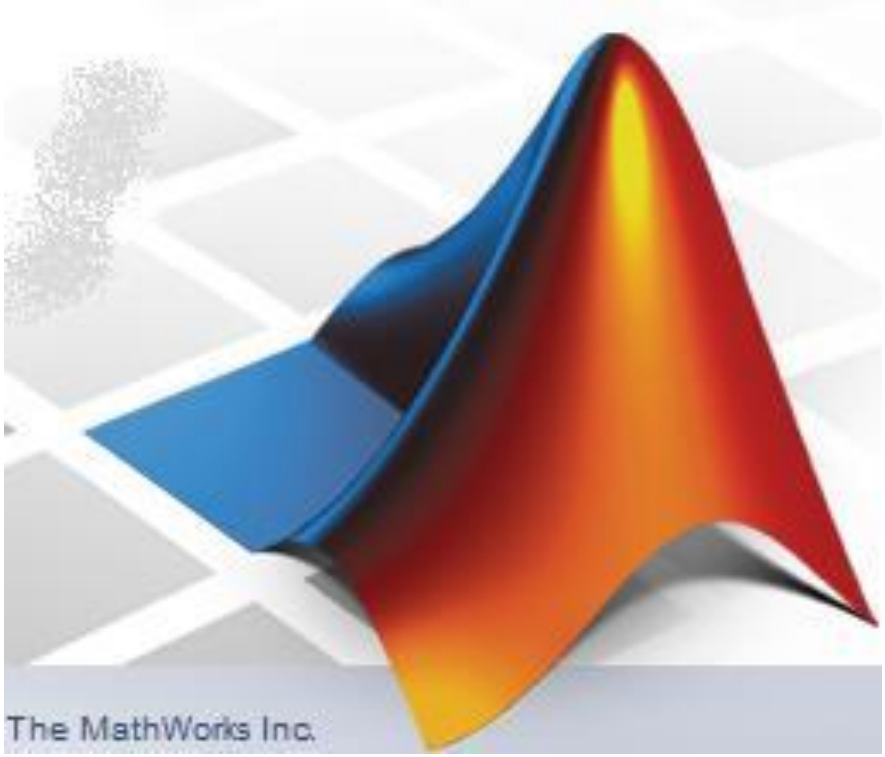
MATLAB (1) - úvod do programovania vedeckých problémov



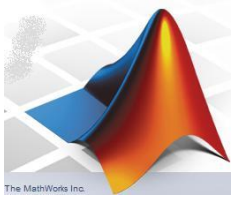
LS 2020, 5.predn.

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, jednoduchý import 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: príklady programovania úloh z oblasti prírodných vied
8. týždeň: príklady programovania úloh z oblasti prírodných vied
9. týždeň: tvorba vlastných aplikácií, práca s GUI (Graphical User Interface)
10. týždeň: tvorba vlastných aplikácií, nástroj GUIDE

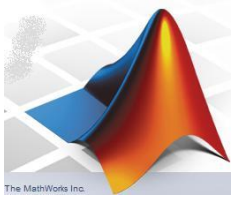
pozn.: zmeny vyhradené



Obsah (5. prednáška):

- práca s reťazcami (je potrebná pre prácu so súbormi)
- práca so súbormi

Pozn.: Reťazce sú vlastné slová a vety (postupnosti znakov).



práca s reťazcami - nájdenie časti reťazca: `strfind()`

Funkcia `strfind()` nájde časť reťazca vo väčšom reťazci a vráti jeho polohu/polohy (pri viacnásobnom výskyte) vo forme indexu (poradového čísla).

```
k = strfind(str, pattern)
```

kde `str` je väčší reťazec, `pattern` je hľadaná časť reťazca a `k` je vrátený index (poradové číslo).

Príklad:

```
S = 'Find the starting indices of the pattern string';
indx = strfind(S, 'in')
```

práca s reťazcami - porovnanie reťazcov: `strcmp()`

Funkcia `strcmp()` porovná 2 reťazce; výsledkom je 0/1 (nie sú identické/sú identické).

```
TF = strcmp(string,string)
```

kde `string,string` sú porovnávané reťazce, `TF` je skalárna premenná s výsledkom porovnania.

Príklady:

```
odp1 = strcmp('Yes', 'No')
odp2 = strcmp('Yes', 'Yes')
```

práca s reťazcami - porovnanie reťazcov: `strncmp()`

Funkcia `strncmp()` porovná prvých `n` znakov z 2 reťazcov; výsledkom je 0/1 (nie sú identické/sú identické).

```
TF = strncmp(string,string,n)
```

kde `string,string` sú porovnávané reťazce, `n` je počet prvých porovnávaných znakov, `TF` je skalárna premenná s výsledkom porovnania

Príklady:

```
odp3 = strncmp('Kansas City, KS', 'Kansas City, MO',13)
odp4 = strncmp('Kansas City, KS', 'Kansas City, MO',14)
```

práca s reťazcami - vyrezanie časti reťazca: `strtok()`

Funkcia `strtok()` vyreže časť reťazca zľava po prvý výskyt prázdneho znaku (medzery) alebo iného oddelovača. V rozšírenej verzii uloží aj zvyšok reťazca do ďalšej premennej.

Existujú jej 3 možnosti:

```
token = strtok(str)
token = strtok(str,delimiter)
[token, remain] = strtok(str,delimiter)
```

kde `str` je väčší reťazec, `delimiter` je oddelovač, `token` je vyrezaná časť reťazca a `remain` je zvyšok reťazca.

Príklady:

```
part1 = strtok('columns rows')
part1 = strtok('columns_rows','_')
[part1, part2] = strtok('columns,rows',',')
```

Pozn.: Tento príkaz využijeme neskôr pri načítaní údajov zo súborov.

práca s reťazcami- transformácia reťazca na číslo: `str2double()`

Funkcia `str2double()` transformuje reťazec s číslicami na samotné číslo. Existuje aj funkcia `str2num(C)`, ktorá to realizuje s nižšou presnosťou (iba na 4 desatinné miesta).

```
X = str2double(str)
```

kde `str` je reťazec, `X` je premenná typu double s výsledkom transformácie.

Príklady:

```
C = '123.48';
X = str2double(C)

D = 'B123.48';
Y = str2double(D)

str2double('123.45e7')
```

Pri reťazci, ktorý nepredstavuje platné číslo je výsledkom tejto operácie výraz NaN („not a number“).
Transformovať sa dajú aj reťazce s komplexnými číslami.

Pozn.: Tento príkaz využijeme neskôr pri načítaní údajov zo súborov.

práca s reťazcami- transformácia čísla na reťazec: `int2str ()`

Funkcia `int2str()` transformuje celé číslo na reťazec.

```
S = int2str(X)
```

kde `S` je reťazec, `X` je matica s celým číslom.

Príklady:

```
X = 17724;
S = int2str(X)

X = 17723.8279736256;
S = int2str(X)

X = 34.4 + 17.2i;
S = int2str(X)

X = 0/0;
S = int2str(X)
```

Pozor, reálne čísla sú zaokrúhlované ! Pri komplexných len reálna časť!

Pozn.: Tento príkaz využijeme neskôr pri zobrazovaní údajov vo Windows oknách.

práca s reťazcami- transformácia čísla na reťazec: `num2str ()`

Funkcia `num2str()` transformuje reálne číslo na reťazec.

```
S = num2str(X)
```

kde `S` je reťazec, `X` je matica s reálnym číslom.

Príklady:

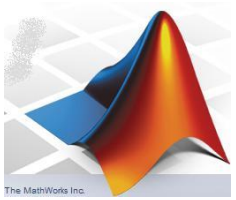
```
X = 17724;
S = num2str(X)

X = 17723.8279736256;
S = num2str(X)

X = 34.4 + 17.2i;
S = num2str(X)

X = 0/0;
S = num2str(X)
```

Reálne čísla sú zaokrúhlované na 3 desatinné čísla.

Pozn.: Tento príkaz využijeme neskôr pri zobrazovaní údajov vo Windows oknách.

Obsah (3. prednáška):

- práca s reťazcami
- práca so súbormi



načítanie matíc zo súborov (najjednoduchšie, ASCII):

využitie príkazu `dlmread()` (doteraz):

```
dlmread(názov súboru,'delimiter')
```

existuje ekvivalentný príkaz `importdata()`:

```
importdata(názov súboru,'delimiter');
```

načítanie matíc zo súborov (najjednoduchšie, ASCII):

príklad: načítanie anomálnych magnetických údajov z observatória (vzorkovaných v mesačných priemeroch)

```
clear all, close all; clc;
% nacitanie dat zo suboru do matice inpTM
inpTM = dlmread('magn_observ.dat',' ');

% stlcpe su separovane do jednoriadkovych matic T and M
T=inpTM(:,1);  M=inpTM(:,2);
% pocet nacitanych prvkov v matici T - dosadene do matice m
m = numel(T)

% vykreslenie grafu nacitanych udajov
plot(T,M);

% popis osi
xlabel('mesiac - poradove cislo','FontSize',10);
ylabel('anomalne magneticke pole [nT]','FontSize',10);
```

načítanie matíc zo súborov (najjednoduchšie, ASCII):

príklad: načítanie anomálnych magnetických údajov z observatória (vzorkovaných v mesačných priemeroch)

```
clear all, close all; clc;
% nacitanie dat zo suboru do matice inpTM
inpTM = importdata('magn_observ.dat',' ');

% stlcpe su separovane do jednoriadkovych matic T and M
T=inpTM(:,1);  M=inpTM(:,2);
% pocet nacitanych prvkov v matici T - dosadene do matice m
m = numel(T)

% vykreslenie grafu nacitanych udajov
plot(T,M);

% popis osi
xlabel('mesiac - poradove cislo','FontSize',10);
ylabel('anomalne magneticke pole [nT]','FontSize',10);
```

zápis matíc do súborov (najjednoduchšie, ASCII):

využitie príkazu `dlmwrite()` :

```
dlmwrite(názov súboru,'delimiter')
```

presný opak príkazu `dlmread()`:

Pozrite si detaily a príklady v Helpe.

načítanie matíc zo súborov (trošku zložitejšie):

ďalšie možnosti (načítania textového súboru):

`textread, textscan, ...` (viď. *help* Matlabu)

načítanie XLS súboru – príkaz `xlsread()`:

```
inp = xlsread(fname);
```

kde `inp` je matica s načítanými údajmi a `fname` je názov súboru.
Pozor! Maximálny počet načítaných riadkov je 65536 a formát je pre Excel 97 a starší .

práca s binárnymi súbormi – príkaz `fread()`:

```
A = fread(fid)
```

(viacej: *help* Matlabu)

%načíta binárne údaje zo súboru

práca so súbormi – načítavanie a ukladanie údajov

načítanie matíc zo súborov (sofistikovanejšie, ASCII):

príkazy `fopen()`, `fscan()`, `fgetl()` a `fclose()`:
(otvorenie súboru, načítanie údajov, zatvorenie súboru)

Otvorenie súboru:

```
fid = fopen(filename)
```
alebo
```
fid = fopen(filename, permission)
```

`fid` – skalárna hodnota („file identifier“),

1: úspešné otvorenie 1.súboru, 2: druhého, ...
-1: neúspešné otvorenie súboru

`permission` – argument, definujúci možnosti práce so súborom

Permission	Description
'r'	Open file for reading (default).
'w'	Open file, or create new file, for writing; discard existing contents, if any.
'a'	Open file, or create new file, for writing; append data to the end of the file.
'r+'	Open file for reading and writing.
'w+'	Open file, or create new file, for reading and writing; discard existing contents, if any.
'a+'	Open file, or create new file, for reading and writing; append data to the end of the file.
'A'	Append without automatic flushing; used with tape drives.
'W'	Write without automatic flushing; used with tape drives.

práca so súbormi – načítavanie a ukladanie údajov

po príkaze `fopen()` nasleduje samotné načítanie prvkov súboru:

Buď:

```
tline = fgetl(fid)
```

načíta textový riadok ako reťazec do premennej `tline`
(podobný je `fgets()`, ale ten načíta aj znak nového riadku /newline character/, funkcia `fgetl()` tento znak ignoruje)

Alebo:

```
A = fscanf(fid)
```

alebo

```
A = fscanf(fid, format)
```

alebo

```
A = fscanf(fid, format, size)
```

načíta formátované údaje zo súboru do matice
s definovanými rozmermi (pozor, načítava údaje po stĺpcoch!!!).

Detaily ohľadom `format` a `size` sú na ďalšom snímku.

práca so súbormi – načítavanie a ukladanie údajov

fscanf()

`format:`

znak percenta a pismenko:

```
%e, %f, %g floating-point numbers
%c sequence of characters; number specified by field width
%d base 10 integers
%i defaults to base 10 integers
%o signed octal integer
%s a series of non-white-space characters (string)
%u signed decimal integer
%x signed hexadecimal integer
```

Príklad:

```
[A,count] = fscanf(fid,'%f');
% nacita hodnoty ako realne cisla (float) do matice A

[A,count] = fscanf(fid,'%f',[ncols,nrows]);
% nacita hodnoty ako realne cisla (float) do matice A,
% pricom jej rozmery su dane vopred pomocou nrows a ncols
```

práca so súbormi – načítavanie a ukladanie údajov

fscanf()

`size:`

nacitanie do jedno alebo dvojrozmernej matice
`n alebo [m,n]`

`n`	Read at most `n` values into a column vector.
`[m,n]`	Read at most m-by-n values filling an m–by–n matrix in column order.

!

Príklad:

```
[A,count] = fscanf(fid,'%f',[columnsn, rowsm]);
```

načíta do matice `A` hodnoty z otvoreného súboru s identifikátorom `fid` ako reálne premenné (floating point numbers), pričom rozmer matice je daný premennými `rowsm` a `columnsn`.

Pozor, keďže načítanie ide po stĺpcoch (column order), poradie rowsm a columnsn je prehodené a maticu A bude treba následne transponovať !!!

načítanie matíc zo súborov (sofistikovanejšie, ASCII):

Zatvorenie súboru:

Na záver práce so súborom musí byť vždy príkaz `fclose()`:

```
status = fclose(fid)
```

alebo

```
status = fclose('all')
```

`status`: 0 (úspešné zatvorenie súboru)
-1 (neúspešné zatvorenie súboru)

práca so súbormi – načítavanie a ukladanie údajov

pozn.: Pri práci so súbormi je veľmi užitočný príkaz `uigetfile()`:

```
[fname,pname] = uigetfile('koncovka','nazov okna');
```

Slúži na na získanie mena súboru , otvorí windovsovské okno a ukazuje cestu k súboru. Po realizácii uloží v premennej `fname` názov súboru a `pname` cestu k nemu.

Príklad:

```
[fname,pname] = uigetfile('*.dat','input of data (ASCII)');
```

Ďalší užitočný príkaz `msgbox()`:

```
msgbox(str,nazov);
```

Slúži na zobrazenie určitej informácie (reťazec `str`) v okne.

Príklad:

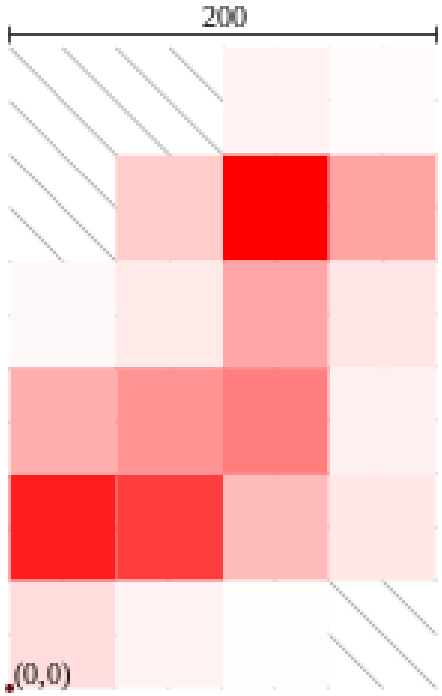
```
s = 'toto okno sluzi na zobrazenie nejakej spravy';
msgbox(s,'message');
```

práca so súbormi – načítavanie a ukladanie údajov

V prípade ukladania gridov (sieťok) z 2D dát existujú rôzne protokoly, napr. ESRI (geodézia, kartografia, GIS), VTK (medicína, biológia), SURFER (geofyzika, geovedy), ... atď.

ESRI (ASCII) protokol:

Grid image



Values

	25	75	125	175
275	NA	NA	5	2
225	NA	20	100	36
175	3	8	35	10
125	32	42	50	6
75	88	75	27	9
25	13	5	1	NA

ASCII grid format

```
ncols          4
nrows          6
xllcorner      0.0
yllcorner      0.0
cellsize       50.0
NODATA_value   -9999
-9999 -9999 5 2
-9999 20 100 36
3 8 35 10
32 42 50 6
88 75 27 9
13 5 1 -9999
```

ESRI (ASCII) protokol – príklad väčšieho gridu:

Lister - [c:\Roman\skola\prednasky\Matlab\x_Matlab pre doktorandov\podklady rozne\ESRI grid\Helens ESRI ASCII.grd]

Súbor Upraviť Možnosti Kódovanie Pomocník

```
ncols 327
nrows 466
xllcorner 557805
yllcorner 5107995
cellsize 30
nodata_value 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
1.70141000000000001E+038 1.70141000000000001E+038 1.70141000000000001E+038
```

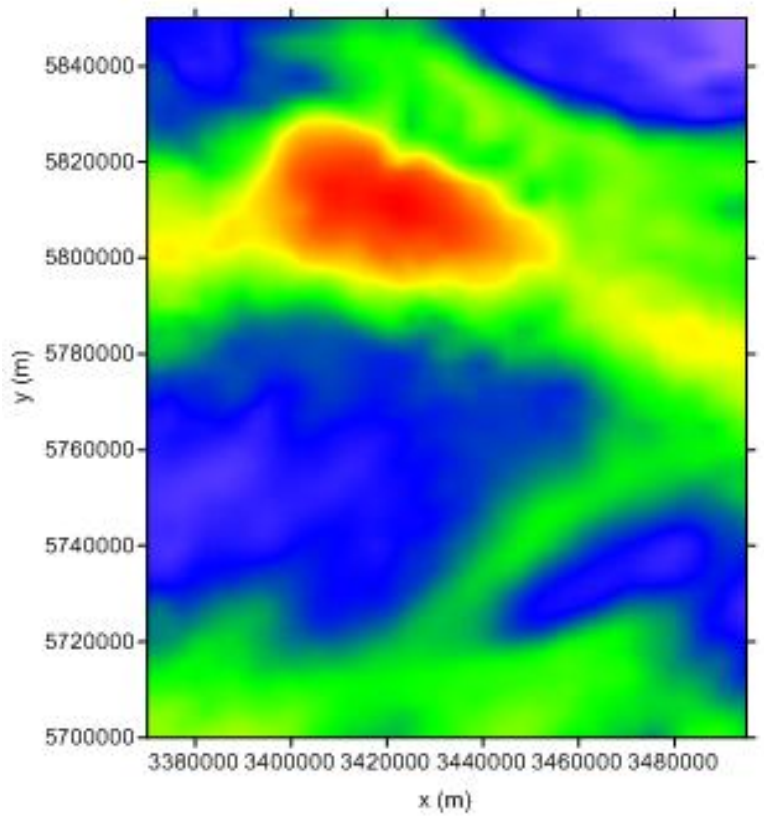
načítanie ESRI gridu (ASCII formát) zo súboru do matice:

```
close all; clear all; clc;
[fname,pname] = uigetfile('*.grd','input of data (ESRI ASCII *.grid file)');
fid = fopen(fname,'r');
%reading of the ESRI ASCII grid header (ncols, nrows, xllcorner, yllcorner, cellsize,
row1 = fgetl(fid); [token,rem] = strtok(row1); ncols = str2double(rem)
row2 = fgetl(fid); [token,rem] = strtok(row2); nrows = str2double(rem)
row3 = fgetl(fid); [token,rem] = strtok(row3); xllcorner = str2double(rem)
row4 = fgetl(fid); [token,rem] = strtok(row4); yllcorner = str2double(rem)
row5 = fgetl(fid); [token,rem] = strtok(row5); cellsize = str2double(rem)
row6 = fgetl(fid); [token,rem] = strtok(row6); nodata_value = str2double(rem)
%reading of the ESRI ASCII grid data
[A,count] = fscanf(fid,'%f',[ncols,nrows]); %fscanf() works in column order - columns
field = flipud(A'); %readed matrix A must be transponed because function fscanf() wor
                    % and fliiped up and down, because ESRI starts with rows in upper
status = fclose(fid);
%generating the x_cor and y_cor vectors
x_cor = xllcorner:cellsize:(ncols-1)*cellsize+xllcorner;
y_cor = yllcorner:cellsize:(nrows-1)*cellsize+yllcorner;
%plotting the contourf() map
figure, contourf(x_cor,y_cor,field)
%information for the user
s = [' size of readed grid: ' int2str(nrows) ' rows x ' int2str(ncols) ' columns'];
msgbox(s,'message');
```

Vyskúšajte si skript `nacitanie_ESRI_grid.m` a s ním načítanie a zobrazenie súboru „test ESRI1.grd“ .

práca so súbormi – načítavanie a ukladanie údajov

Príklad – súbor „test ESRI2.grd“ s údajmi anomálneho gravitačného poľa na území cca 10 x 15 km (severné Nemecko).

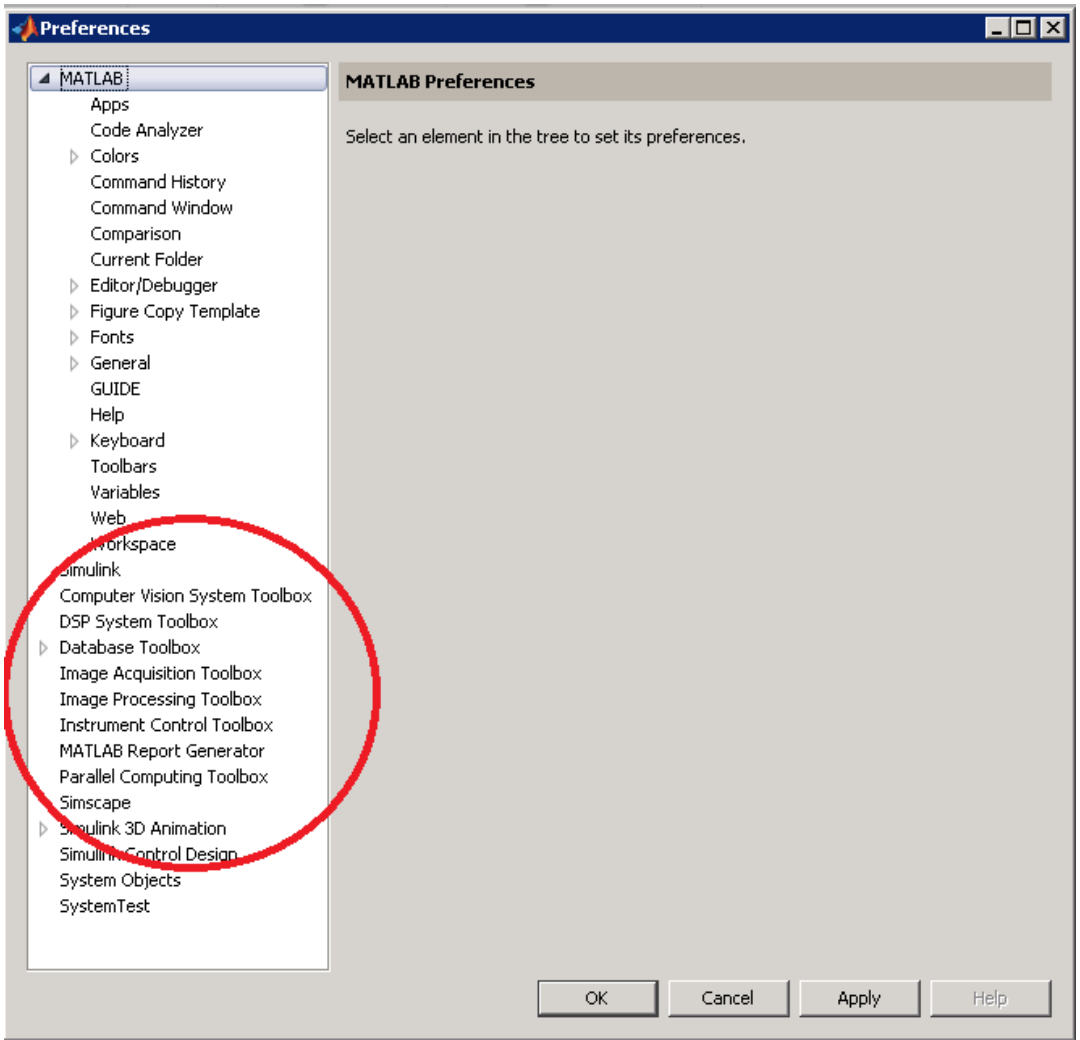


(táto farebná mapa bola vytvorená iným softvérom – GS Surfer)

Vyskúšajte si skript `nacitanie_ESRI_grid.m` a s ním načítanie a zobrazenie súboru „test ESRI2.grd“ .

práca so súbormi – načítavanie a ukladanie údajov

Pre načítanie ESRI ASII gridu existuje aj špeciálna funkcia `arcgridread(,nazov.grd')` v Mapping toolbox,
ale žiaľ túto nemáme k dispozícii v tom prístupe cez CVTI server.



zápis matíc do súborov (sofistikovanejšie, ASCII):

Opäť príkazy `fopen()(s premission: 'w')` a `fclose()`:
ale pracuje sa s príkazom na zápis:

`fprintf()`

Príklad:

```
close all; clear all; clc;
A = [3; 6; 8; 2; 1; 9; 10; 23; 2; 3; 78; 12; 3; 18; 5; 25];
B = [9; 0; 1; 3; 2; 8; 12; 34; 3; 8; 88; 32; 9; 12; 9; 35];
savearr(:,1) = A; savearr(:,2) = B;
savefile = fopen('check.dat','w');
fprintf(savefile,'%s %s','matica_A','matica_B');
fprintf(savefile, '\n%i %i', savearr');
fclose(savefile);
```

Vyskúšajte si rôzne formáty zápisu samotných hodnôt matíc:

`%i` – zápis celého čísla (integer)
`%f` – zápis reálneho čísla (float)
`%e` – zápis reálneho čísla v exponenciálnom tvare (exponential)
`%g` – zápis reálneho čísla vo všeobecnom tvare (general)
`%s` – postupnosť znakov bez medzier (reťazec = string)
`\n` – znamená, že zakaždým bude zápis realizovaný v novom riadku

práca so súbormi – načítavanie a ukladanie údajov

zápis matíc do súborov (sofistikovanejšie, ASCII):

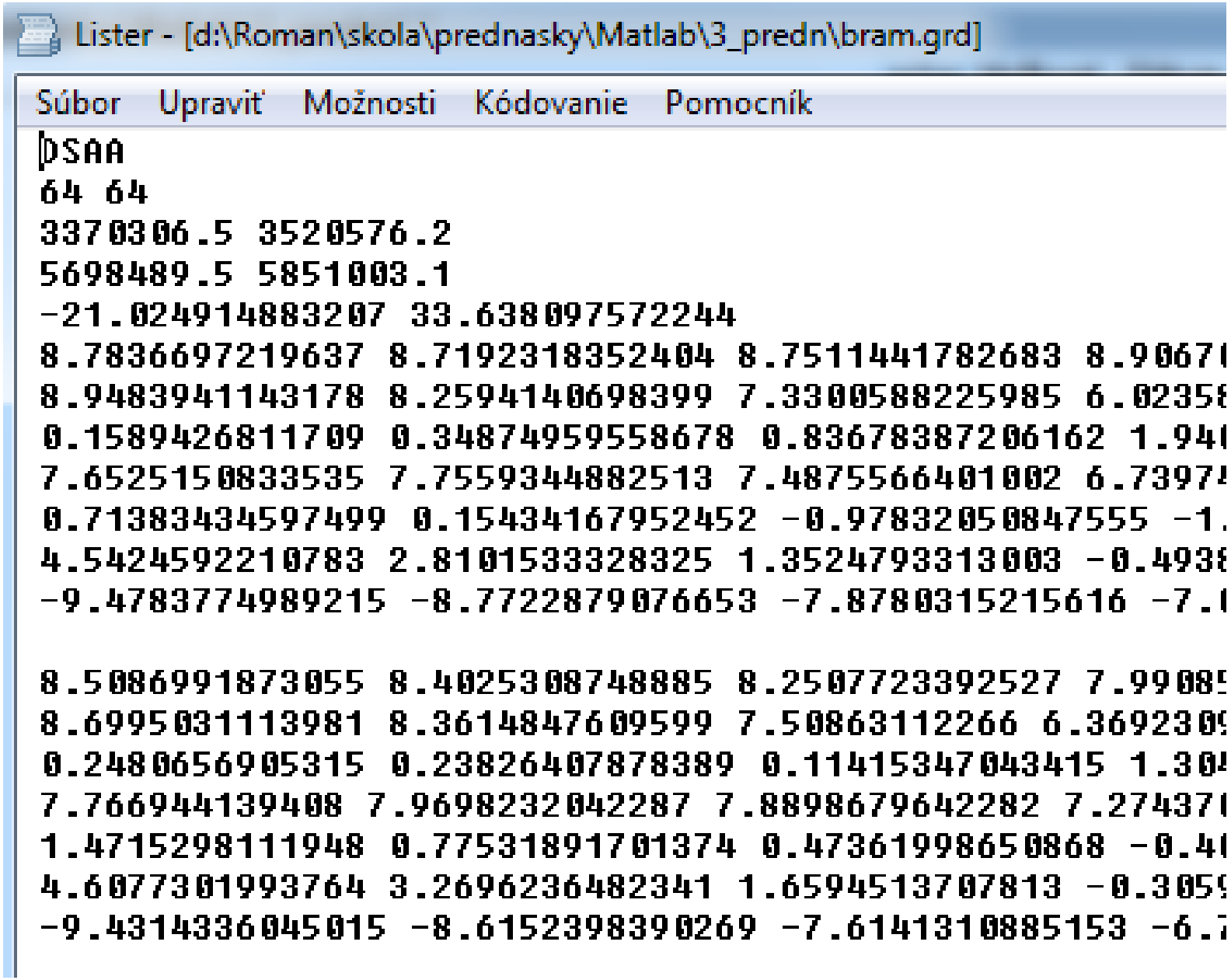
Opäť príkazy `fopen()(s premission: 'w') a fclose()`:
ale pracuje sa s príkazom na zápis:

`fprintf()`

Príklad (zápis hlavičky súboru ESRI ASCII):

```
close all; clear all; clc;
ncols=100; nrows=200; xllcorner=500100; yllcorner = 5304000;
cellsize=50; nodata_value=1.70141000000000001E+038;
savefile1 = fopen('ESRI_header.grd','w');
fprintf(savefile1,'%s %i','ncols', ncols);
fprintf(savefile1,'\n%s %i','nrows', nrows);
fprintf(savefile1,'\n%s %f','xllcorner', xllcorner);
fprintf(savefile1,'\n%s %f','yllcorner', yllcorner);
fprintf(savefile1,'\n%s %f','cellsize', cellsize);
fprintf(savefile1,'\n%s %e','nodata_value', nodata_value);
fclose(savefile1);
```

Keby sme mali dané pole (2D maticu) na zápis, mohli by sme pokračovať príkazom: `fprintf(savefile1,,%f ',pole);`



iný príklad formátu gridu (Golden Software Surfer)

iný príklad formátu gridu (Golden Software Surfer)

```
close all; clear all; clc;
rowsm = 0; columnsn = 0; minx = 0; maxx= 0; miny = 0; maxy= 0; minf = 0; maxf
[fname,pname] = uigetfile('*.grd','input of data (Surfer ASCII *.grid file)')
fid = fopen(fname,'r');
% reading of the GS ASCII grid header (rows, columns, minx, maxx, miny, maxy,
tline = fgetl(fid); row1 = tline; %reading the first line - the identificator
tline = fgetl(fid); row2 = tline; [token,rem] = strtok(row2); %reading the se
columnsn = str2double(token); rowsm = str2double(rem);
tline = fgetl(fid); row3 = tline; [token,rem] = strtok(row3); %reading the th
minx = str2double(token); maxx = str2double(rem);
tline = fgetl(fid); row4 = tline; [token,rem] = strtok(row4); %reading the fo
miny = str2double(token); maxy = str2double(rem);
tline = fgetl(fid); row5 = tline; [token,rem] = strtok(row5); %reading the th
minf = str2double(token); maxf = str2double(rem);
% reading of the main field of the grid - into the matrix A
[A,count] = fscanf(fid,'%f',[columnsn, rowsm]); %fscanf() works in column ord
status = fclose(fid);
field = A'; %readed matrix A must be transponed because function fscanf() wor
s = [' size of readed grid: ' int2str(rowsm) ' rows x ' int2str(columnsn) ' c
msgbox(s,'message');
figure, contourf(field); colorbar;
```

Zadanie č.5:

Vyskúšajte načítať údaje o výške terénu (reliéf) zo súboru „Helens ESRI ASCII.grd“, skúste údaje výšok nejakým spôsobom matematicky upraviť (umocniť, odmocniť, vydeliť určitým číslom,...), prípadne maticu s týmto poľom výšok preklopiť, rotovať, ….
potom zobraziť ako `contourf()`
a napokon zapísať ako nový súbor vo formáte grid ESRI ASCII.

Pozn.: Pri tvorbe exportu do formátu grid ESRI ASCII vám môže poslúžiť aj skript zo snímku č. 29 dnešnej prednášky.
Správnosť exportu matematicky upravených hodnôt výšok do nového súbor vo formáte grid ESRI ASCII si môžete skontrolovať opäť pomocou skriptu z dnešnej prednášky.

Zadanie č.5:

Príklad správneho výsledku tohto zadania
(pri umocnení pôvodných údajov, bez preklopenia alebo rotácie):