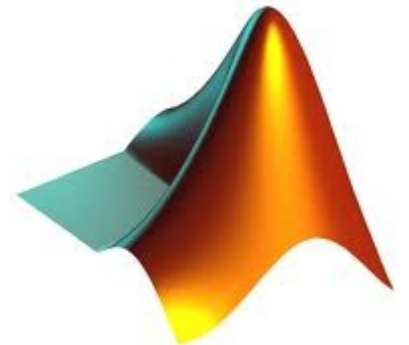


Úvod do MATLAB-u

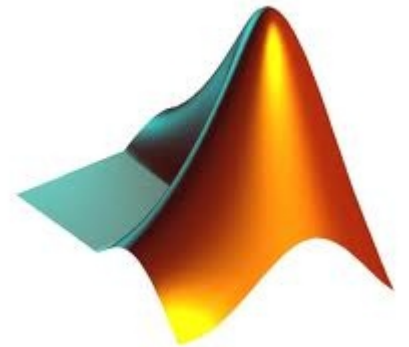
MATLAB

- **Pôvodne:** Interaktívny program na operácie s maticami
- **Teraz:** Vysoko úrovňový jazyk na technické výpočty a interaktívne prostredie na:
 - tvorbu algoritmov, vizualizáciu a analýzu dát a numerické výpočty



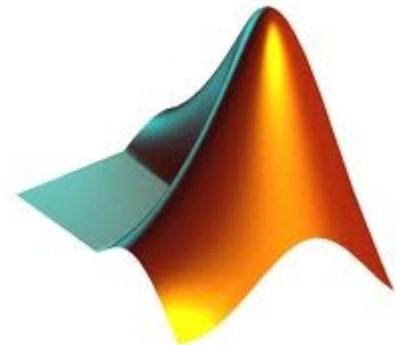
MATLAB - functions

- Function list
 - <http://www.mathworks.com/help/matlab/functionlist.html>
- Tutorial:
 - http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf

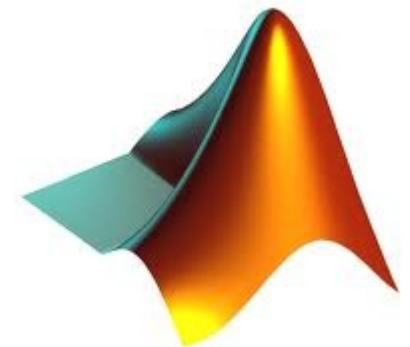
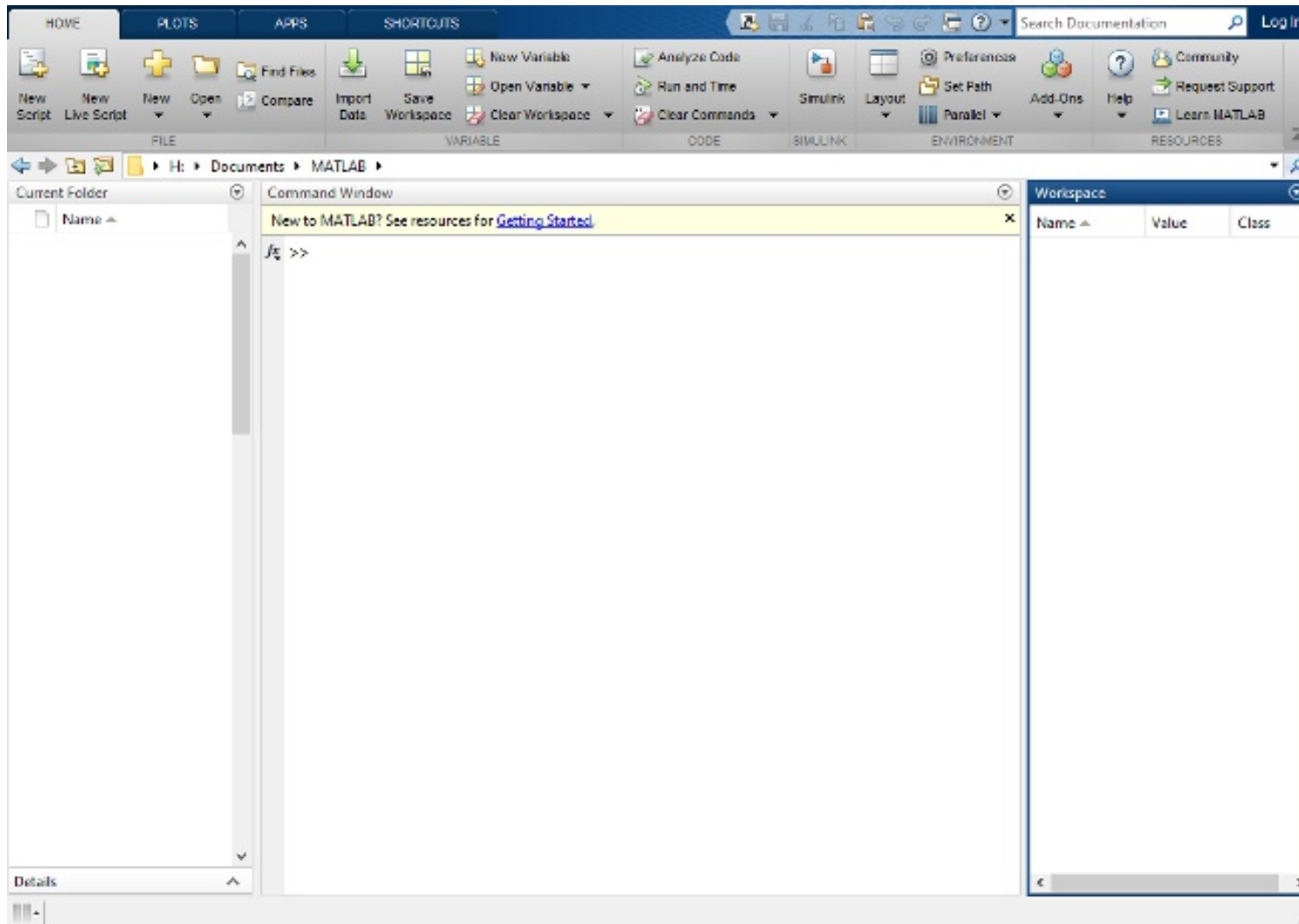


Toolbox

- Image Processing Toolbox
- Computer Vision Toolbox
- Iné Toolboxy
 - Statistics, Bioinformatics, Wavelet, Fuzzy Logic, Econometrics

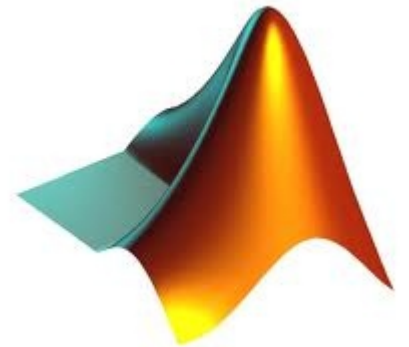


MATLAB prostredie



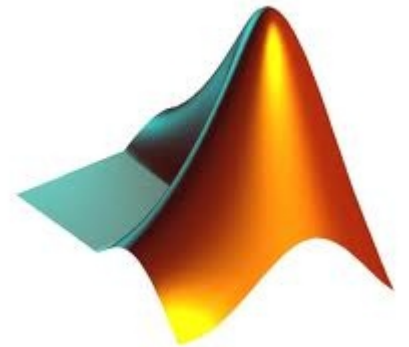
MATLAB prostredie

- Command window
 - písanie príkazov, výstupy, chyby
- Workspace
 - premenné, ich hodnoty a typy
- Command History
 - použité príkazy sa dajú „drag and drop“ do command window



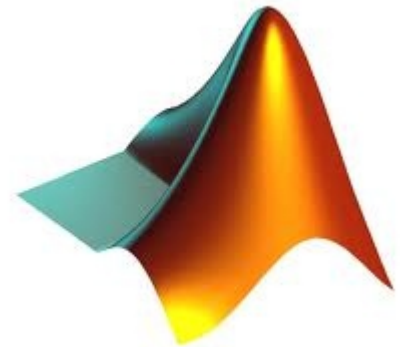
Demá

- >> demo
 - Záložka Demos
 - 3D Visualisation
 - Teapot, Images and Colormaps
- >> help commandname
- >> lookfor keyword



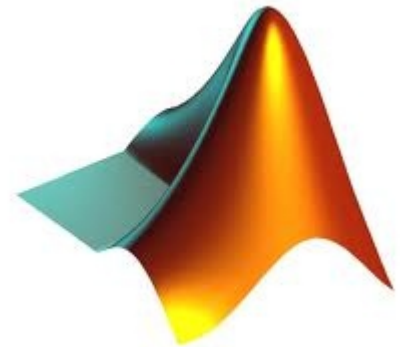
Command window

- $3 + 4 - 7$
- $t = 3 + 4 - 7$
- $k = 3 + 4 - 7;$
- k
- $k;$
- $3^2 * 4$
- $2 + 2 / 1 + 1$



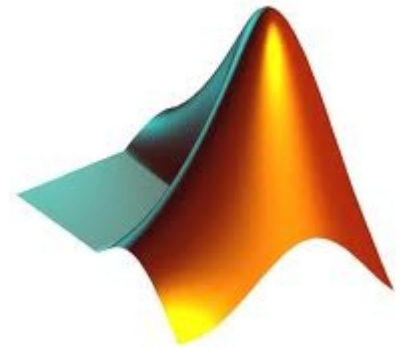
Command window

- 1/0 (Inf)
- 0/0 (NaN)
- **MATLAB je Case Sensitive!**
- K a k sú rôzne premenné
- 15 miest, ale ukazuje len 5
- format long / format short



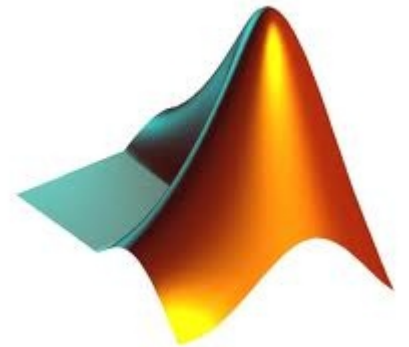
Command window

- MATLAB má množství vstavaných funkcí
- \sin , \cos , \tan , asin , acos
- $\sin(\pi/2)$
- \log , \log_{10} , \log_2
- $\log_{10}(100)$



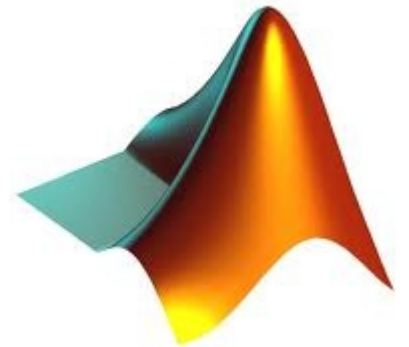
Vektory v MATLAB-e

- $v = [1, 2, 3, 4]$
- $v = [1\ 2\ 3\ 4]$
- $v = [1; 2; 3; 4]$
- $v = \text{start: step: end}$
- $v = 2:2:9$
 - $v = [2, 4, 6, 8]$
- $v = 2:5$
 - $v = [2, 3, 4, 5]$



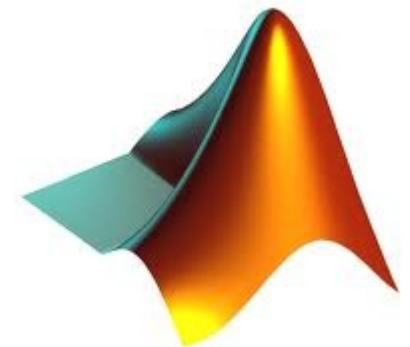
Vektory v MATLAB-e

- $v = \text{linspace}(1, 5, 10)$
- $v(4) = 0$
- $v(5:7) = 0$
- $v(1:2:7) = 0$



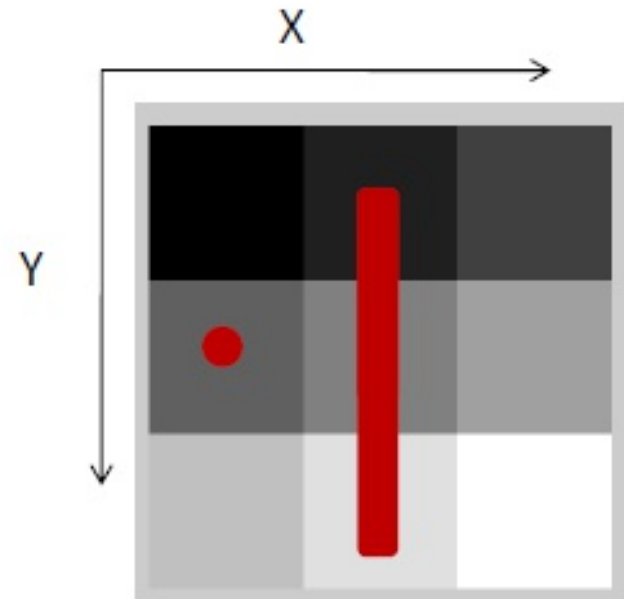
Matice v MATLAB-e

- vytvorenie
 - $A=[1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9];$
 - 3×3
- špeciálne:
 - `zeros()`, `ones()`, `eye()`, `rand()`, `randn()`, `magic()`
 - `p = zeros(3, 3) == zeros(3);`
 - `o = ones(3, 3) == ones(3);`
 - `r = rand(3, 3) == rand(3);`
 - `r1 = randn(1,10);`
 - `k = magic(3);`

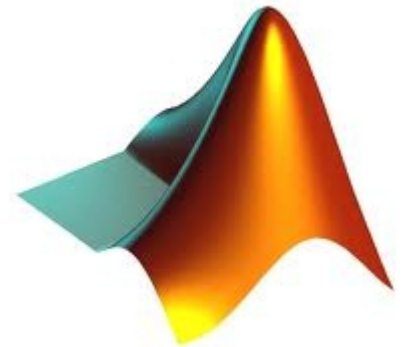


Matice v MATLAB-e

- Prístup (riadok, stĺpec) `>> A(2,1)`
`ans = 4`
- `:` celý riadok alebo stĺpec `>> A(:,2)`
`ans =`
2
5
8
- Interval `>> A(1:2,2)`
`ans =`
2
5



```
A =  
12 3  
4 5 6  
7 8 9
```



Operácie

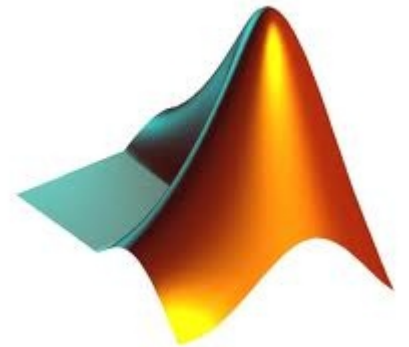
- maticové:

$+$, $-$, $*$, $/$, $^{\wedge}$

- Medzi prvkami:

$.*$, $./$, $.^{\wedge}$, $\text{sqrt}()$, $\text{sin}()$, $\text{cos}()$, ...

- $\text{size}(A)$ – rozmery
- $\text{sum}(A)$ – suma po stĺpcoch
- $\text{sum}(\text{sum}(A))$ – suma všetkých prvkov
- $\text{sum}(A(:))$



Operácie

- `>> A+A`

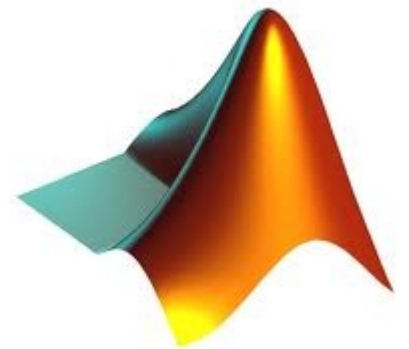
```
ans = 2 4 6  
      8 10 12  
      14 16 18
```

- `>> A*A`

```
ans = 30 36 42  
      66 81 96  
      102 126 150
```

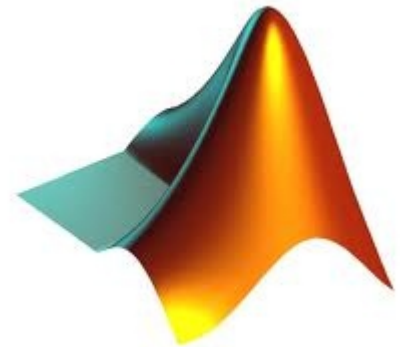
- `>> A.*A`

```
ans = 1 4 9  
      16 25 36  
      49 64 81
```



Názvy premenných

- Názvy premenných
- 63 signifikantnych znakov
- Začína písmenom
- Bez diakritiky a medzier
- Rozlišuje veľkosť písmen
- Odlišné od názvov príkazov a preddefinovaných premenných (pi, i, j, eps, ...)
- exist meno



Logické operátory

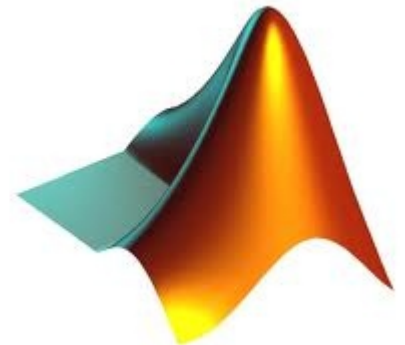
- Logické operátory

`==`, `<`, `>`, `~=`, `~`, ...

- `find('podmienka')`

– vráti indexy vyhovujúcich prvkov

Symbol	Represents	Symbol	Represents
<code>></code>	Greater than	<code>>=</code>	Greater or equal to
<code><</code>	Less than	<code><=</code>	Less or equal to
<code>~=</code>	Not equal to	<code>==</code>	Equal to
Not	<code>~</code>	And	<code>&</code>
Or	<code> </code> (single vertical line)		



```
>>A=[7 3 5; 6 2 1]
```

```
>>Idx=find(A<4)
```

```
A=
```

```
7 3 5
```

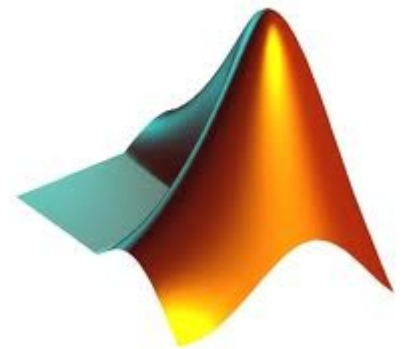
```
6 2 1
```

```
Idx=
```

```
3
```

```
4
```

```
6
```



```
>> [row col]=find(A==7)
```

```
row = 3
```

```
col = 1
```

```
>> [row col]=find(A>7)
```

```
row = 3
```

```
3
```

```
col = 2
```

```
3
```

```
>> Indx=find(A<5)
```

```
Indx = 1
```

```
2
```

```
4
```

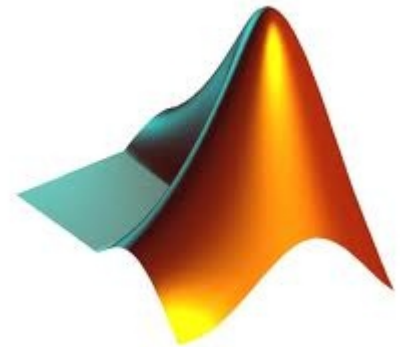
```
7
```

```
A =
```

```
12 3
```

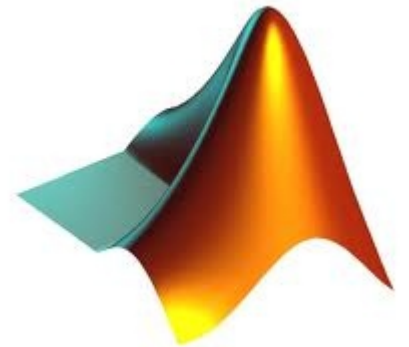
```
4 5 6
```

```
7 8 9
```



Graf $y=x^2$

- `x = linspace(0, 5, 100);`
- `y1 = x;`
- `plot(x, y1); grid;`
- `y2 = x.^2;`
- `y3 = x.^3;`
- `plot(x, y1, x, y2, x, y3); grid;`



- **flow control:**

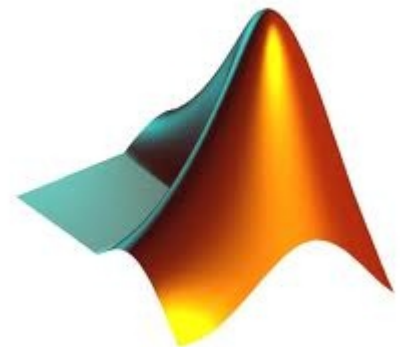
- `if`

- `switch`

- `for`

- `while`

- `break`



What if...

IF expression

statements

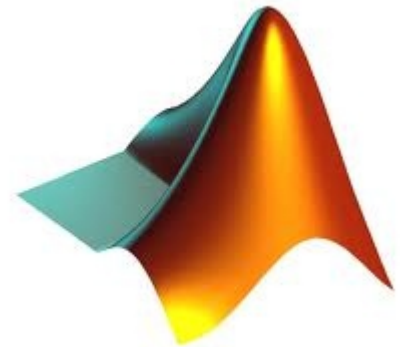
ELSEIF expression

statements

ELSE

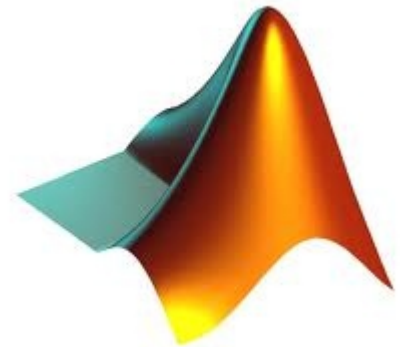
statements

END



for

```
FOR variable=expr  
  statements  
END
```



Timing

- tic; příkazy; toc;

- V sekundách

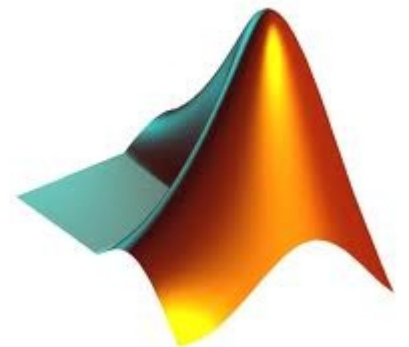
- V m-file

t0 = cputime

....příkazy, výpočty

t1 = cputime

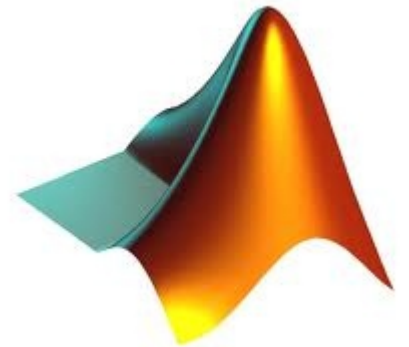
fprintf('vypočet trval %g', t1- t0)



Alokácia premenných

```
tic
x = 0;
for k = 2:1000000
    x(k) = x(k-1) + 5;
end
toc
```

```
tic
x = zeros(1, 1000000);
for k = 2:1000000
    x(k) = x(k-1) + 5;
end
toc
```



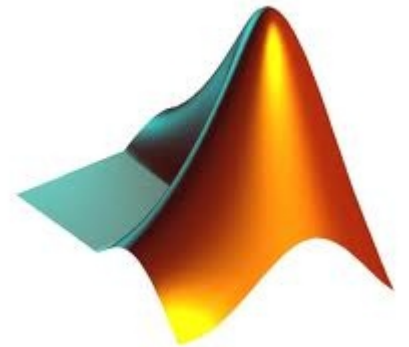
MATLAB vektorizácia kódu

- Chceme vytvoriť pole kde $v(p) = \frac{p}{\sin(p) + 2}$
- **1:**

```
for p = 1:1000  
v(p) = p / (sin(p) + 2); end
```
- **2:**

```
v = zeros (1, 1000);  
for p = 1:1000  
v(p) = p / (sin(p) + 2); end
```
- **3:**

```
p = 1:1000  
v = p ./ (sin(p) + 2)
```



MATLAB špecialitka

- Chceme vytvoriť pole kde $v(p) = \frac{p}{\sin(p) + 2}$
- **1:**

```
for p = 1:1000  
v(p) = (p/sin(p)+2); end
```

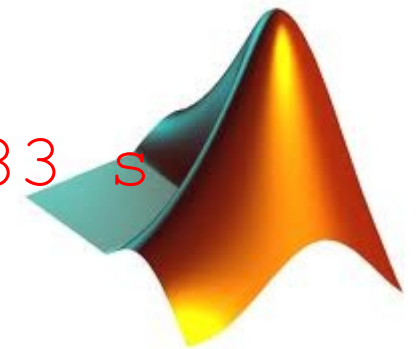
 1.82 s
- **2:**

```
v = zeros (1, 1000);  
for p = 1:1000  
v(p) = (p/sin(p)+2); end
```

 0.16 s
- **3:**

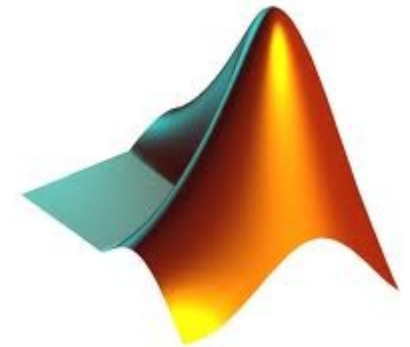
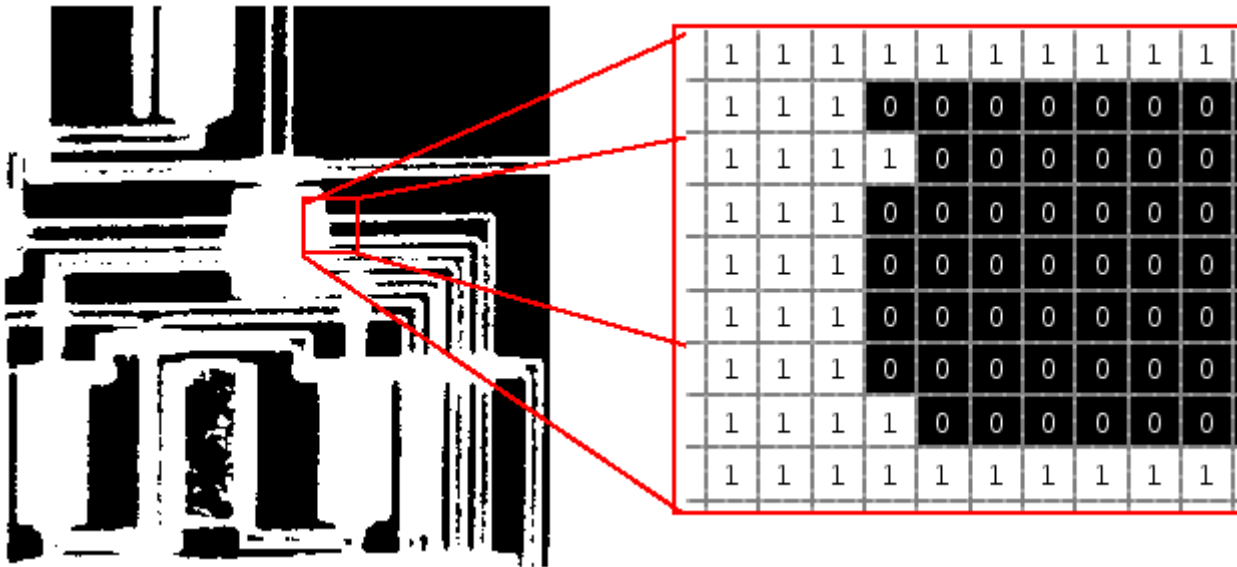
```
p = 1:1000  
v = (p./sin(p)+2)
```

 0.0083 s



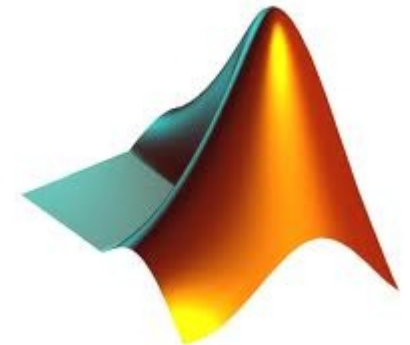
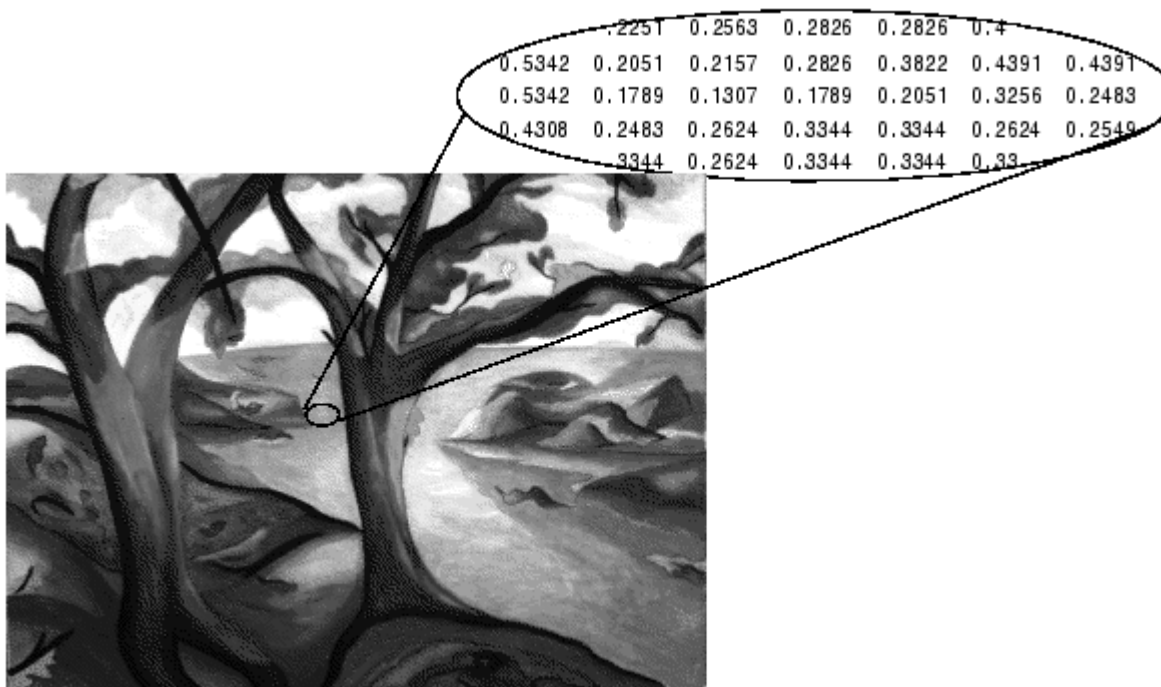
Obrázky

- binárne: {0,1}
- šedotónové: uint8, double ...
- RGB: $m \times n \times 3$



Obrázky

- binárne: {0,1}
- šedotónové: uint8, double ...
- RGB: $m \times n \times 3$



Obrázky

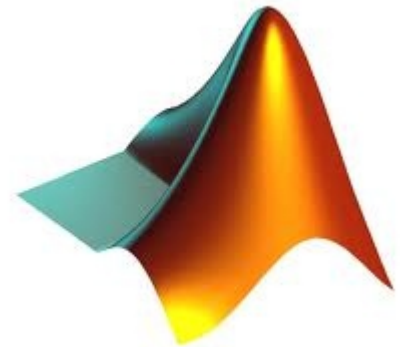
- binárne: {0,1}
- šedotónové: uint8, double
- **RGB: $m \times n \times 3$**

	0.2235	0.1294	Blue	0.4196		
	0.5804	0.2902	0.0627	0.2902	0.2902	0.4824
	0.5804	0.0627	0.0627	0.0627	0.2235	0.2588
	0.5176	0.1922	0.0627	Green	0.1922	0.2588
	0.5176	0.1294	0.1608	0.1294	0.1294	0.2588
	0.5176	0.1608	0.0627	0.1608	0.1922	0.2588
	0.5490	0.2235	0.5490	Red	0.7412	0.7765
	0.5490	0.3882	0.5176	0.5804	0.5804	0.7765
	0.5490	0.2588	0.2902	0.2588	0.2235	0.4824
	0.2235	0.1608	0.2588	0.2588	0.1608	0.2588
	0.2588	0.1608	0.2588	0.2588	0.2588	0.2588



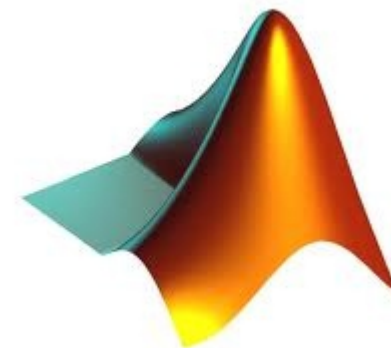
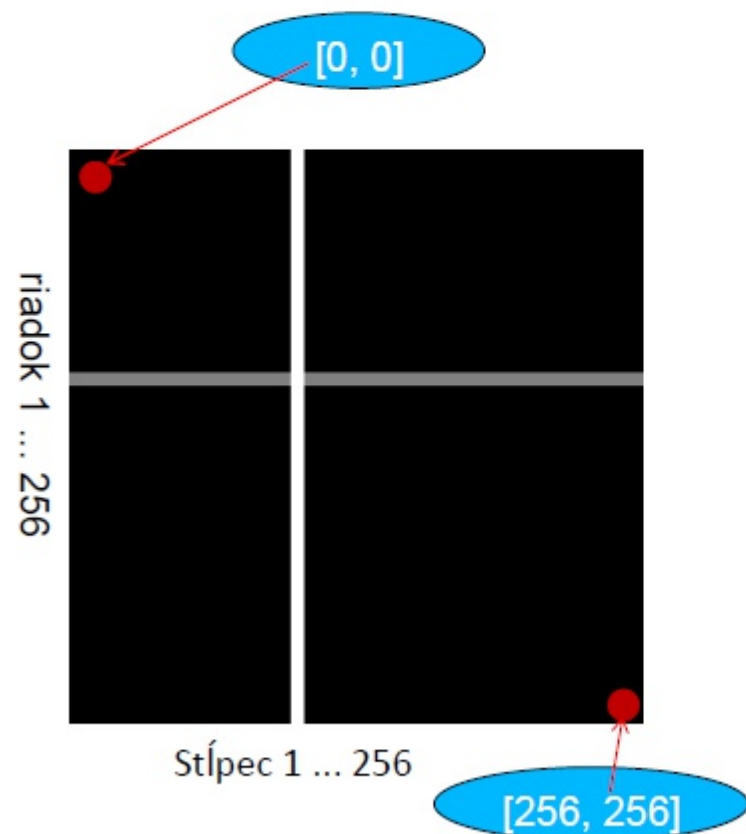
Import a Export

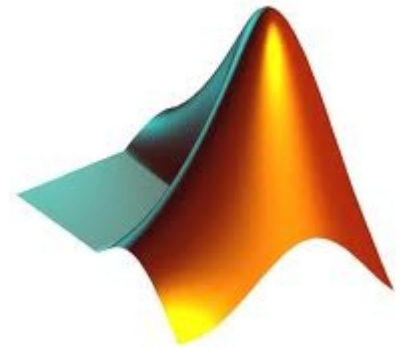
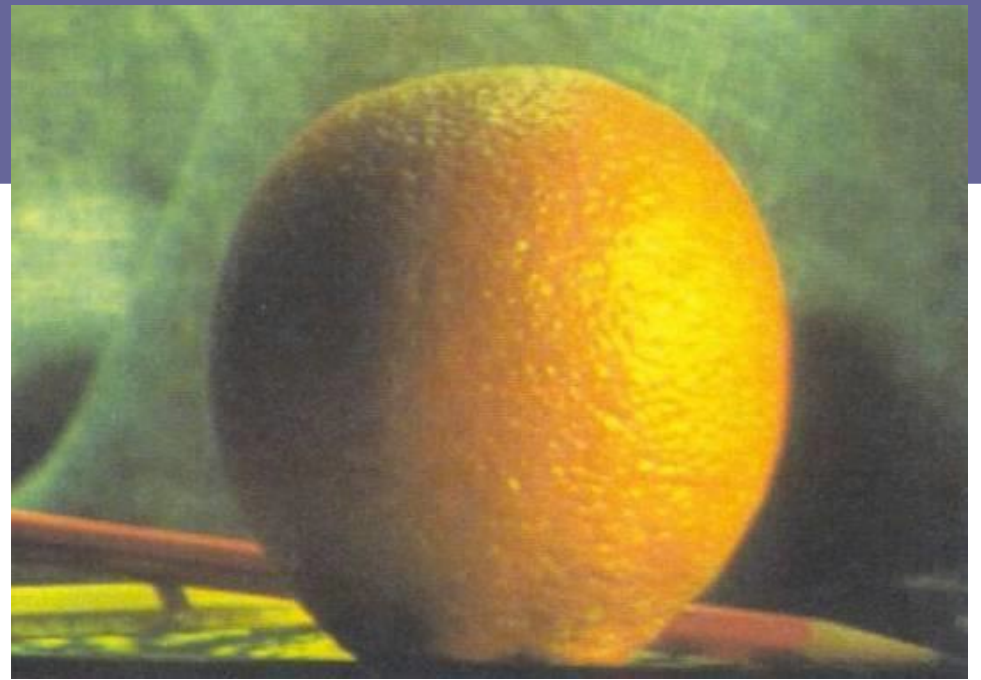
```
img = imread('apple.jpg');  
dim = size(img);  
figure;  
imshow(img);  
imwrite(img, 'output.bmp', 'bmp');
```

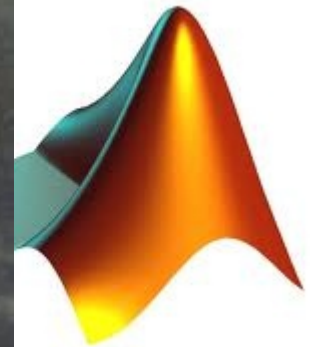
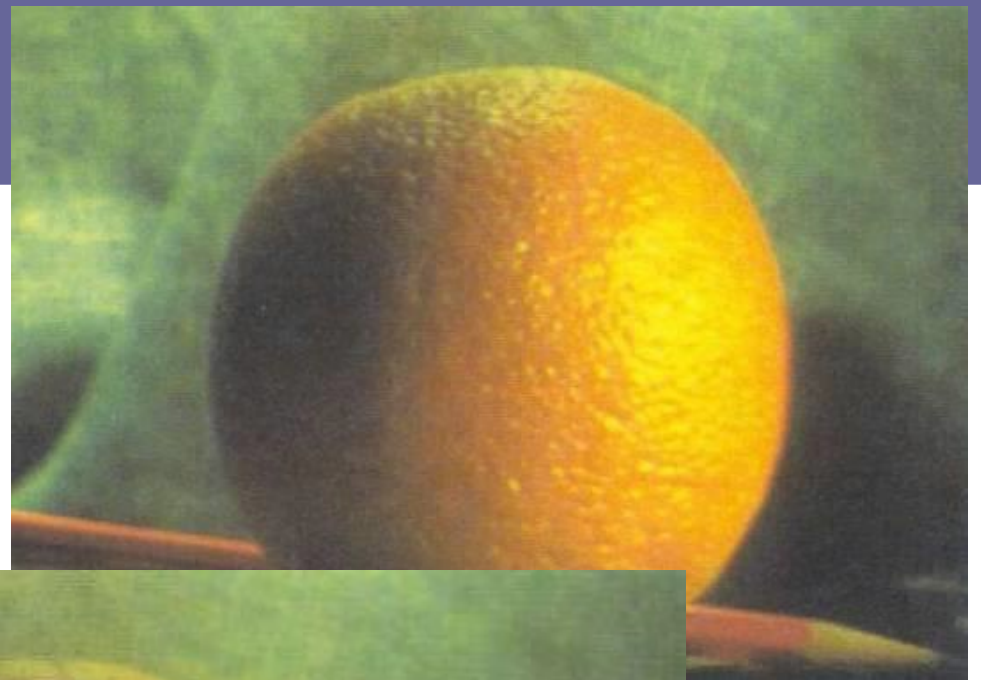


Šedotónový obraz

```
row = 256;  
col = 256;  
img = zeros(row, col);  
img(100:105, :) = 0.5;  
img(:, 100:105) = 1;  
figure;  
imshow(img);
```







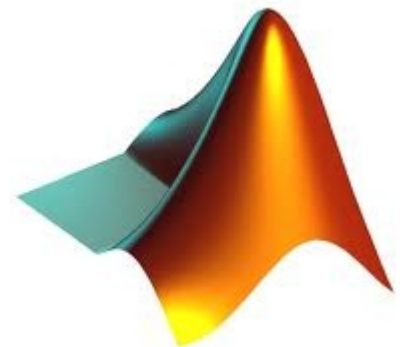
A a B veľkosti (540*380), zmiešanie

```
apple = imread('apple.jpg');  
orange = imread('orange.jpg');
```

Hrubá sila

```
% measure performance using stopwatch timer  
tic  
for i = 1 : size(apple, 1)  
    for j = 1 : size(apple, 2)  
        for k = 1 : size(apple, 3)  
            output(i, j, k) = (apple(i, j, k) + orange(i, j, k))/2;  
        end  
    end  
end  
toc
```

? sekúnd



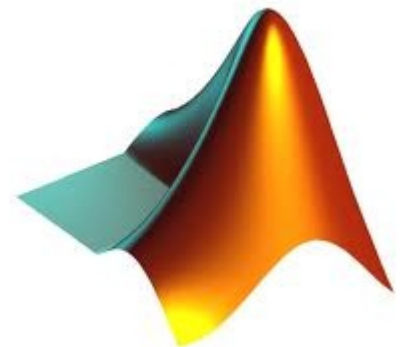
A a B veľkosti (540*380), zmiešanie

```
apple = imread('apple.jpg');  
orange = imread('orange.jpg');
```

Maticový prístup

```
tic  
    Output = (apple + orange) / 2;  
toc
```

? sekúnd



Optimalizácia výkonu

- Rýchle vektorové a maticové operácie
- Pomalé cykly
- Ako vektorizovať kód
 - <http://www.mathworks.com/support/tech-notes/1100/1109.html>

