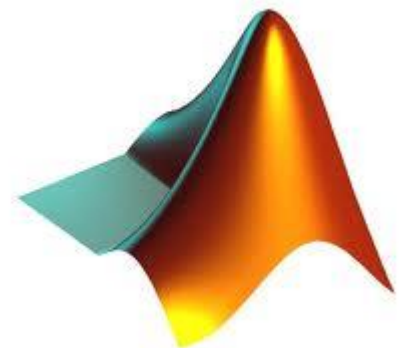


# Obrázky v MATLABe GUI

Cvičenia z Počítačového Videnia I.

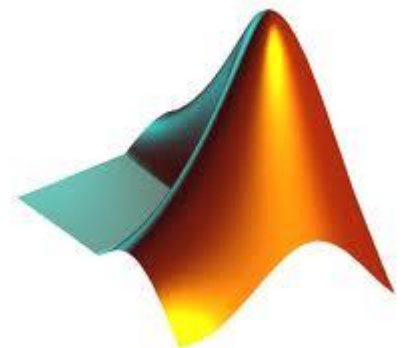
# Opakovanie

- vektory  $v = [1,2,3,4]$
- matice  $m = [v; v^*2; v.^2]$   
 $m = \text{eye}(3);$
- $[\text{riadky}, \text{stlpce}] = \text{size}(m);$
- $M = m(:);$



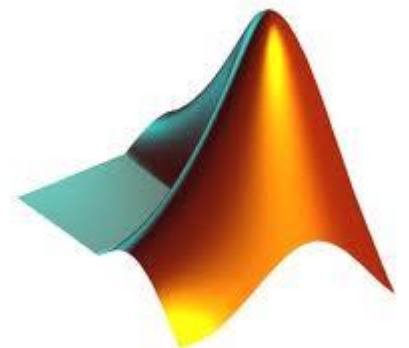
# Opakovanie

- Case Sensitive
  - B a b sú rôzne premenné!
- premenná ans
- ; za príkazom zabráni výpisu na obrazovku



# Opakovanie

- volanie fcií: `k = sin(pi/2)`
- grafy: `plot(x,y); grid;`
- `bar(x);`
- výpis:
  - `fprintf('hodnota z je %g', z);`
  - `disp(['hodota z je ', num2str(z)]);`
- `['hodnota z je', num2str(z)];`
- `str = 'string';`
- `s = length(str);`



# Opakovanie

- fcie sa píšu ako M-files

- fcia mocnina.m:

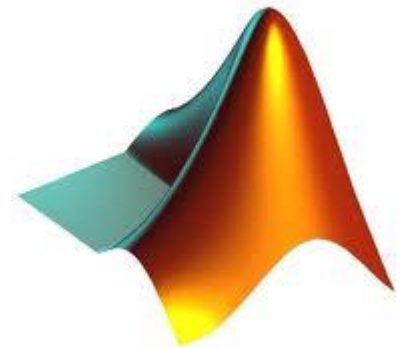
```
function y = mocnina(x)
```

```
%fcia vráti druhú mocninu
```

```
y = x^2
```

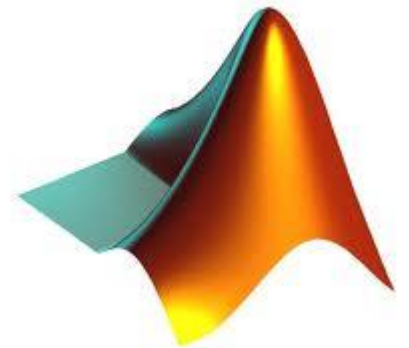
- viac výstupných hodnôt:

```
function [y,z,p] = prvy(x)
```



# Obrázky

- zobrazovanie matíc ako obrázkov
- zobrazenie obrázkov ako matíc



# Obrázky

- zobrazovanie matíc ako obrázkov

```
k = 1:20;
```

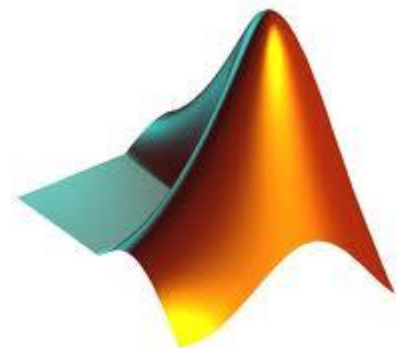
```
M = ones (20);
```

```
M (:,4) = k;
```

```
M(17,:) = k*2;
```

```
M(2,:) = k*4;
```

```
image(M);
```



# Zobrazovanie obrázkov

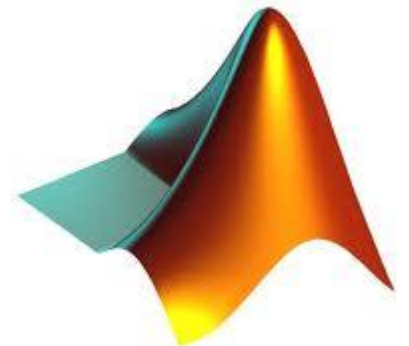
- `image(M);`

- priamo v matlabe
- farby zobrazovaného obrazu vôbec nemusia zodpovedať reálnym farbám

- `imshow(M);`

- IPT
- predpokladá, že zobrazované hodnoty sú intenzity pixlov

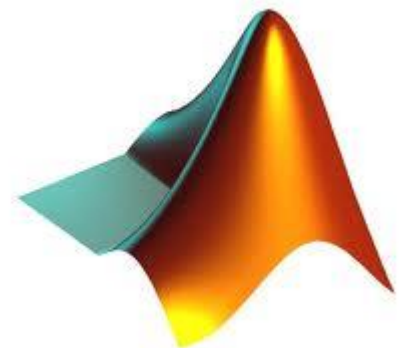
- `figure;`





# Zobrazovanie obrázkov - rozdiel

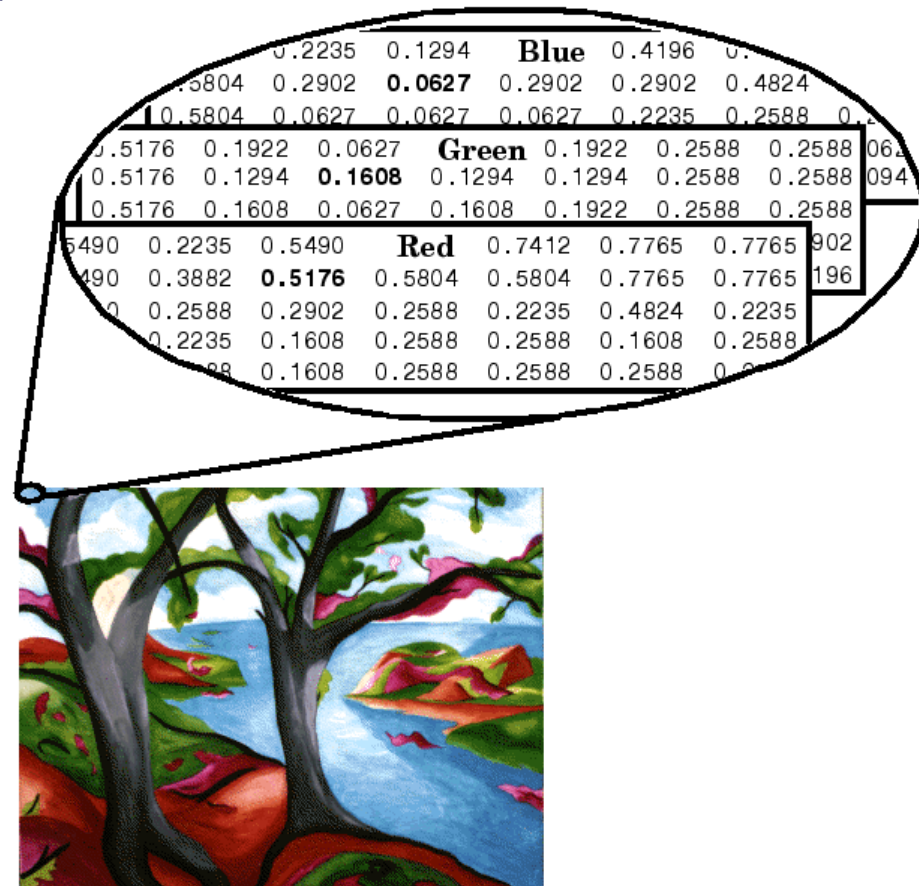
```
img = imread('cameraman.tif');  
figure;  
subplot(1,2,1);  
image(img); %axis off; axis image;  
subplot(1,2,2);  
imshow(img);
```



# Typy obrázkov: True Color vs. Indexed Images

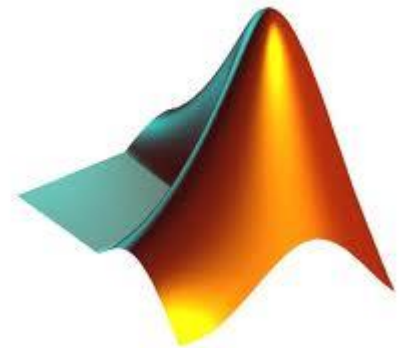
## True color:

- obrázok veľkosti MxN je uchovaný v 3-rozmernom poli
- M x N x 3 (RGB hodnoty)



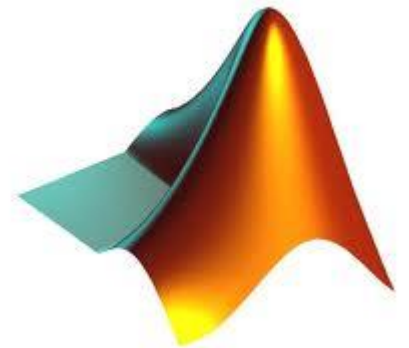
# True color

- 3 hodnoty RGB môžu byť typu:
  - **Double precision** (double array)  
rgb v rozsahu [0, 1]
  - **8-Bit Data** (uint8)  
rgb v rozsahu [0, 255]
  - **16-Bit Data** (uint16)  
rgb v rozsahu [0, 65535]



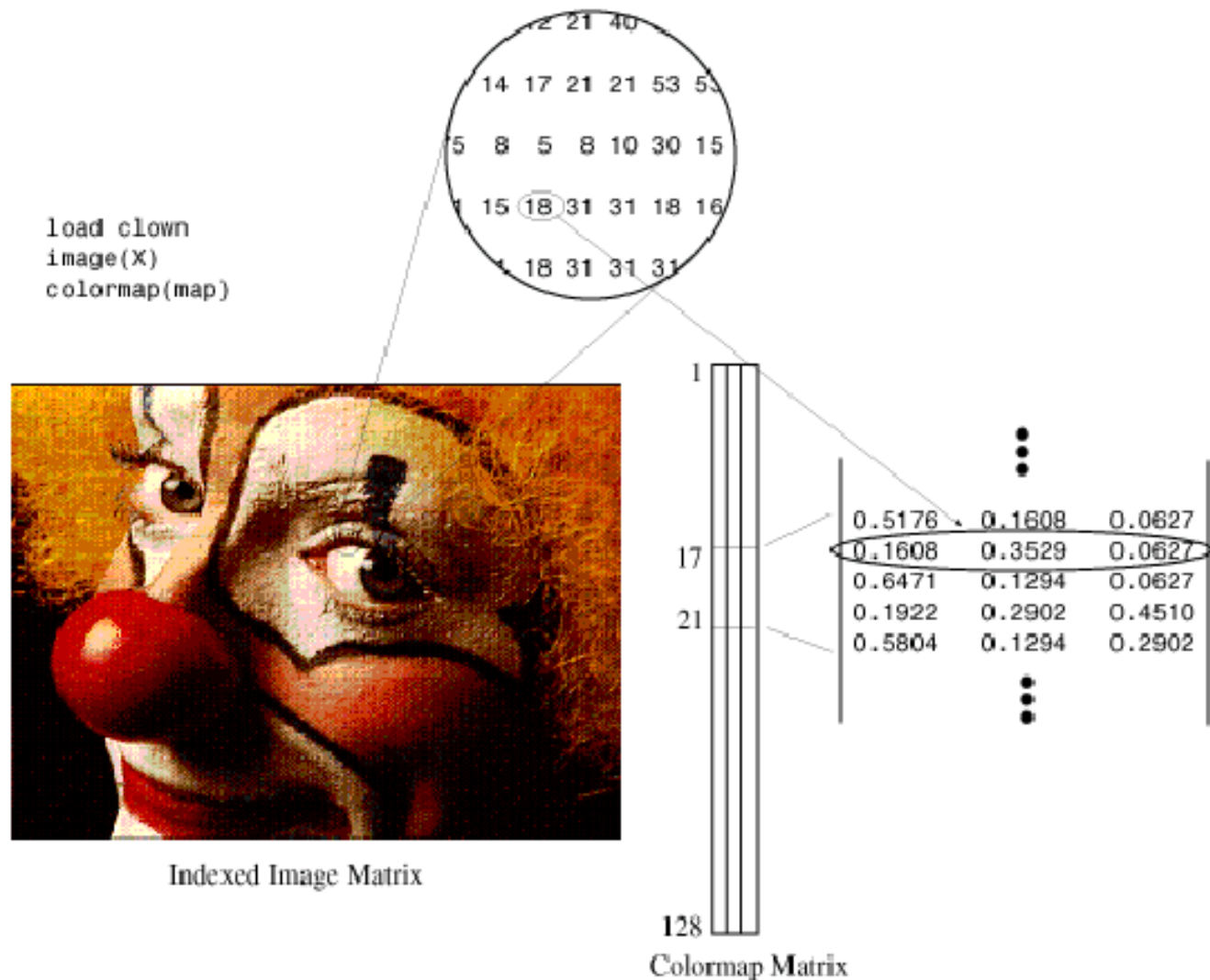
# Read True color image

```
im = imread('nazov.jpg');  
imshow(im);
```



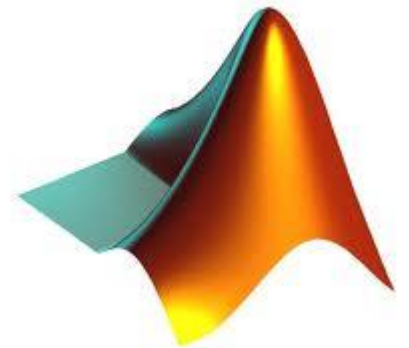
# Indexed color

- obrázok  $M \times N$  je uchovaný v poli  $M \times N$
- colormap je pole



# Read Indexed image

- `load Durer`
- vytvorí sa `X` (pole  $M \times N$ ) a `map` (`colormap`)
- `X` je typu `DOUBLE` teda `1 - length(map)`
- `imshow(X);`



# Colormap

hodnoty v intervale [0,1]

```
colormap(map);
```

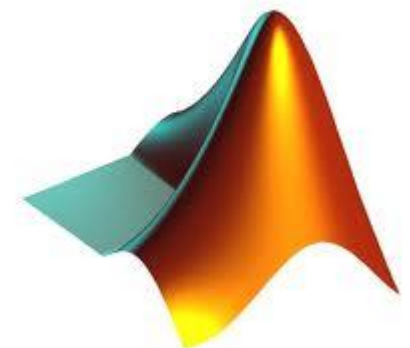
```
colormap(hsv(128));
```

load clown

```
imshow(X,colormap(map));
```

```
imshow(X,colormap(jet));
```

```
imshow(X,colormap(spring));
```



# Indexed color

podľa typu dát:

- Double precision (Double array):

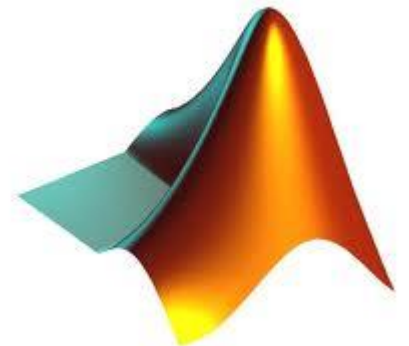
pole  $M \times N$  je pole int v  $[1, \text{length}(\text{colormap})]$

Colormap je pole float v  $[0,1]$

- 8 Bit (uint8):

pole  $M \times N$  je pole int v  $[0, 255]$

Colormap je pole float v  $[0,1]$





# Indexed color

- 16 Bit (uint16):

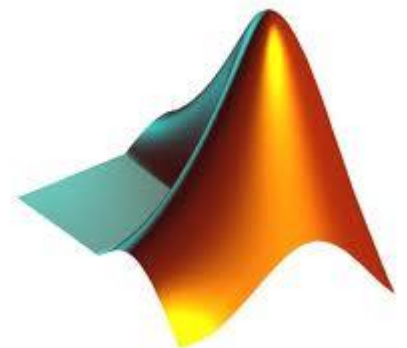
Pole  $M \times N$  je pole int v  $[0, 65535]$

Colormap je pole float v  $[0,1]$

- posun v indexovaní:

Double: 1 == prvý riadok v colormap

Int: 0 == prvý riadok v colormap



# Konverzie

```
im = imread('nazov.jpg');
```

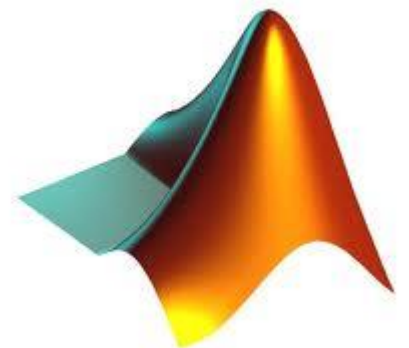
```
imshow(im);
```

```
[X,map] = rgb2ind(im, 5);
```

```
load clown
```

```
imshow(X,colormap(map));
```

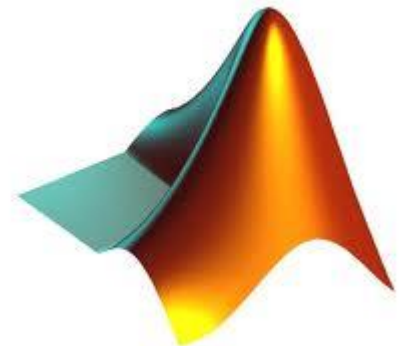
```
RGB = ind2rgb(X,map);
```



# Supported files

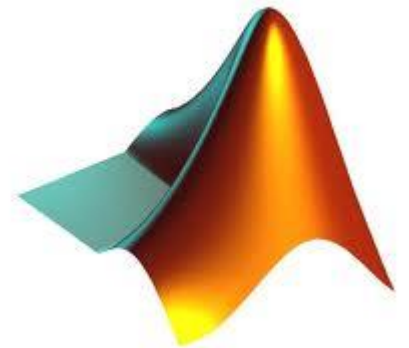
MATLAB podporuje formáty

- BMP, JPG, PNG, TIFF, GIF
- JPEG 2000 formáty: JP2, JPX...
- Iné: PNM, PCX, ICO, PBM, HDF...



# Images and Colormaps

- >> demo
  - Graphics/
  - Examples of Images and Colormaps



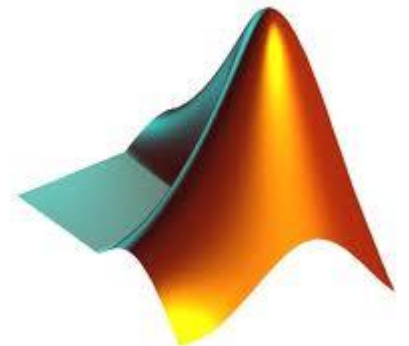
# Uloženie obrázka

```
imwrite(I, filename, fmt)
```

```
imwrite(X, map, filename, fmt)
```

```
imwrite(X, colormap, 'obrazok.jpg', 'jpg')
```

<http://www.mathworks.com/help/techdoc/ref/imwrite.html>



# Prahovanie obrázka

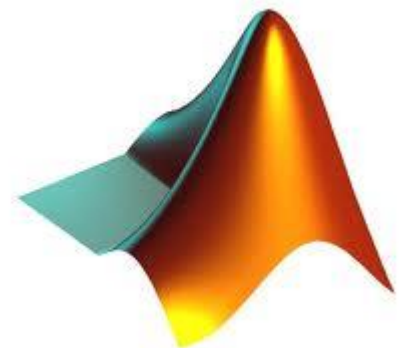
- Prevod na šedotónový

```
A = rgb2gray(I);
```

- Prahovanie

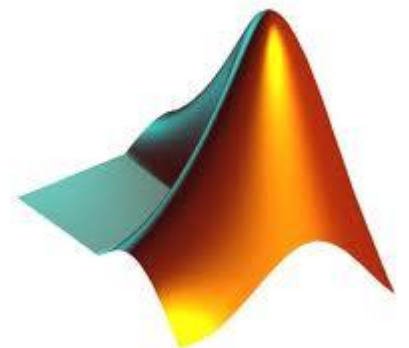
```
A = I > 200;
```

```
imshow(A);
```



# Histogram

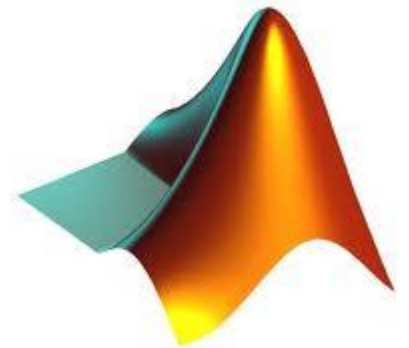
- `hist(v)`;  $v$  je vektor
- `hist(M)`;  $M$  je matica
- 2-D alebo 1-D
- pri 2-D vráti histogram pre každý stĺpec spoločne v jednom grafe
- šedoúrovňový



# Histogram

`imhist(I); imhist(I, n); imhist(X, map)`

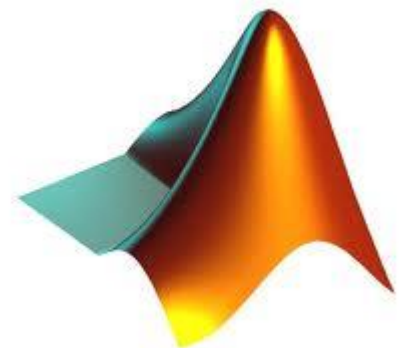
- `I` je obrazok, `n` počet chlievikov
- `J = histeq(I, n)`
- `J = imadjust(I)`
- `J = imadjust(I,[low_in; high_in],  
[low_out;high_out])`





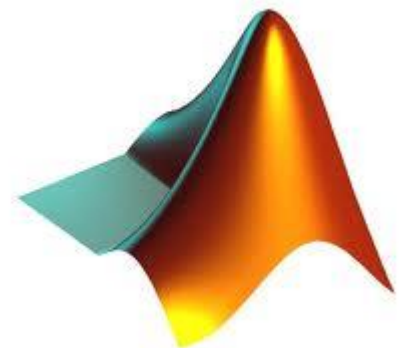
# Histogram

```
I = imread('pout.tif');  
J = imadjust(I);  
imshow(I), figure, imshow(J);  
K = imadjust(I,[0.3 0.7],[]);  
figure, imshow(K);
```



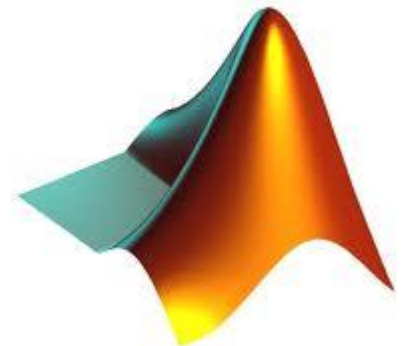
# GUI

- `>> guide`
- Blank GUI
- Vytvorí dva súbory:
  - `mojogui.fig`
  - `mojogui.m`



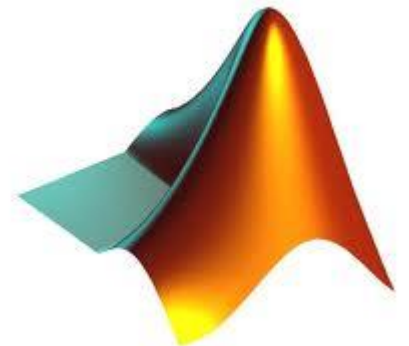
# GUI

- GUI objekty:
  - Button, radio button, check box, slider
  - Edit text, Static text
  - Axes
  - Pop-up menu, list box
  - Panel, Button group...



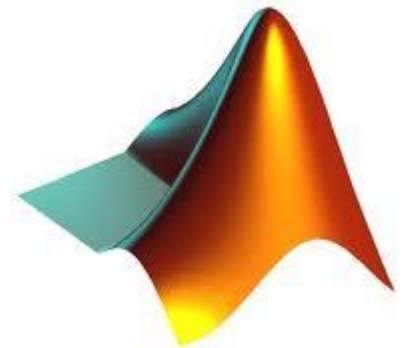
# GUI

- Property Inspector
  - Color, text, name, position, opacity
- Handles
  - `set(handles.text2, 'Visible', 'on');`
  - `g = get(handles.radiobutton1, 'Value');`



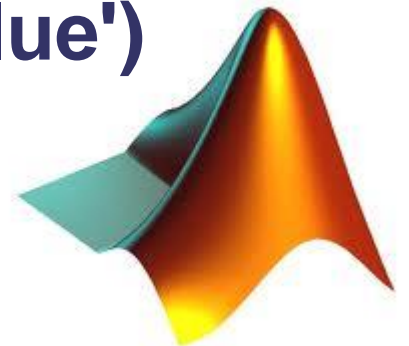
# GUI

- Callbacks
  - Callback
  - ButtonDownFcn
  - KeyPressFcn
  - CreateFcn
  - ...



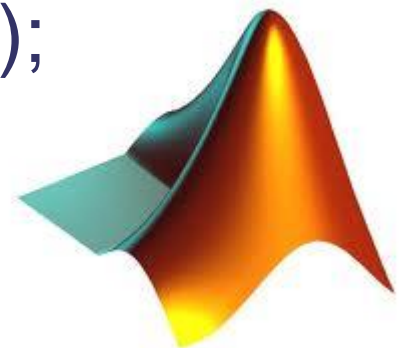
# GUI

- Callbacks
  - Callbacks fungujú ako funkcie,
  - ak chceme využívať v jednom callbacku premennú ktorú sme vytvorili v inom, musíme použiť funkcie get a set
  - V get a set sa dajú okrem štandardných parametrov ako `get(handles.edit1, 'Value')` využívať tzv. 'UserData'



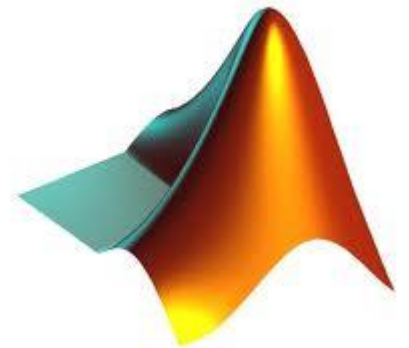
# GUI

- UserData
  - môžeme vložiť ľubovoľné dáta (obrázok, číslo)
- V jednom callbacku načítame
  - `RGB = imread('1.jpg');`
  - `set(handles.pushbutton1, 'UserData', RGB);`
- V druhom zavoláme
  - `I = get(handles.pushbutton1, 'UserData');`
  - `imshow(I);`



# GUI

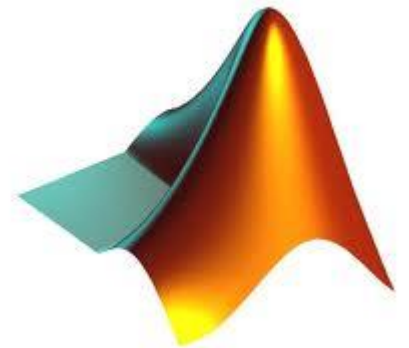
- Načítanie obrázku
  - `[FileName,PathName]=uigetfile('* .jpg','Vyber .jpg');`
  - `I = imread(fullfile(PathName, FileName));`
  - `figure; imshow(I);`
- Načítanie údajov
  - `gg = get(handles.edit1,'Value')`





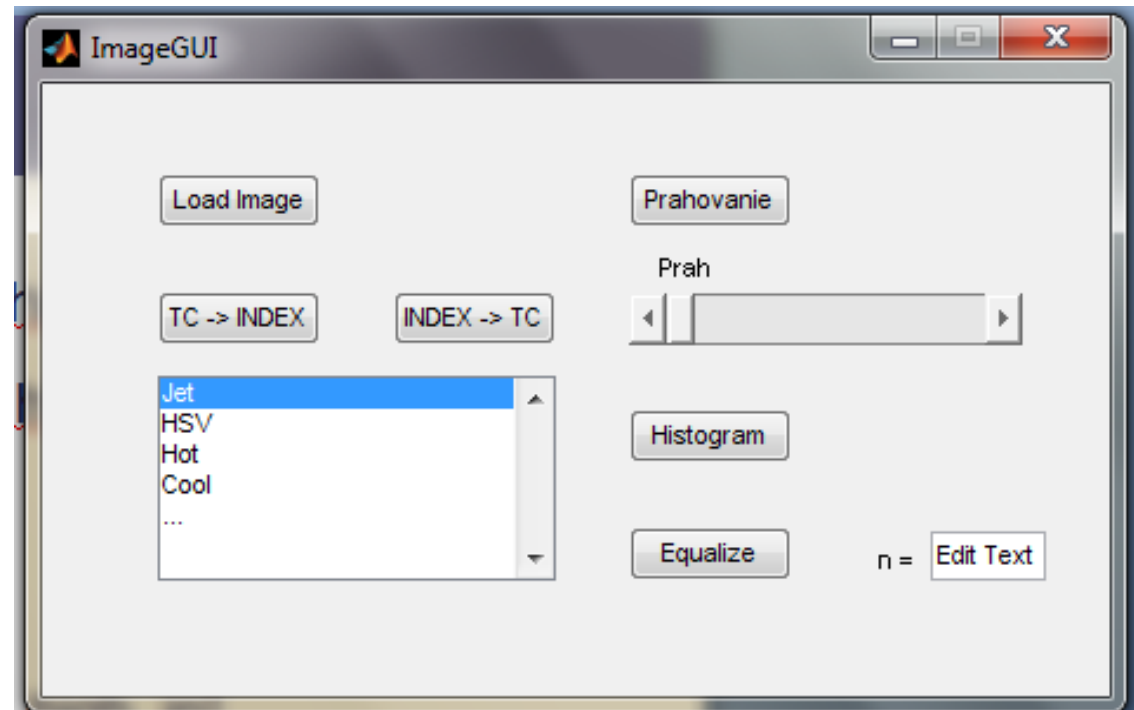
# GUI

- ListBox
  - Pridávanie Probert. Inspec./ String
  - Max, Min počet vybraných položiek
  - `s = get(handles.listbox, 'value');`  
(1,2,3)
- PopUp Menu
- Podobne ako ListBox



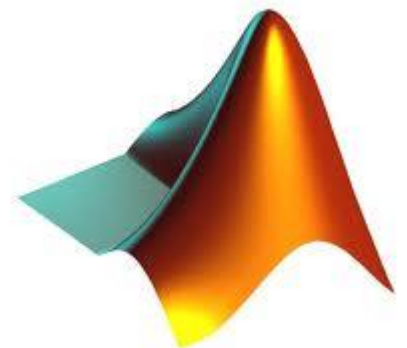
# Úloha 1

- Do 18.10 23:59
  - ImageGUI.fig, ImageGUI.m
  - Email: [DUvidenie@gmail.com](mailto:DUvidenie@gmail.com)
  - Subjekt správy: CV1 DU1



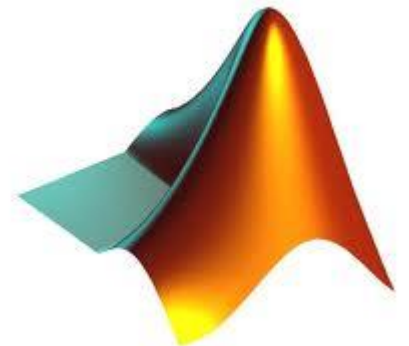
# Úloha 1

- Musí spĺňať
  - Načítanie obrázka z adresárovej štruktúry pomocou dialógového okna
  - Konverzia z truecolor na indexovaný
    - Možnosť vybrať colormap
    - aspoň 4



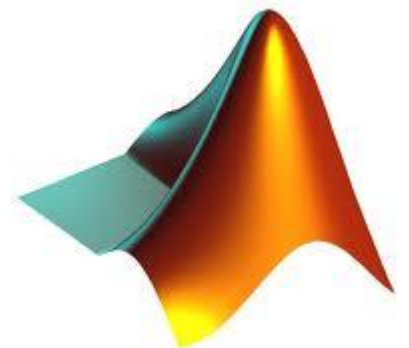
# Úloha 1

- Prahovanie obrázka
  - prevod na šedoúrovňový
  - možnosť výberu prahu (thresholdu)  
==> binárny obrázok
- Vytvorenie Histogramu Obrázka
  - Ekvalizácia histogramu



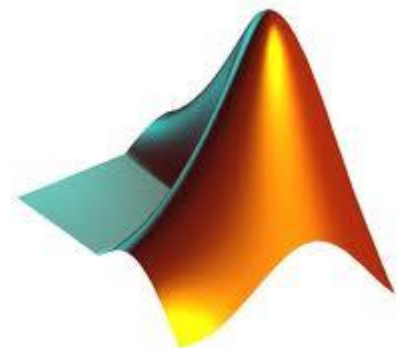
# Úloha 1

- Typy
  - Môžete si vytvoriť GUI aké chcete
  - Namiesto ListBox ==> PopUpMenu, Edit, Slider...
- Kto urobí niečo navyše dostane bonusové body



# Úloha 1

- Typy:
- Vykresľujte obrázky pomocou
  - `figure; imshow();`



# Užitočné skratky

- Ctrl r
  - zakomentovať vyznačené
- Ctrl t
  - odkomentovať vyznačené
- Ctrl c
  - prerušiť bežiaci program/príkaz
- why

