     0   -1    5    6
```