

A0B17MTB – Matlab

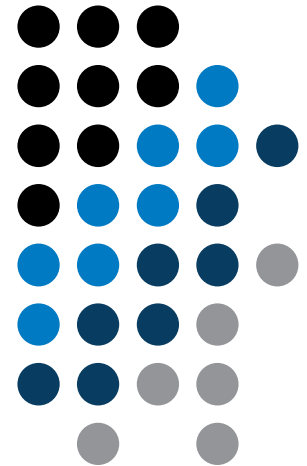
Část #1



Miloslav Čapek
miloslav.capek@fel.cvut.cz

Filip Kozák, Viktor Adler

Katedra elektromagnetického pole
B2-626, Dejvice



Naučíte se ...

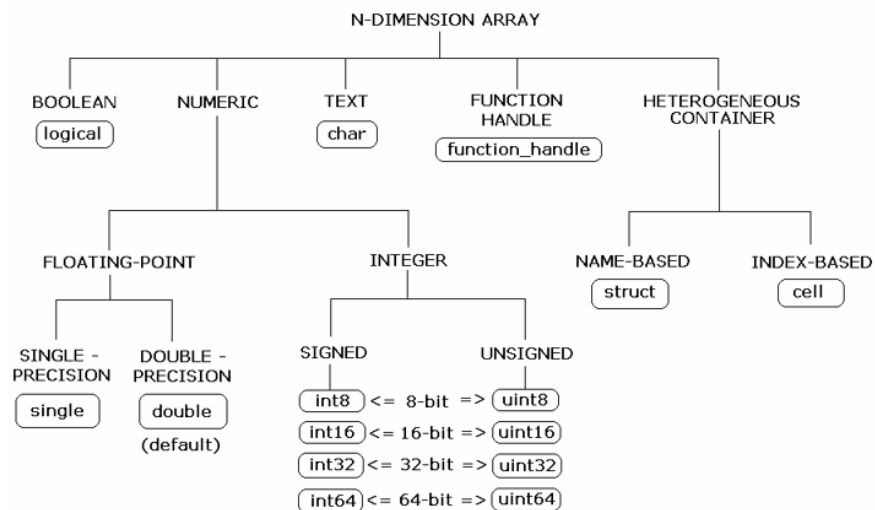
Skaláry, vektory, matice (třída numeric)

Maticové operace

Command Window, Command History

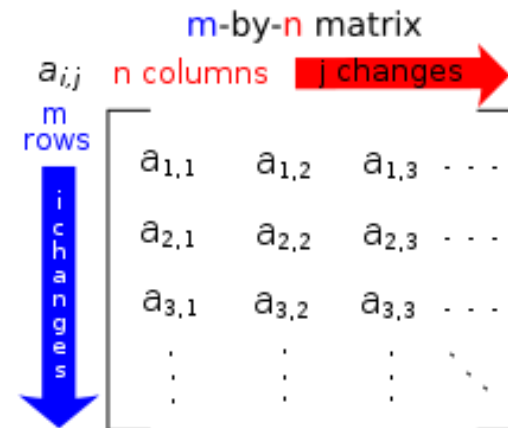
Ukládání a načítání proměnných

Příklady



Matice v Matlabu

- matice je základní datová struktura v Matlabu
- podle velikosti mluvíme o:
 - 1×1 (skaláru)
 - $M \times 1$, příp. $1 \times N$ (vektoru)
 - $M \times N$ (matici)
 - pole (vícerozměrné matice) $M \times N \times P \times Q \times R \times \dots$
- mohou být komplexní
- mohou být vyplněny i textem (pozor na velikost)



Zadávání matic

- obecně lze využít následující techniky:
 - zadání přímo výčtem prvků (vhodné pouze pro menší matice)
 - použití dvojtečky „:“ pro definici prvků řady
 - generování pomocí vestavěných funkcí
 - vytvoření matic v m-souborech
 - import a export z externích souborů (.mat, .txt, .xls)

Zadávání matic výčtem prvků

- vyzkoušejte si následující příkazy na zadávání matic pomocí výčtu jednotlivých prvků
 - vhodné pouze pro malé matice

```
>> a1 = -1
>> a2 = [-1] % nadbytečné
```

$$a_1 = a_2 = -1$$

$$\mathbf{v}_1 = (-1 \ 0 \ 1)$$

$$\mathbf{v}_2 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

```
>> v1 = [-1 0 1]
>> v2 = [-1;0;1]
```

```
>> M1 = [-1 0 1; -2 0 2]
>> M2 = [-1 -2; 0 0; 1 2]
>> M3 = [[-1 -2]; [0 0]] % nadbytečné
```

$$\mathbf{M}_1 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \end{pmatrix}, \quad \mathbf{M}_2 = \begin{pmatrix} -1 & -2 \\ 0 & 0 \\ 1 & 2 \end{pmatrix}, \quad \mathbf{M}_3 = \begin{pmatrix} -1 & -2 \\ 0 & 0 \end{pmatrix}$$

Zadávání matic výčtem prvků

90 s ↑

- vytvořte matice:
 - hodnoty v matici jsou uvozeny hranatými závorkami []
 - středník „;“ odděluje jednotlivé řádky matice

$$\mathbf{A} = \begin{pmatrix} -1 & 1 \\ 1 & -2 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Zadávání matic

120 s ↑

- středník za příkazem potlačí výpis do Command Window

```
>> a = 1  
>> b = 5;
```

- čárka umožňuje řadit více příkazů za sebe

```
>> a = 1, b = 5  
>> a = 1; b = 5;
```

- poznámka: do Matlabu můžete kopírovat příklady ze slajdů vč. „>>“

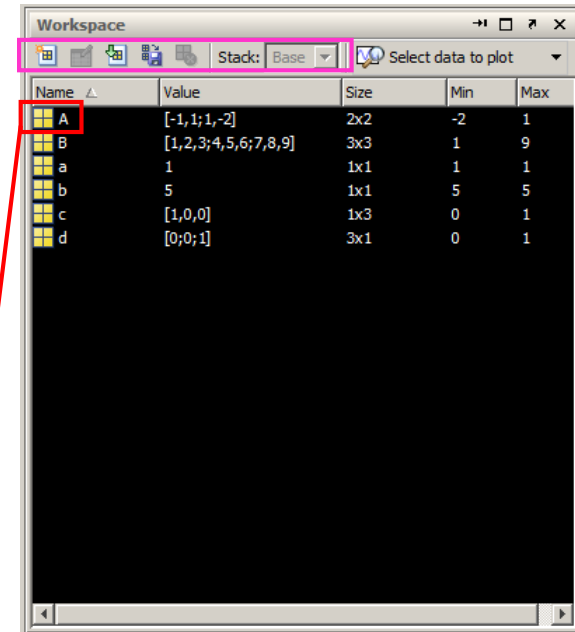
- „ležatý“ vs. „stojatý“ vektor

```
>> c = [1 0 0]  
>> d = [0;0;1]
```

$$\mathbf{c} = (1 \ 0 \ 0) \quad \mathbf{d} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Okno Workspace

- nové proměnné
- mazání / úprava existujících
- ukládání / načítání
- přehled o velikosti, prvních atp.
 - lze nastavit další informace
- možnost rychlého vykreslení dat
- všechny operace lze provést i pomocí funkcí Matlabu, které se později naučíme, např. `min`, `max`, `size`, `length`



	1	2	3	4	5	6	7
1	-1	1					
2	1	-2					
3							
4							
5							
6							
7							

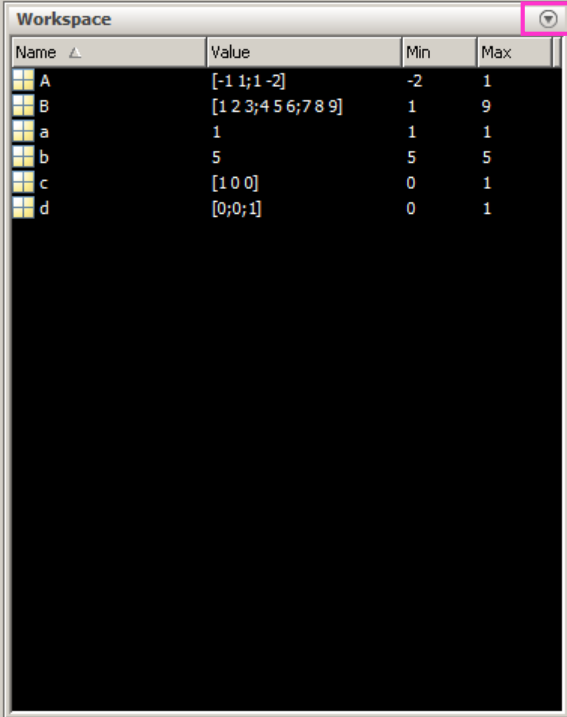
Okno Workspace

- pracovní prostor nyní obsahuje (mj.) A, B, a, b, c, d
 - zobrazuje všechny proměnné v základním pracovním prostoru

- proměnná `ans` obsahuje poslední výsledek
 - lze jí užívat i zpětně
 - je však přepisována!

```
>> 2*2, ans^2
```

- zkuste editovat proměnné A, a
 - přímým příkazem
 - změnou hodnoty v okně Workspace
- zkuste vymazat proměnné B, c



Name	Value	Min	Max
A	[-1 1;1 -2]	-2	1
B	[1 2 3;4 5 6;7 8 9]	1	9
a	1	1	1
b	5	5	5
c	[1 0 0]	0	1
d	[0;0;1]	0	1

Základní matematické operátory

- dělíme na:
 - aritmetické
 - maticové
 - vektorové
 - relační
 - logické

+	sčítání
-	odčítání
*	násobení
^	mocnění
'	transpozice
\	dělení zprava
/	dělení zleva
.	tečková notace

- další operace pomocí funkcí Matlabu
 - např. komplexní sdružení,
 - suma, determinant, odmocnina
 - a stovky dalších funkcí ...

Přednost operátorů v Matlabu

- podle následující tabulky
 - viz Matlab → Language Fundamentals → Operators and Elementary Operations → Arithmetic

1	závorky	()					
2	transpozice, umocňování	'	.'	^	.'	^	
3	unární plus, mínus, negace	+	-	~			
4	násobení, dělení	*	.*	/	\	./	.\
5	sčítání, odečítání	+	-				
6	operace s dvojtečkou	:					
7	relační operátory	<	>	<=	>=	==	~=
8	logický AND (po složkách)	&					
9	logický OR (po složkách)						
10	logický AND (zkrácené vyhodnocování)	&&					
11	logický OR (zkrácené vyhodnocování)						

Základní matematické operátory

200 s ↑

- zkuste si zadat následující příkazy
 - pro desetinná čísla začínající nulou lze tuto vynechat (nedoporučuje se!)

```
>> a3 = -2/4  
>> a4 = -0.5  
>> a5 = -.5
```

- jaký je rozdíl mezi a3, a4, a5?
- vysvětlete rozdíl mezi $a2/a3*a4$ a $a2/(a3*a4)$
 - pozor na přednost operátorů (více dále)

```
>> 3*5*6  
>> a1 = 15  
>> a2 = 10;  
>> a2/a3  
>> a2/a3*a4  
>> a2/(a3*a4)
```

- ověřte pravidla přednosti operátorů z tabulky na předcházejícím slajdu

Dlouhé příkazy v Matlabu

120 s ↑

- pro přehlednost je vhodné vhodně strukturovat příkazy:
 - další řádka: SHIFT+ENTER

```
>> A = [1 1 1]; B = [2 2 2]; % SHIFT+ENTER  
C = [2 3 2];
```

- notace tří teček
 - pro pokračování stejného příkazu na další řádce
 - srovnejte výsledky:

```
>> A = [1 1 ...  
2 3]
```

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

Základní matematické funkce

- v Matlabu dělíme obecně na:
 - skalární
 - funkce je vyhodnocena nad jednotlivými prvky matice
 - na příklad: `sin`, `sqrt`, `log`, `factorial`
 - vektorové
 - funkce je vyhodnocena nad vektory / sloupci v matici
 - na příklad: `sum`, `max`
 - maticové
 - funkce je vyhodnocena nad celou maticí
 - na příklad: `det`, `trace`

Základní matematické funkce #1

600 s ↑

- s využitím nápovědy v Matlabu vypočítejte $a \sin^2(\alpha) + a \cos^2(\alpha) - a$
 - využijte konkrétních hodnot

- ověřte větu o součtu logaritmů $\log_{10}(a) + \log_{10}(b) - \log_{10}(ab) = 0$

- najděte součet všech prvků matice v jednotlivých řádcích

$$\mathbf{T} = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 6 & 7 & 8 & 9 \\ 0.2 & 0.3 & 0.4 & 0.5 \end{pmatrix}$$

Základní matematické funkce #2

600 s ↑

- předpokládejte vektory \mathbf{u} , \mathbf{v} ve tvaru: $\mathbf{u} = (1 \ 2 \ 3)$, $\mathbf{v} = (3 \ 2 \ 1)$
 - vypočtete

$$\begin{aligned} &\mathbf{u}\mathbf{v}^T, && \mathbf{v}\mathbf{u}^T, \\ &\mathbf{v}^T\mathbf{u}, && \mathbf{u}^T\mathbf{v}, \\ &\mathbf{u} \cdot \mathbf{v}, && \mathbf{u} \times \mathbf{v}, \end{aligned}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

wikipedia.org

- k výpočtu je potřeba:
 - funkce transpozice matice (*transpose of a matrix*)
 - skalární součin (*dot / scalar product*)
 - vektorový součin (*cross product*)
- co je výsledkem jednotlivých operací?

$$\begin{aligned} &10, && 10, \\ &\begin{pmatrix} 3 & 6 & 9 \\ 2 & 4 & 6 \\ 1 & 2 & 3 \end{pmatrix}, && \begin{pmatrix} 3 & 2 & 1 \\ 6 & 4 & 2 \\ 9 & 6 & 3 \end{pmatrix}, \\ &10, && (-4 \ 8 \ -4). \end{aligned}$$

Základní matematické funkce #3

420 s



- využijte následující příkaz a zaokrouhlete výsledné číslo r , tak aby bylo zaokrouhleno směrem k
 - (a) nejbližšímu celému číslu
 - (b) nejbližšímu vyššímu celému číslu
 - (c) nejbližšímu nižšímu celému číslu
 - (d) nule
 - (e) k nule na 2 desetinná čísla
- najděte zbytek po dělení r číslem 0.1
 - *modulus vs. remainder after division*

pozn.: jedna z funkcí se jmenuje round

Maticové dělení v Matlabu

- maticová operace
- rozlišujeme dva případy – dělení zleva („\“) a dělení zprava („/“)
- A je invertibilní (regulární), b je řádkový (sloupcový) vektor

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

soustavu

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

$$\mathbf{x} = A^{-1}\mathbf{b}$$

řešíme

$$\mathbf{x} = \mathbf{b}A^{-1}$$

```
>> x = A \ b
```

```
>> x = b / A
```

Základní matematické funkce #4

500 s ↑

- najděte součet prvků na diagonále (stopu) matice \mathbf{T} , která je generována s normálním rozložením kolem hodnoty 10 se směrodatnou odchylkou 4

```
>> T = 10 + 4*randn(7,7);
```

- najděte determinant matice

$$\mathbf{U} = \begin{pmatrix} 1 & 2 & \frac{17}{81} \\ 0 & 2 & 0 \\ 0 & -2 & -1 \end{pmatrix}$$

- řešte soustavu rovnic

$$x_1 + 2x_2 + 3x_3 = 6$$

$$4x_1 + 5x_2 + 6x_3 = 15$$

$$7x_1 + 8x_2 + x_3 = 16$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

Příkazy v Matlabu

- Matlab je **cAsE sEnSiTiVe**
 - platí krom drobných výjimek (vlastnosti grafických objektů atp.) takřka všude
 - pozor na překlepy a názvy proměnných (viz dále)
 - nové verze Matlabu již nabízí možné alternativy

```
>> AA = [1 1 1]
>> Aa
```

- pozor na záměnu syntaxe s Mathematicou
 - následující syntaxe je chybná v Matlabu i Mathematice:

```
>> Sin(pi/2) % funkce začínají malými písmeny
>> cos[pi/3] % argument funkcí je v ()
```

Předdefinované hodnoty v Matlabu

- Matlab zná několik předdefinovaných hodnot
 - `eps` – přesnost, s jakou Matlab pracuje (single / double precision)
 - `ans` – most recent answer
 - `NaN` – not a number (každý algeb. výraz s `NaN` končí zase `NaNem`)
 - pro některé účely lze `NaN` s výhodou využít
 - `Inf` – infinite number (s proměnnou `Inf` lze dále počítat :))
 - pozor na šíření `Inf` hodnot ve Vašem kódu (používejte jen povolené operace)
 - `i`, `j` – komplexní jednotka
 - ve všech případech jde de facto o funkce (bez vstupních argumentů)
- podívejte se na výsledky příkazů:


```

>> t1 = 10/0      % t1 = Inf
>> t2 = 0/0      % t2 = NaN
>> t3 = t1*5     % t3 = Inf
>> t4 = t1 + t2  % t4 = NaN
      
```
- `pi`, `intmin`, `intmax`, `realmin`, `realmax`, ... (funkce)

Workspace – mazání výstupu #1

- pokud je potřeba vyčistit (vymazat) příkazové okno, využijeme

```
>> home % vrátí prompt (>>) doleva nahoru  
>> clc % vymaže Command Window
```

- vyzkoušejte a porovnejte

```
Command Window  
>> a2 = 10  
a3 = -2/4  
a2/a3  
  
a2 =  
    10  
  
a3 =  
   -0.5000  
  
ans =  
   -20  
fx >> |
```

```
>> home
```

```
>> clc
```

```
Command Window  
fx >>
```

Workspace – mazání výstupu #2

- chceme-li vymazat proměnnou (i více), využijeme příkaz `clear`

```
>> clear           % vymaže celý Workspace  
>> clear XX       % vymaže proměnnou XX  
>> clear XX YY    % vymaže proměnné XX a YY  
>> clear z*       % vymaže vše co začíná na z
```

- příkaz `clear` má mnoho dalších variant (pro handle grafiku, I/O vstup/výstup)
- **zkuste** si smazat vybrané proměnné z pracovního prostoru

Workspace – mazání výstupu #3

- občas chceme vymazat všechny proměnné až na jednu (či několik málo), potom:

```
>> clearvars -except a3 % vymaže vše krom proměnné a3
```

- pro další informace `doc clear`, `doc clearvars`

Okno Command History

- ukládá všechny příkazy z řádky
- lze je vyvolat zpět (↑ popř. ↓)
- množinu příkazů lze filtrovat pomocí znaku / znaků
 - např. `>> A = [` + ↑
- lze vybrat oblast / všechny příkazy, ty poté kopírovat do schránky
 - SHIFT / CTRL / CTRL+A → CTRL+C
- později budeme pracovat zejm. se skripty a funkcemi
 - ⇒ příkazy budou uchovány tam

```

Command History
U = [1 2 17/81; 0 2 0; ...
0 -2 -1];
det(U)
clear,clc
T = 10 + 4*randn(10,10);
trace(T)
U = [1 2 17/81; 0 2 0; ...
0 -2 -1];
det(U)
A = [1 2 3; 4 5
b = [6 15 16]';
x = inv(A)*b;
x = A \ b;
t1 = 10/0 % t
t2 = 0/0 % t
t3 = t1*5 % t
t4 = t1 + t2 % t
home % vrátí pro
clc % vymaže Co
  
```

Ukládání a načítání proměnných

- existující proměnné z Matlab Workspace lze uložit na disk

```
>> save % uloží všechny proměnné do matlab.mat v akt. adresáři  
>> save ukoll % uloží všechny proměnné do ukoll.mat  
>> save ukoll a b c % uloží do ukoll.mat proměnné „a“, „b“ a „c“
```

- kliknutí do Command Window / Command History a CTRL+S
- načtení probíhá obdobně
 - příp. přetáhneme soubor z Current Folder do Command Window

```
>> load % načte soubor matlab.mat z aktuálního adresáře  
>> load ukoll % načte všechny proměnné z ukoll.mat  
>> load ukoll a b c % načte z ukoll.mat proměnné „a“, „b“ a „c“
```

Ukládání historie a proměnných

180 s ↑

- uložte si dnes používané příkazy z `Command History`
 - využijte `*.txt` soubor
- do proměnné `Data.mat` uložte všechny proměnné z `Workspace`

- zkuste si uložit pouze vybrané proměnné

- vymažte `Workspace` a načtěte do Matlabu výše uvedené soubory

- jak ukládání, tak načítání lze provést pomocí myši!!

Struktura mat souboru

- mat soubory verze 7.3 mají podobu HDF5 souboru
 - HDF = Hierarchical Data Format
 - umožňují uložit i proměnnou větší než 2GB (pro 64-bitový systém)
 - jedná se o vědecký formát na ukládání dat
- pro některé aplikace je vhodnější pracovat přímo s HDF
 - rychlejší
 - lze přímo definovat strukturu souboru a dat v něm uložených
 - k dispozici (Matlab) *High-Level* funkce, i (HDF) *Low-Level* funkce
- více viz:
 - MATLAB → Data and File Management → Data Import and Export → Scientific Data

Jména proměnných #1

- maximálně 63 znaků začínajících písmenem
 - povolené je v názvu podtržítka „_“ (ne na začátku!)
 - není povoleno užití „:“, „-“ a další
- skaláry a pomocné proměnné malými písmeny ($a = 17.59;$)
- matice zpravidla velkým prvním písmenem ($A = [\dots]$)
 - velké matice po použití smažte (`clear ...`, šetříte paměť)
- iterační proměnné, hodnoty z cyklů zpravidla m, n, k atp.
 - je vhodné se vyhnout i, j (komplexní jednotka)
- volte jména tak, aby alespoň částečně vysvětlovala svůj účel
- vyhněte se, pokud možno, osamocenému písmenu l (plete se s 1) a předdefinovaným proměnným v prostředí Matlab

Jména proměnných #2

- překročení délky jména proměnné:

```
>> a01234567890123456789012345678901234567890123456789012345678901234567890123456789 = 10
Warning: 'a01234567890123456789012345678901234567890123456789012345678901234567890123456789'
exceeds the MATLAB maximum name length of 63 characters and will be truncated to
'a0123456789012345678901234567890123456789012345678901'.

a01234567890123456789012345678901234567890123456789012345678901 =

10
```

Jména proměnných #3

- příklady korektních názvů proměnných

```
a, A, b, c, x1, x2, M_12, test1, matice_A, fx, fX
```

- příklady chybných názvů

```
lprom      % začíná číslem (není v Matlabu možné)
matice A   % obsahuje mezeru
koef.a     % pokud je koef typu struct, je možné
Test-1     % algebraicky výraz: ans = Test - 1
f(y)       % ukážeme si, že má smysl u symbolic. výpočtu
```

- příklady přípustných čísel v Matlabu
 - pozor na omylem vložené mezery za exponent a imag. jednotku

```
3, -66, +0.0015, .015, 1.6025e-10, 3i, 3.17e5i, -3.51j
```

Probrané funkce

<code>sin, cos</code>	trigonometrické funkce	
<code>sqrt</code>	druhá odmocnina	
<code>max</code>	vrací největší prvek ve sloupci matice, případně max. prvek z vektoru	•
<code>sum</code>	sumace prvků ve sloupci matice, případně součet prvků vektoru	•
<code>log, log10</code>	přirozený logaritmus, logaritmus o základu 10	
<code>factorial</code>	faktoriál	
<code>det, trace</code>	determinant (čtvercové) matice, stopa (čtvercové) matice	
<code>transpose</code>	transpozice	
<code>dot, cross</code>	skalární součin, vektorový součin	•
<code>inv</code>	inverze matice	
<code>round, ceil, floor, fix</code>	zaokrouhlování	
<code>rem</code>	zbytek po dělení	
<code>rand, randn</code>	generace náhodných čísel	
<code>save, load</code>	ukládání, načítání proměnných	•
<code>clear, clearvars</code>	mazání proměnných a funkcí, mazání pouze proměnných	•
<code>home, clc</code>	posune příkazovou řádku, vymaže výstup	
<code>ans, eps</code>	vrací poslední odpověď, numerická přesnost Matlabu	•

Cvičení #1

180 s ↑

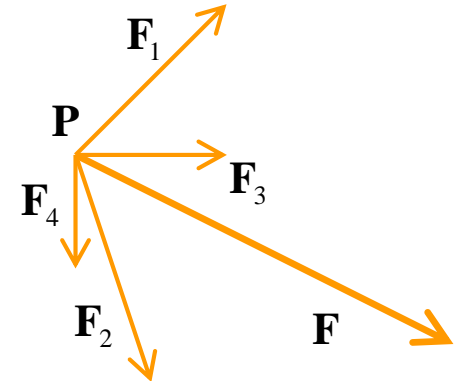
- v bodě **P** byly lokalizovány síly:

$$\mathbf{F}_1 = (2 \quad 2)$$

$$\mathbf{F}_3 = (2 \quad 0)$$

$$\mathbf{F}_2 = (1 \quad -3)$$

$$\mathbf{F}_4 = (0 \quad -1.5)$$



- jakým směrem působí výslednice sil **F**?

- převed'te výsledný vektor na jednotkový

$$\mathbf{n}_F = \frac{\mathbf{F}}{|\mathbf{F}|} = \frac{\mathbf{F}}{\sqrt{F_x^2 + F_y^2 + F_z^2}}$$

- zadejte následující příkazy:

```
>> clear, clc;  
>> w1 = [1 2 3], w2 = [-2 -3 -4] ',  
>> w3 = [-2; -3; -4],  
>> w4 = w2 - w3, w5 = w2 - w1
```

- porovnejte rozdíly
- v čem spočívá chyba v případě výpočtu w5?

- vyzkoušete si rovněž

```
>> w1*3, w1 - 3,  
>> w1 + [5 5 5],  
>> w6 = 5*w1 - [3 5 6] - w2,
```

- vypočtete normu (velikost) vektoru w1
- vyzkoušejte si více možností

$$\mathbf{w}_1 = \frac{\mathbf{w}_1}{|\mathbf{w}_1|}$$

- jak je potřeba výpočet modifikovat v případě komplexního vektoru?

Cvičení #3

180 s



- vypočtete kořeny kvadratické funkce $-2x^2 - 5x = 3$
 - funkci je vhodné si nejprve vhodně upravit

$$2x^2 + 5x + 3 = 0, \Rightarrow a = 2, b = 5, c = 3$$

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-5 \pm \sqrt{25 - 24}}{4}$$

$$x_1 = -1, \quad x_2 = -\frac{3}{2}$$

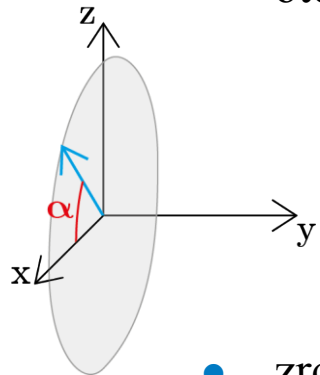
- Matlab obsahuje i funkci, určenou přesně pro tento účel, zkuste ji najít

Cvičení #4

300 s ↑

- uvažujte matice (matice si připravte pro navazující výpočty)

- otočení o úhel α v rovině x - z



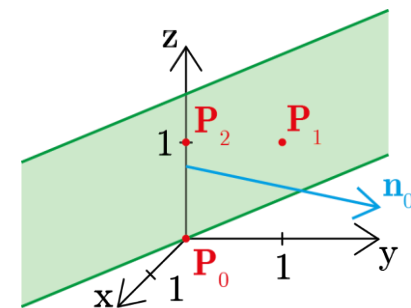
$$\mathbf{R} = \begin{pmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix}$$

- zrcadlení přes rovinu $1x + 2y + 0z = 0$

- použijte Householderovu transformaci $\mathbf{P} = \mathbf{I} - 2\mathbf{n}_0\mathbf{n}_0^T$

$$\mathbf{n}_0 = \frac{\mathbf{v}_1 \times \mathbf{v}_2}{|\mathbf{v}_1 \times \mathbf{v}_2|} \quad \mathbf{P}_1 = [-2; 1; 0]$$
$$\mathbf{P}_2 = [0; 0; 1]$$

$$\mathbf{v}_k = (\mathbf{P}_k - \mathbf{0}) \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{y}_0 \\ \mathbf{z}_0 \end{pmatrix}, \quad k \in \{1, 2\}$$



Cvičení #5

180 s ↑

- užití matici rotace \mathbf{R} pro otočení vektoru $\mathbf{k} = [1; 0; 0]$ o úhel $\alpha = \pi/2$

$$\mathbf{m} = \mathbf{R}\mathbf{k} = (0 \ 0 \ 1)^T$$

- užití matici reflexe \mathbf{P} podél roviny: $1x + 2y + 0z = 0$
 - pro zrcadlení vektorů:

$$\mathbf{u}_1 = \mathbf{n}_0, \quad \mathbf{u}_2 = \left(\frac{5}{2} \ 0 \ 3 \right)^T$$

$$\mathbf{m}_1 = \mathbf{P}\mathbf{u}_1 = -\mathbf{n}_0, \quad \mathbf{m}_2 = \mathbf{P}\mathbf{u}_2 = \left(\frac{3}{2} \ -2 \ 3 \right)^T$$

- pro matice \mathbf{R} a \mathbf{P} vypočítejte determinant
 - dokážete vysvětlit výsledné hodnoty

$$\det \mathbf{R} = 1, \quad \det \mathbf{P} = -1$$

Děkuji!



ver. 3.1 (09/02/2015)

Miloslav Čapek

miloslav.capek@fel.cvut.cz

Jakékoliv úpravy přednášky jsou zakázány.
Využití mimo výuku na ČVUT-FEL není bez souhlasu autorů dovoleno.
Materiál vytvořen v rámci předmětu A0B17MTB.

