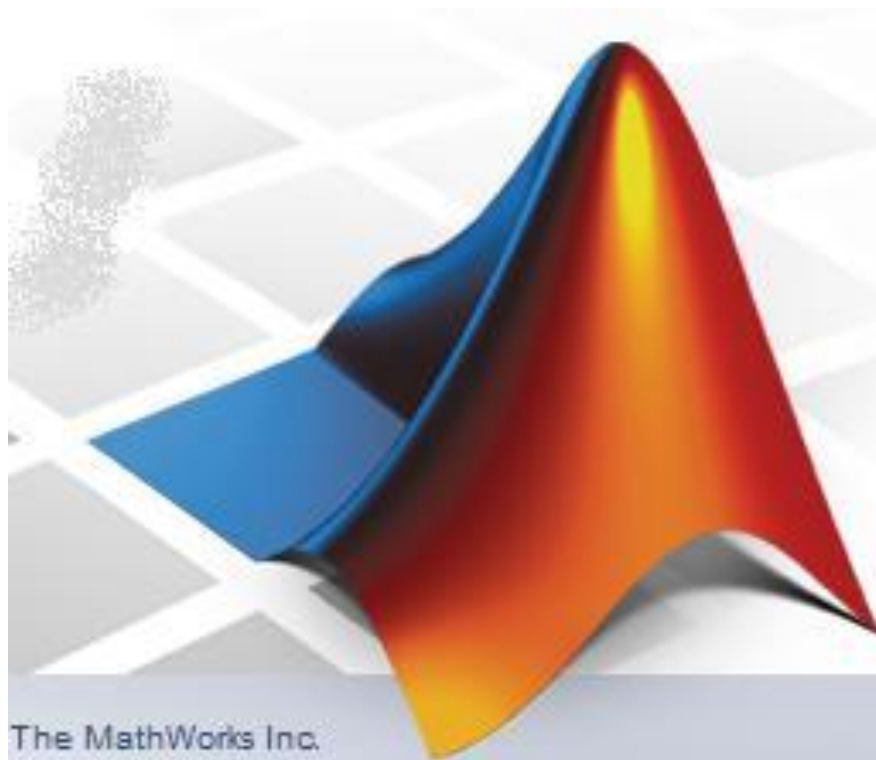


MATLAB (1) - úvod do programovania vedeckých problémov

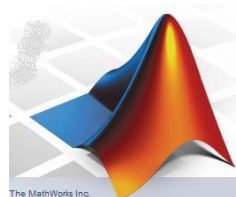


MATLAB (1) - úvod do programovania vedeckých problémov, LS 2020

Program predmetu:

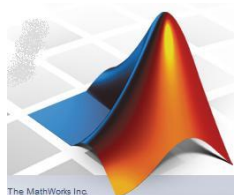
1. týždeň: úvod, základné info o Matlabe, pracovné prostredie Matlabu, interaktívny režim, prvé info o písaní skriptov
2. týždeň: základné operácie s maticami, import a export dát, základné grafické zobrazovanie (grafy a mapy)
3. týždeň: práca s reťazcami, práca so súbormi
4. týždeň: pokročilejšia grafika - popis grafov a máp, 2D grafy
5. týždeň: príkazy, stavba programov, M-súborov
6. týždeň: funkcie – zabudované v Matlabe, tvorba vlastných funkcií
7. týždeň: tvorba vlastných aplikácií, práca s GUI (Graphical User Interface)
8. týždeň: tvorba vlastných aplikácií, nástroj GUIDE

pozn.: zmeny vyhradené



Obsah (6. prednáška):

- pokračovanie v tvorbe M-súborov: funkcie
- tvorba funkcií, vlastností funkcií
- poznámka ku Toolboxom v Matlabe
- zadanie



MATLAB (1) - úvod do programovania vedeckých problémov, LS 2020

Repetitórium z ostatnej prednášky:

M-súbory delíme na:

- **skripty**: neprímajú vstupné a nevracajú výstupné argumenty, pracujú s údajmi premenných vo workspace,
- **funkcie**: prijímajú vstupné údaje a vracajú výstupné údaje, sú väčšinou „volané“ nejakým iným skriptom.

Funkcie v Matlabe – základné informácie:

- realizuje samostatný výpočet alebo vizualizáciu,
- môže mať (ale nemusí) niekoľko vstupných a výstupných parametrov,
- prenos dát sa realizuje cez vstupné a výstupné parametre,
- má svoj pamäťový priestor, ktorý nezdieľa s inými skriptami a ani s hlavným programom, všetky premenné (vrátane vstupných parametrov) definované vo vnútri funkcie sú použiteľné iba vo vnútri funkcie a nie sú v prostredí Workspace – sú to tzv. **lokálne premenné** (príkazy `whos` a `who` však fungujú vo vnútri tela funkcie),
- je uložená v samostatnom M-súbore (môže v ňom byť aj viacej funkcií – o tom neskôr), kde je na začiatku špeciálna hlavička s kľúčovým slovom `function`.

Funkcie v Matlabe:

- **zadefinované Matlabom** (napr. `sin(x)`, `log(x)`, ...)
volanie – buď `meno(parametre)` alebo
`premenna = meno(parametre)`
dôležitý je tu pojem tzv. *handle* funkcie, čo je pomocná premenná (presnejšie údajový typ), ktorá obsahuje informáciu použitú pri tzv. referencovaní funkcie a táto môže byť neskôr využitá (podrobnejšie niekedy s Davidom ...),
- **definované užívateľom**
musia byť samostatných M-súboroch alebo vhodne zakomponované v skripte (samotný celý skript potom musí byť funkciou a v rámci nej môžu byť definované ďalšie pod-funkcie)

zadefinovanie:

```
function [fout] = meno(vstupne parametre)
%jadro funkcie;
%dalsie prikazy ...
fout = vysledny vztah;
end; %nie je povinne pri jednej funkcii v jednom M-subore
alebo prvý riadok všeobecnejšie:
function {[out1,out2,...]} = meno_funkcie1(inp1,inp2,...)
```

Funkcie v Matlabe:

- definované užívateľom

prvý riadok všeobecnejšie:

```
function {[out1,out2,...]} meno_funkcie1(inp1,inp2,...)
```

Príklad (rozdelí jednu matice na dve):

```
function [A1,A2] = rozdel(A,N)
%funkcia rozdeli riadkovu maticu na dve,
%vracia dve matice, ktore vzniknu
%rozdelenim matice A na pozicii N
    A1=A(1:N);
    A2=A(N+1:length(A))
end
```

Pozn.: Výstupné argumenty sa zadávajú v hranatých zátvorkách. Pri jednom výstupnom argumente ich však netreba písať. Ak by funkcia nevracala žiadne hodnoty, stačí túto časť vynechať alebo napísať prázdne hranaté zátvorky. Meno funkcie sa musí začínať písmenom, nesmie obsahovať nepovolené znaky, prekročiť povolenú dĺžku a nesmie sa zhodovať s nejakým kľúčovým slovom Matlabu.

Prvý riadok nemusí byť riadok s kľúčovým slovom `function`, ale môže to byť komentárový riadok označený s `%`.

Prvý komentárový riadok za riadkom s kľúčovým slovom sa nazýva H1 riadok a vypíše sa pri aplikovaní príkazu `help meno_funkcie`

„zabudované“ funkcie v Matlabe:

Definované funkcie v Matlabe:

- matematické funkcie (napr.: `sin()`, `error()`, `quadl()`),
- grafické (napr.: `plot()`, `image()`),
- funkcie pre prácu s reťazcami (napr.: `strtok()`, `str2int()`,)
- funkcie pre prácu so súbormi (napr.: `dlmread()`, `importdata()`,)
- systémové (napr.: `computer`, `system()`, ...),
- ďalšie...

V podstate naprostá väčšina „príkazov“ v Matlabe sú svojim spôsobom funkcie... (okrem tzv. kľúčových slov).

tvorba vlastných funkcií v Matlabe – jednoduchý príklad:

Vytvorili sme funkciu s menom `funkcia1` (presne tak sa musí volať aj samotný m-skript, v ktorom sa nachádza), jej úlohou je umocniť vstupnú hodnotu `vstpar` na druhú a vypísať jej vnútorné premenné:

```
function funkc1a1(vstpar)
% funkc1a na testovanie - s jedným vstupným parametrom
all=vstpar %fungovalo by to aj bez tejto pomocne premennej all
b22=all^2
whos
```

Zavoláme ju napr. z Command Window (alebo z iného skriptu) – zadaním:

`funkc1a1(100)`

Výstup je:

```
>> funkc1a1(100)
```

```
all =
```

```
    100
```

```
b22 =
```

```
   10000
```

Name	Size	Bytes	Class	Attributes
all	1x1	8	double	
b22	1x1	8	double	
vstpar	1x1	8	double	

tvorba vlastných funkcií v Matlabe – zložitejší príklad:

Príklad vytvorenia vlastnej funkcie (s trošku zložitejším vzorcom).

Pomocná funkcia je v M-súbore *samplfsqrt.m*:

```
function [fout] = samplfsqr(x)
% square of sampling function: sin(x)/x;
fout = (sin(x)/x)^2;
```

Ide o tzv. sampling
function (a jej mocninu):

$$\left[\frac{\sin(x)}{x} \right]^2$$

hlavný program (prekopírujte si ho do Command Window):

```
clear all; close all; clc;
x=-10; i=1;
while x<=10
    y(i) = samplfsqr(x);
    x=x+0.1; i=i+1;
end; % koniec cyklu while()
plot(y)
```

tvorba vlastných funkcií v Matlabe – zložitejší príklad A:

Príklad využitia volania pomocnej funkcie pri zložitejšom výpočte.

Výpočet špeciálnej funkcie V_{xz} ako funkcie $f(x)$ podľa nasledujúceho vzorca (s využitím pomocnej funkcie na výpočet čiastkovej hodnoty funkcie).

Pomocná funkcia je v M-súbore *Vxz_pomoc.m*:

```
function [fout] = Vxz_pomoc(X,Y,Z)
% pomocna funkcia sqrt(X*X + Y*Y + Z*Z)
R = sqrt(X*X + Y*Y + Z*Z);
fout = log(Y + R);
% -----
```

Vzorec na výpočet:

$$V_{xz} = G\sigma \left| \ln(\eta + R) \right| \left| \frac{\xi_2}{\xi_1} \right| \left| \frac{\eta_2}{\eta_1} \right| \left| \frac{\zeta_2}{\zeta_1} \right|,$$

$$R = \sqrt{(\xi - x)^2 + \eta^2 + \zeta^2}.$$

hlavný program (viac priamo v skripte *prism_3Dgrav_Vxz.m*):

```
G = 6.67E-11; sigma = 1000;
xsi1 = -500; xsi2 = 500; eta1 = -500; eta2 = 500; dzeta1 = 0.1; dzeta2 = 4000;
zc = 0; yc = 0; xc = -3000; i = 1;
while xc <= 3000
    X1=xsi1-xc; X2=xsi2-xc; Y1=eta1-yc; Y2=eta2-yc; Z1=dzeta1-zc; Z2=dzeta2-zc;
    pomoc = Vxz_pomoc(X2,Y2,Z2)-Vxz_pomoc(X1,Y2,Z2)-Vxz_pomoc(X2,Y1,Z2) ...
        +Vxz_pomoc(X1,Y1,Z2)-Vxz_pomoc(X2,Y2,Z1)+Vxz_pomoc(X1,Y2,Z1) ...
        +Vxz_pomoc(X2,Y1,Z1)-Vxz_pomoc(X1,Y1,Z1);
    Vxz(i) = -1E5*G*sigma*pomoc; xcor(i) = xc;
    xc = xc + 2; i = i + 1;
end;
```

Pomocná funkcia je *SL_pomoc.m*, hlavný program v skripte *spherical_layer_Vz.m*).

Funkcie v Matlabe – ďalšie vlastnosti:

V Matlabe funguje aj **rekurzia** – práca s funkciami volajúcimi samu seba.
Príklad (výpočet faktoriálu – skript *fakt.m*):

```
function y = fakt(N)
%vypocet faktorialu
if N == 0
    y = 1;
else
    y = N * fakt(N-1);
end;
```

Volanie:

```
fakt(5)
```

alebo

```
fakt(11)
```

Pozn.: V Matlabe neexistuje nekonečná rekurzia – došlo by ku preplneniu zásobníka a ku chybovému hláseniu.

Funkcie v Matlabe – ďalšie vlastnosti:

Sú situácie, kedy nevieme dopredu presne koľko argumentov budeme mať.

Funkcie s **premenným počtom argumentov** - práca s pomocnými Matlab.

funkciami `varargin(:)` a `nargin`

(`varargin` = „values of arguments in“, `nargin` = „number of arguments in“)

Príklad nájdete ako m-súbor *funkcia3.m*:

```
function funkcia3(varargin)
disp('Pocet argumentov je: '); nargin
S = varargin(:)
```

Príklady volania:

```
funkcia3(1,10)
```

alebo

```
funkcia3(1,10,2+3i)
```

Pozn.: Pomocná funkcia `nargout` dáva info o počte výstupných argumentov funkcie (`nargout` = „number of arguments out“).
Podobne `varargout()` dáva samotné hodnoty výstupných argumentov.

Funkcie v Matlabe – ďalšie vlastnosti:

Viacej funkcií v jednom m-súbore:

Je vhodné použiť v prípadoch, kedy chceme zvyšovať rýchlosť programov. Podľa prvej funkcie sa nazýva daný M-súbor, a v ňom sa odkazuje na druhú ev. ďalšiu funkciu, napísanú v rovnakom M-súbore (nesmie existovať iná referencia na tieto ďalšie funkcie okrem tohto M-súboru).

Príklad: (skript: *Pok12.m*) (nakreslí zadaný počet grafov $\sin()$ do jedného obrázku)

```
function Pok12(vstpar)
Pocetgrafu=vstpar;  t=0:6*pi/100:6*pi;
Pok123(Pocetgrafu,t)
```

```
function Pok123(vstpar1,vstpar2)
Pocgraf=vstpar1;  t=vstpar2; u=sin(t);
for k=1:Pocgraf
    subplot(1,Pocgraf,k);
    plot(t,u);
end;
```

Príklad volania:

```
Pok12(3)
```

Funkcie v Matlabe – ďalšie informácie:

Delenie funkcií podľa ich úlohy:

- anonymné funkcie – funkcia v príkazovom riadku
- primárna funkcia – prvá funkcia v M-súbore
- subfunkcie – funkcie za primárnou funkciou
- vnorené funkcie – funkcie v ďalšej funkcii
- súkromné funkcie – funkcie pracovného adresára
- preťažené funkcie – funkcie s rovnakým menom a rôznymi argumentmi

Anonymné funkcie

Anonymná funkcia je zjednodušená forma funkcie v Matlabe, ktorá nepotrebuje M-súbor. Pozostáva z jedného výrazu v Matlabe a rôzneho počtu vstupných alebo výstupných argumentov. Môže byť zadefinovaná v príkazovom riadku, ďalšej funkcii alebo skripte.

Syntax anonymnej funkcie je nasledujúci:

```
f = @ (argumenty) výraz
```

Samozrejme môžeme použiť niekoľko vstupných argumentov alebo žiadne.

Využitie anonymnej funkcie si môžeme ukázať na nasledujúcom príklade (2. mocnina):

```
sqr = @(x) x.^2;  
a = sqr(5)
```

Funkcie v Matlabe – ďalšie informácie:

Delenie funkcií podľa ich úlohy:

- anonymné funkcie – funkcia v príkazovom riadku
- primárna funkcia – prvá funkcia v M-súbore
- subfunkcie – funkcie za primárnou funkciou

Primárna funkcie

Každá iná funkcia ako anonymná musí byť definovaná v M-súbore. **Prvá funkcia v M-súbore sa nazýva primárna.** Za primárnou funkciou môže nasledovať niekoľko subfunkcií, ktoré môže primárna funkcia volať. Vo väčšine prípadov bude však M-súbor obsahovať iba primárnu funkciu.

Subfunkcie (podfunkcie)

M-súbor môže obsahovať viac ako jednu funkciu. Ďalšie funkcie okrem primárnej sa potom nazývajú subfunkcie (prípadne podfunkcie). Subfunkcie sú viditeľné iba vrámci M-súboru, teda ich môže volať iba primárna funkcia alebo iné subfunkcie v M-súbore. Ak by sme chceli zavolať popis funkcie (help) mysubfun v súbore funkcie myfun, museli by sme použiť nasledujúcu konštrukciu:

```
help myfun>mysubfun
```


Funkcie v Matlabe – ďalšie informácie:

Delenie funkcií podľa ich úlohy:

- vnorené funkcie – funkcie v ďalšej funkcii

Vnorené funkcie

Funkcie sa môžu definovať aj v rámci tela inej funkcie. Vnorená funkcia obsahuje všetky komponenty ako obyčajná funkcia a nachádza sa v tele inej funkcie.

Vnorené funkcie sa definujú nasledujúcim spôsobom:

```
function x = A(p1, p2)
    ...
    function y = B(p3)
        ...
    end
    ...
end
```

Samozrejme vnorené funkcie môžu byť aj v rámci inej vnorenej funkcie. Volanie vnorených funkcií má niekoľko pravidiel. Vnorenú funkciu môžeme zavolať z vyššej úrovne ako je vnorená funkcia, z inej vnorenej funkcie na rovnakej úrovni alebo z vnorenej funkcie na nižšej úrovni.

Funkcie v Matlabe – ďalšie informácie:

Delenie funkcií podľa ich úlohy:

- súkromné funkcie – funkcie pracovného adresára
- preťažené funkcie – funkcie s rovnakým menom a rôznymi argumentmi

Súkromné funkcie

Súkromné funkcie sú špeciálnym typom funkcií. Sú uložené v adresároch so špeciálnym názvom `private`. Sú viditeľné iba pre funkcie v rodičovskom adresári. Pretože sú schované môžu sa nazývať ako funkcie v iných adresároch. Matlab najskôr hľadá súkromné funkcie pred štandardnými funkciami. Pomocníka ku súkromnej ziskame nasledovným príkazom:

```
help private/myprivfun
```

Pret'ažené funkcie

Pret'ažovanie funkcií funguje ako v iných programovacích jazykoch. Pret'ažované funkcie slúžia na volanie rovnakých funkcií s rôznym počtom alebo typom argumentov. Matlab potom dokáže rozlíšiť, ktorú funkciu má použiť. Dajú sa vytvoriť v jednom M-súbore podobne ako subfunkcie funkcie s rovnakým menom, ale inými argumentami.

Funkcie v Matlabe – ďalšie informácie:

Globálne premenné:

Niekedy chceme používať niektoré premenné „všade“ – aj v hlavnom skripte, aj vo funkciách. Na to môžeme použiť globálne premenné.

Globálne premenné sa využívajú keď chceme, aby viaceré funkcie pristupovali k jednej premennej, prípadne aby mali prístup k premennej vo Workspace príkazového riadka.

Premenná sa musí deklarovat' kľúčovým slovom `global` v rámci každej funkcie (aj v hlavnom skripte), kde sa použije. Globálne premenné sa často píšú veľkými písmenami na lepšie rozoznanie.

Pozor! Niekedy môže dôjsť k prepísaniu ich hodnôt a chyby spôsobené zlým použitím globálnych premenných sa hľadajú ťažšie. Alternatíva ku globálnej premennej je predať hodnotu ako argument funkcie a jej novú hodnotu vrátiť ako návratovú hodnotu funkcie.

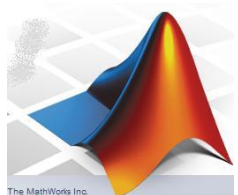
```
function h = falling(t)
    global GRAVITY
    h = 1/2*GRAVITY*t.^2
end
```

Volanie:

```
global GRAVITY
GRAVITY = 10;
y = falling(10);
```

Obsah (6. prednáška):

- pokračovanie v tvorbe M-súborov: funkcie
- tvorba funkcií, vlastností funkcií
- poznámka ku Toolboxom v Matlabe
- zadanie



MATLAB - toolboxes

- MATLAB poskytuje množstvo „ucelených balíkov funkcií“, zameraných na určitú tému – nazývajú sa **toolboxes**,
- často neboli vyvinuté priamo Mathworks, ale treťou stranou, neskôr však boli odkúpené a stali sa súčasťou ich ponuky,
- niektoré funkcie boli časom prebrané z toolboxov do základnej výbavy MATLAB (napr. rýchla Four. transformácia - FFT),
- väčšinou si ich treba dokúpiť ku základnej výbave MATLAB,
- prehľad existujúcich toolboxes – napr.:
<http://www.mathworks.com/help/index.html>
- **získanie informácie o dostupných toolboxes na vašom počítači** – najjednoduchšia cesta je zadanie príkazu `ver` v Command window,
- iná cesta je cez *Preferences* v základnom Matlab menu (tento zoznam však býva niekedy neúplný...),

MATLAB - toolboxes (inštalácia na CVTI) 1/2

MATLAB R2015a - academic use

HOME PLOTS APPS

Find Files New Script New Open Compare Import Data Save Workspace Open Variable Open Variable Clear Workspace Analyze Code Run and Time Clear Commands Simulink Library Layout Set Path Parallel Preferences Help Request Support Community Add-Ons

FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES

C:\Users\Roman.Pasteka\Documents\MATLAB

Current Folder

Command Window

```
Academic License

>> ver

-----
MATLAB Version: 8.5.0.197613 (R2015a)
MATLAB License Number: 884063
Operating System: Microsoft Windows Server 2008 R2 Enterprise Version 6.1 (Build 7601: Service Pack 1)
Java Version: Java 1.7.0_60-b19 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
-----

MATLAB Version 8.5 (R2015a)
Simulink Version 8.5 (R2015a)
Aerospace Blockset Version 3.15 (R2015a)
Aerospace Toolbox Version 2.15 (R2015a)
Bioinformatics Toolbox Version 4.5.1 (R2015a)
Communications System Toolbox Version 6.0 (R2015a)
Computer Vision System Toolbox Version 6.2 (R2015a)
Control System Toolbox Version 9.9 (R2015a)
Curve Fitting Toolbox Version 3.5.1 (R2015a)
DSP System Toolbox Version 9.0 (R2015a)
Database Toolbox Version 5.2.1 (R2015a)
Datafeed Toolbox Version 5.1 (R2015a)
Econometrics Toolbox Version 3.2 (R2015a)
Embedded Coder Version 6.8 (R2015a)
Filter Design HDL Coder Version 2.9.7 (R2015a)
Financial Instruments Toolbox Version 2.1 (R2015a)
Financial Toolbox Version 5.5 (R2015a)
Fixed-Point Designer Version 5.0 (R2015a)
Fuzzy Logic Toolbox Version 2.2.21 (R2015a)
Global Optimization Toolbox Version 3.3.1 (R2015a)
HDL Coder Version 3.6 (R2015a)
HDL Verifier Version 4.6 (R2015a)
Image Acquisition Toolbox Version 4.9 (R2015a)
Image Processing Toolbox Version 9.2 (R2015a)
Instrument Control Toolbox Version 3.7 (R2015a)
MATLAB Coder Version 2.8 (R2015a)
MATLAB Compiler Version 6.0 (R2015a)
MATLAB Compiler SDK Version 6.0 (R2015a)
MATLAB Report Generator Version 4.1 (R2015a)
Mapping Toolbox Version 4.1 (R2015a)
Model Predictive Control Toolbox Version 5.0.1 (R2015a)
Model-Based Calibration Toolbox Version 4.8.1 (R2015a)
Neural Network Toolbox Version 8.3 (R2015a)
OPC Toolbox Version 3.3.3 (R2015a)
```

Workspace

Name	Value
------	-------

Details

MATLAB - toolboxes (inštalácia na CVTI) 2/2

MATLAB R2015a - academic use

HOME PLOTS APPS

New Script New Open Find Files Import Data Save Workspace New Variable Open Variable Analyze Code Run and Time Simulink Library Layout Set Path Help Request Support Add-Ons

FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES

C:\Users\Roman.Pasteka\Documents\MATLAB

Current Folder

Name

Details

Command Window

Instrument Control Toolbox	Version 3.7	(R2015a)
MATLAB Coder	Version 2.8	(R2015a)
MATLAB Compiler	Version 6.0	(R2015a)
MATLAB Compiler SDK	Version 6.0	(R2015a)
MATLAB Report Generator	Version 4.1	(R2015a)
Mapping Toolbox	Version 4.1	(R2015a)
Model Predictive Control Toolbox	Version 5.0.1	(R2015a)
Model-Based Calibration Toolbox	Version 4.8.1	(R2015a)
Neural Network Toolbox	Version 8.3	(R2015a)
OPC Toolbox	Version 3.3.3	(R2015a)
Optimization Toolbox	Version 7.2	(R2015a)
Parallel Computing Toolbox	Version 6.6	(R2015a)
Partial Differential Equation Toolbox	Version 2.0	(R2015a)
Phased Array System Toolbox	Version 3.0	(R2015a)
RF Toolbox	Version 2.16	(R2015a)
Robust Control Toolbox	Version 5.3	(R2015a)
Signal Processing Toolbox	Version 7.0	(R2015a)
SimBiology	Version 5.2	(R2015a)
SimDriveline	Version 2.8	(R2015a)
SimElectronics	Version 2.7	(R2015a)
SimEvents	Version 4.4	(R2015a)
SimHydraulics	Version 1.16	(R2015a)
SimMechanics	Version 4.6	(R2015a)
SimPowerSystems	Version 6.3	(R2015a)
SimRF	Version 4.4	(R2015a)
Simscape	Version 3.13	(R2015a)
Simulink 3D Animation	Version 7.3	(R2015a)
Simulink Code Inspector	Version 2.3	(R2015a)
Simulink Coder	Version 8.8	(R2015a)
Simulink Control Design	Version 4.2	(R2015a)
Simulink Design Optimization	Version 2.7	(R2015a)
Simulink Design Verifier	Version 2.8	(R2015a)
Simulink Desktop Real-Time	Version 5.0	(R2015a)
Simulink Report Generator	Version 4.1	(R2015a)
Simulink Verification and Validation	Version 3.9	(R2015a)
Spreadsheet Link EX	Version 3.2.3	(R2015a)
Stateflow	Version 8.5	(R2015a)
Statistics and Machine Learning Toolbox	Version 10.0	(R2015a)
Symbolic Math Toolbox	Version 6.2	(R2015a)
System Identification Toolbox	Version 9.2	(R2015a)
SystemTest	Version 2.6.9	(R2015a)
Trading Toolbox	Version 2.2	(R2015a)
Wavelet Toolbox	Version 4.14.1	(R2015a)

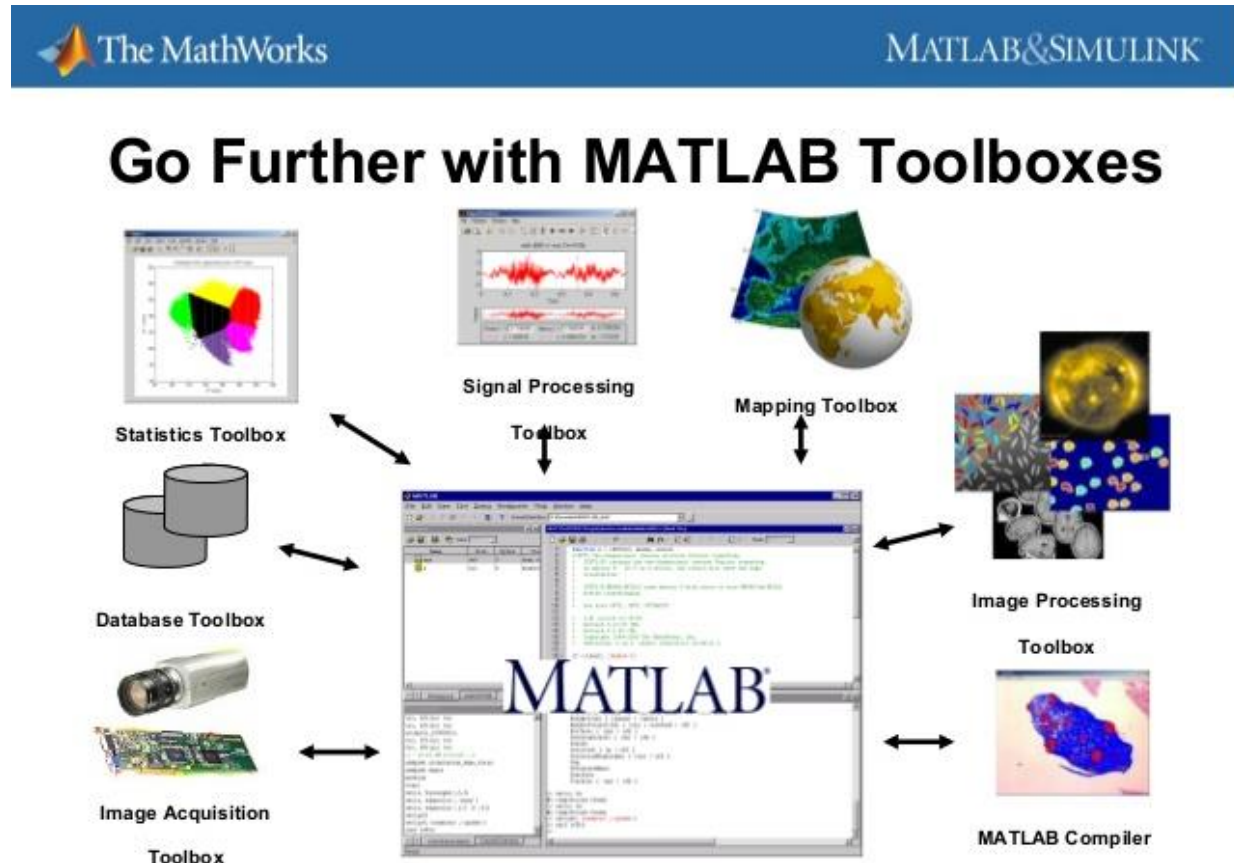
>>

Workspace

Name	Value
------	-------

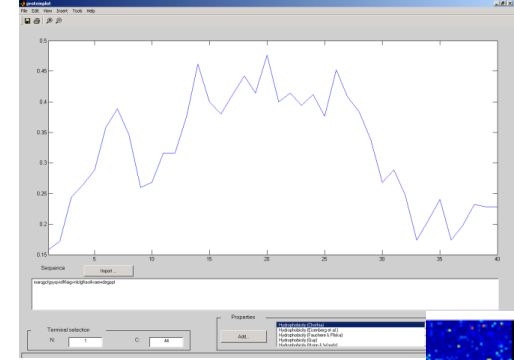
MATLAB - toolboxes – asi najdôležitejšie z nich:

- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing
- Spline
- Statistics
- System Identification
- Wavelet

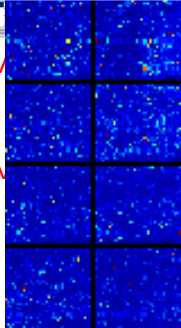


MATLAB - toolboxes – niekoľko príkladov:

- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing
- Spline
- Statistics
- System Identification
- Wavelet



```
212 PYESFTFPELMRKGSYNPVTHIYTAQDVKEV
    ||| :||| :||| :||| :|||
: |::
321 PYISRYPELAVHGAYSE-SETYSEQDVREV
```



File I/O

- FASTA, PDB, SCF, GPR, GAL

Web Connectivity

- GenBank, EMBL, PIR, PDB

Sequence Analysis & Alignment

- Needleman-Wunsch, Smith-Waterman
- DNA/RNA/AA conversions, pattern searching

Microarray Normalization & Visualization

- Lowess, global mean, MAD (median absolute deviation)

Protein Visualization

- Atomic composition, molecular weight, hydrophobicity profile

MATLAB - toolboxes – niekoľko príkladov:

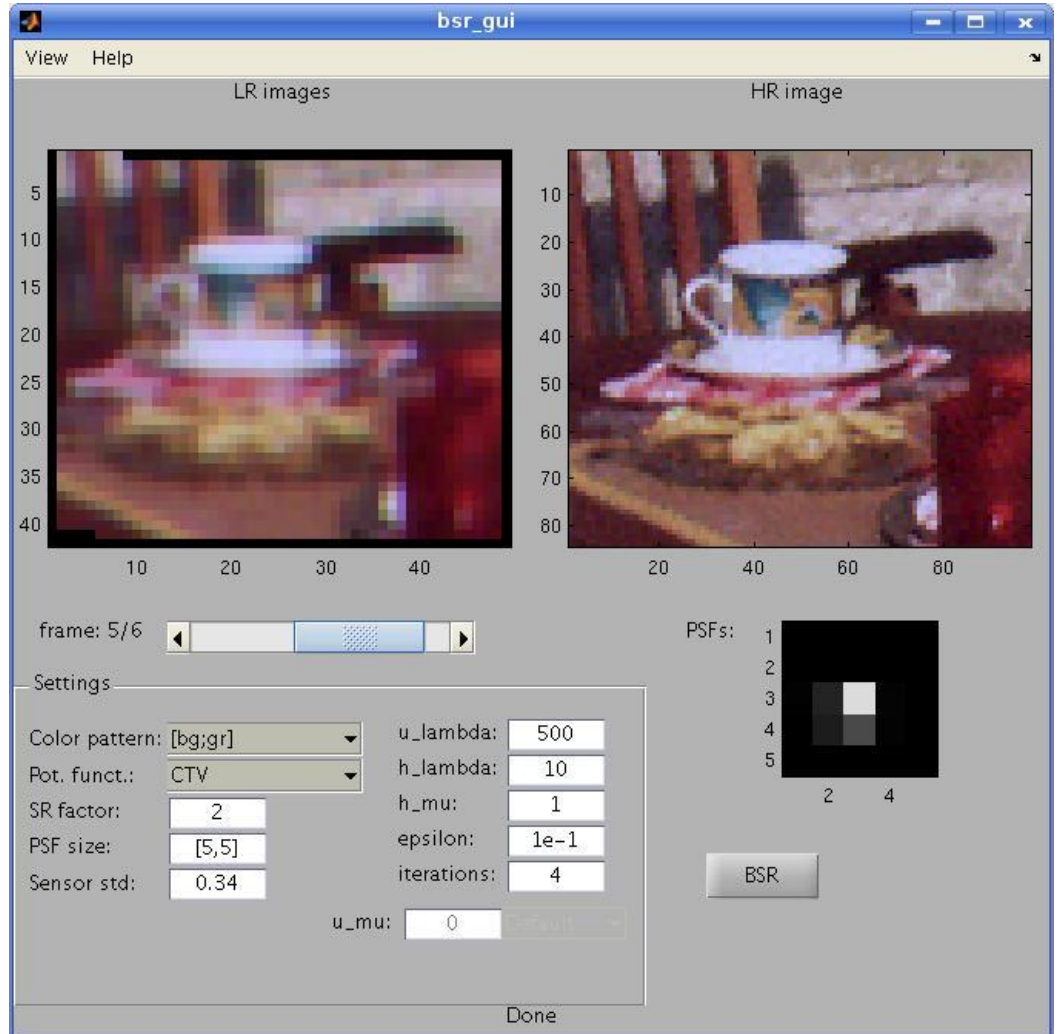
- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing
- Spline
- Statistics
- System Identification
- Wavelet

- špeciálne postupy na riešenie optimalizačných problémov (motivované procesmi v biologickej evolúcii)

MATLAB - toolboxes – niekoľko príkladov:

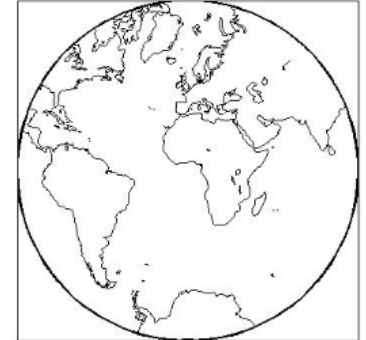
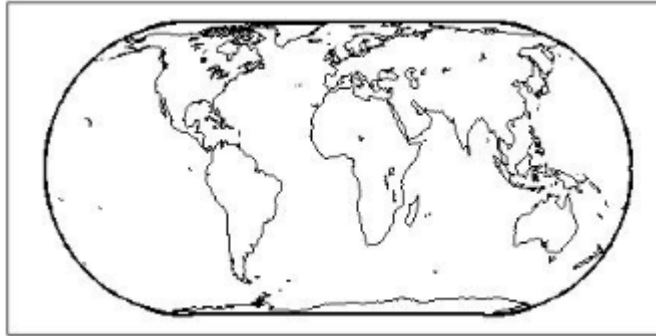
- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing
- Spline
- Statistics
- System Identification
- Wavelet

- spracovanie obrazu, filtrácia, ...

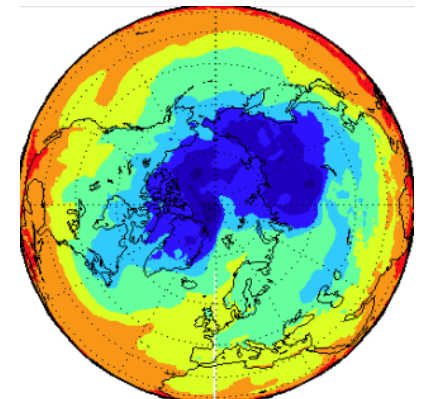
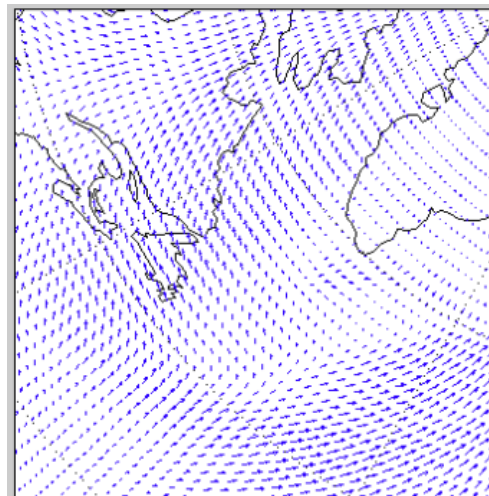


MATLAB - toolboxes – niekoľko príkladov:

- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing
- Spline
- Statistics
- System Identification
- Wavelet



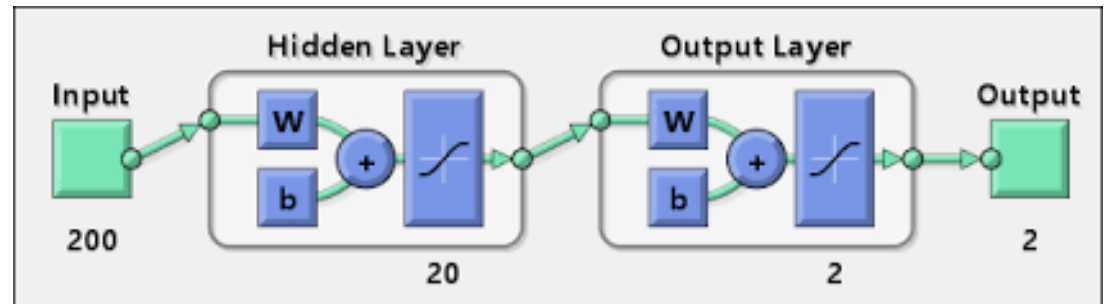
- práca s rastrami a aj vektorovými údajmi (napr. celosvetové gridy),
- tvorba máp, projekcie, súradnicové systémy,
- analýza dát,



MATLAB - toolboxes – niekoľko príkladov:

- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network**
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing
- Spline
- Statistics
- System Identification
- Wavelet

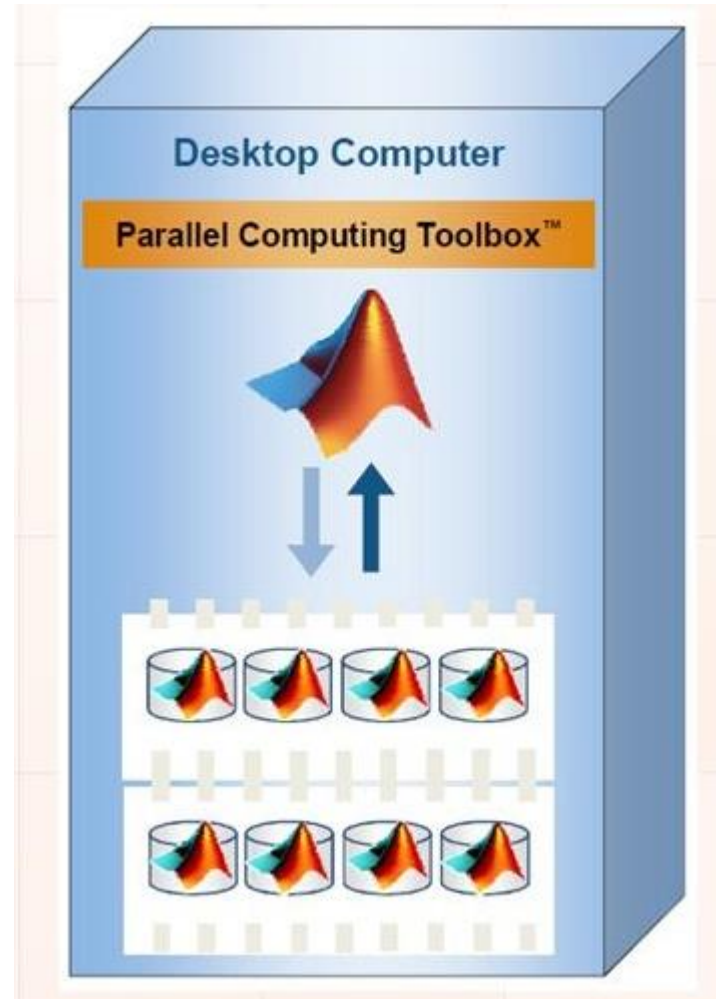
- špeciálne postupy na riešenie optimalizačných problémov (motivované popisom procesov v mozgu)



MATLAB - toolboxes – niekoľko príkladov:

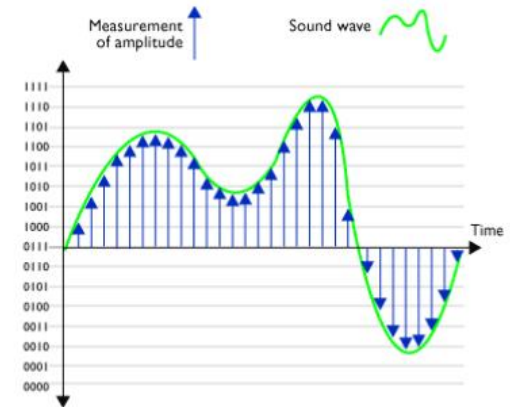
- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing
- Spline
- Statistics
- System Identification
- Wavelet

- dôležitý pre urýchlenie výpočtov
(paralelizácia výpočtu)

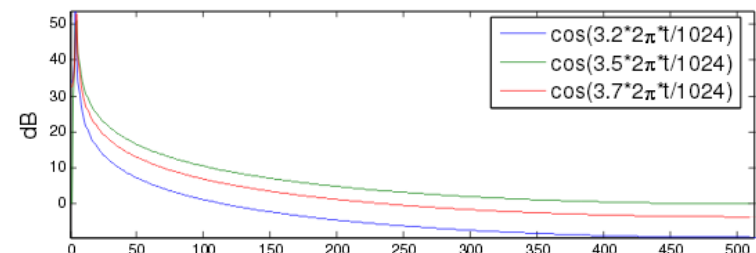
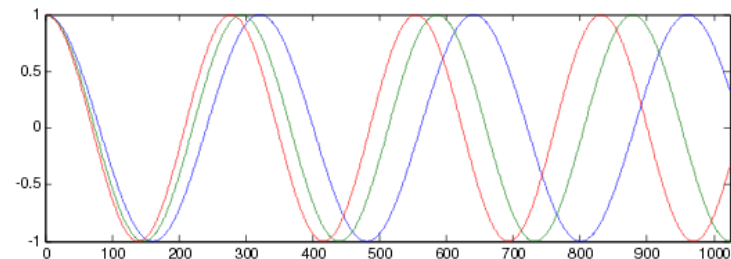


MATLAB - toolboxes – niekoľko príkladov:

- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing**
- Spline
- Statistics
- System Identification
- Wavelet



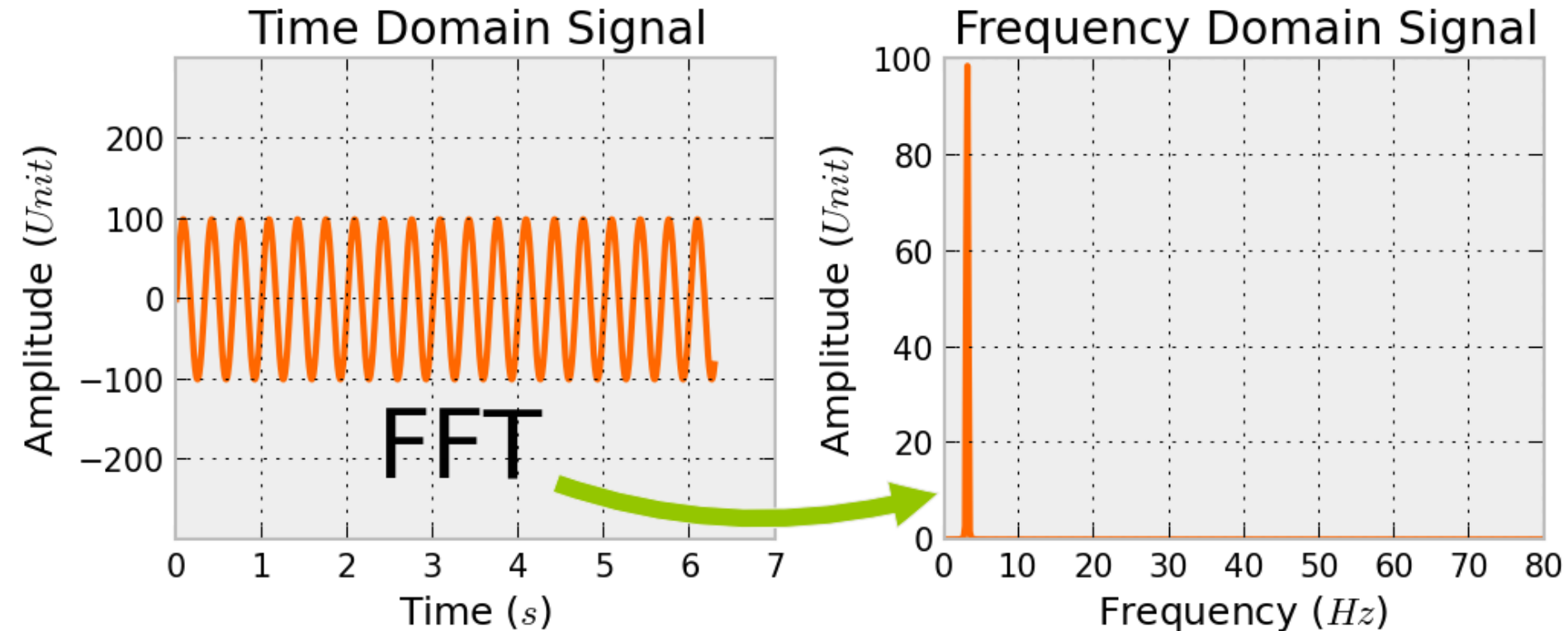
- práca s diskrétnymi údajmi,
- filtrácia, konvolúcia
- spektrálna analýza



MATLAB (1) - rýchla Fourierova transformácia (FFT):

Pre získanie základných výpočtov Fourierovho spektra nie je potrebné použiť Signal processing toolbox, niektoré základné funkcie sa nachádzajú v rámci základnej inštalácie Matlab:

FFT (Fast Fourier Transform) – funkcia: $z = \text{fft}(y)$
(pozor, výsledok je komplexná funkcia!)



MATLAB - toolboxes – niekoľko príkladov:

- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing
- Spline
- Statistics
- System Identification
- Wavelet

Statistics and Machine Learning Toolbox

Descriptive Statistics and Visualization

Data import and export, descriptive statistics, visualization

Probability Distributions

Data frequency models, random sample generation, parameter estimation

Hypothesis Tests

t-test, F-test, chi-square goodness-of-fit test, and more

Cluster Analysis

Unsupervised learning techniques to find natural groupings and patterns in data

ANOVA

Analysis of variance and covariance, multivariate ANOVA, repeated measures ANOVA

Regression

Linear, generalized linear, nonlinear, and nonparametric techniques for supervised learning

Classification

Supervised learning algorithms for binary and multiclass problems

Dimensionality Reduction

PCA, factor analysis, nonnegative matrix factorization, sequential feature selection, and more

Industrial Statistics

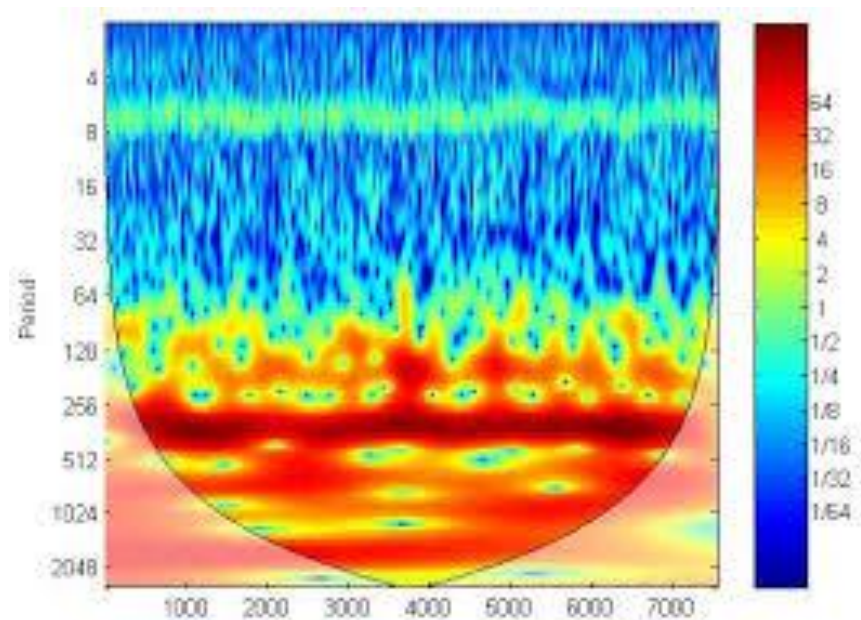
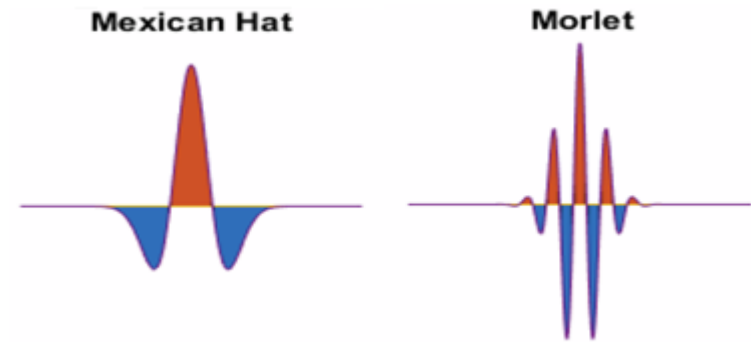
Design of experiments (DOE); survival and reliability analysis; statistical process control

Speed Up Statistical Computations

Parallel or distributed computation of statistical functions

MATLAB - toolboxes – niekoľko príkladov:

- Bioinformatics
- Communications
- Control System
- Curve Fitting
- Filter Design
- Fixed-Point
- Fuzzy Logic
- Genetic Algorithm
- Image Acquisition
- Image Processing
- Mapping
- Model Predictive Control
- Neural Network
- Optimization
- Parallel Computing
- Partial Differential Equation
- Signal Processing
- Spline
- Statistics
- System Identification
- Wavelet

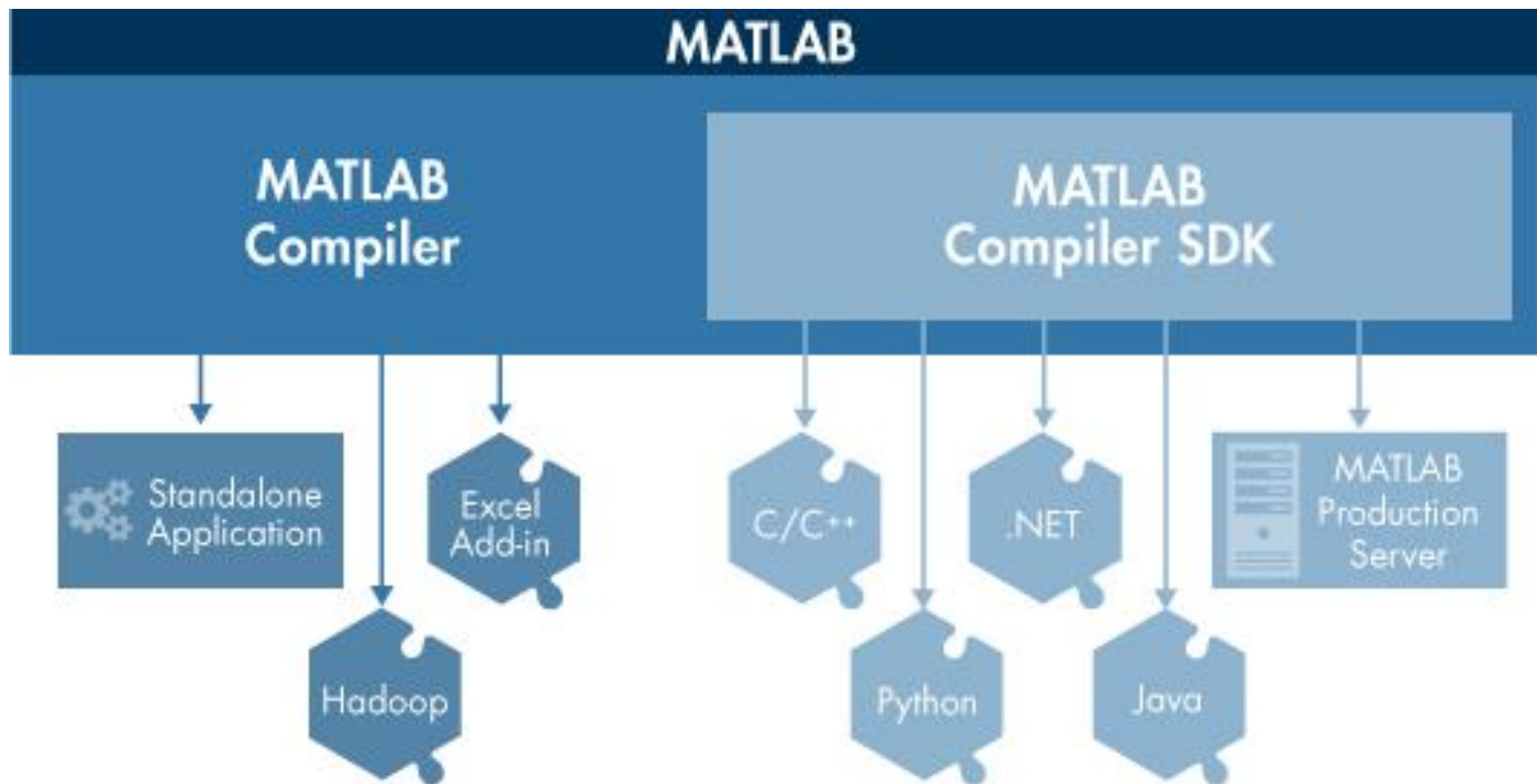


- aproximácie pomocou systémov špeciálnych funkcií

MATLAB	Parallel Computing Toolbox
Simulink	Partial Differential Equation Toolbox
Aerospace Blockset	Phased Array System Toolbox
Aerospace Toolbox	Polyspace Bug Finder
Antenna Toolbox	Polyspace Code Prover
Audio System Toolbox	Polyspace Products for Ada
Bioinformatics Toolbox	RF Toolbox
Communications System Toolbox	Robotics System Toolbox
Computer Vision System Toolbox	Robust Control Toolbox
Control System Toolbox	Signal Processing Toolbox
Curve Fitting Toolbox	SimBiology
Data Acquisition Toolbox	SimEvents
Database Toolbox	SimRF
Datafeed Toolbox	Simscape
DO Qualification Kit (for DO-178)	Simscape Driveline
DSP System Toolbox	Simscape Electronics
Econometrics Toolbox	Simscape Fluids
Embedded Coder	Simscape Multibody
Filter Design HDL Coder	Simscape Multibody Link
Financial Instruments Toolbox	Simscape Power Systems
Financial Toolbox	Simulink 3D Animation
Fixed-Point Designer	Simulink Code Inspector
Fuzzy Logic Toolbox	Simulink Coder
Global Optimization Toolbox	Simulink Control Design
HDL Coder	Simulink Design Optimization
HDL Verifier	Simulink Design Verifier
IEC Certification Kit (for ISO 26262 and IEC 61508)	Simulink Desktop Real-Time
Image Acquisition Toolbox	Simulink PLC Coder
Image Processing Toolbox	Simulink Real-Time
Instrument Control Toolbox	Simulink Report Generator
LTE System Toolbox	Simulink Test
Mapping Toolbox	Simulink Verification and Validation
MATLAB Coder	Spreadsheet Link
MATLAB Compiler	Stateflow

Dôležitá poznámka – Matlab compiler

- transformuje m-skripty buď na kód iných programovacích jazykov (C/C++, Python, Java) alebo vytvorí samostatné aplikácie (exe-verzie).



Zadanie č.7:

Skúste naprogramovať vlastné riešenie (pomocou vlastných funkcií) pre lineárnu regresiu (podľa zadaných vzorcov) a skontrolovať jej výsledok pomocou Matlabovskej funkcie `polyfit()`.

K dispozícii máte súbor *G344.dat*, v ktorom sa nachádzajú výsledky merania teploty a vlhkosti z miestnosti G-344 (sprievodné údaje ku meraniu prirodz. úhrnnej rádioaktivity) – 1.stĺpec: teplota [°C], 2. stĺpec: vlhkosť [%] (sú to vybrané stĺpce zo súboru z 2.prednášky).

Načítajte tieto dáta do matic x a y , a pre výpočet koeficientov a a b aproximujúcej priamky $y_{\text{aprox}} = ax + b$ použite nasledujúce známe vzťahy:

$$a = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - \left(\sum x_i \right)^2} \quad b = \frac{\sum y_i - a \sum x_i}{n}$$

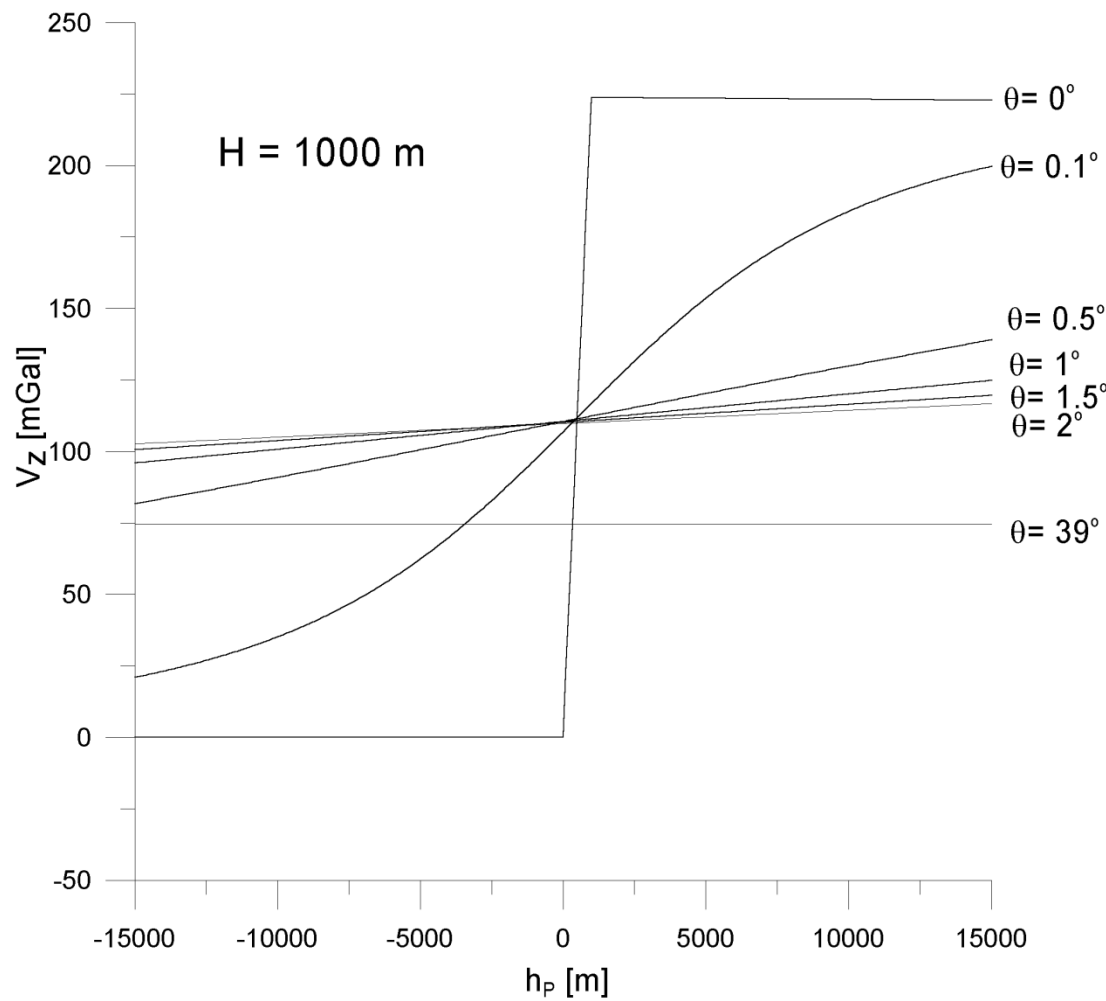
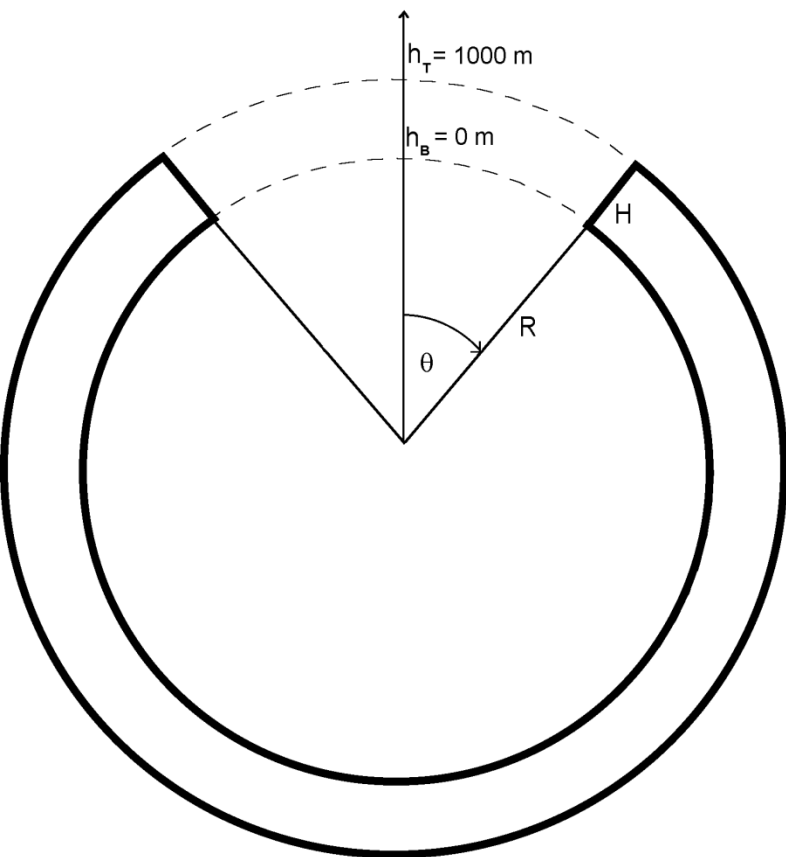
kde n je počet údajov v súbore.

Pozn.: Pri výpočtoch môžete efektívne využiť matlabovskú funkciu `sum()`.

Výsledky môžete samozrejme vhodne vizualizovať.

**pomocné obrázky ku skriptu
pre Vz účinok orezanej sférickej vrstvy**

orezaná sférická vrstva



Interval h_P od -15 km do +15 km

orezaná sférická vrstva – väčší interval vertikálnej súradnice

