

Základy počítačovej grafiky a spracovania obrazu

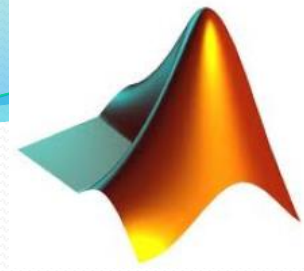
Základné operácie
pre prácu s obrazom

Doc. RNDr. Milan Ftáčnik, CSc.

Cvičenia – obrazárska časť

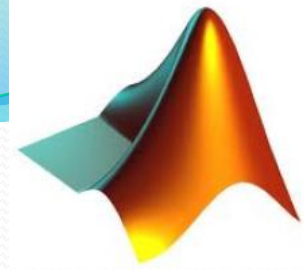
- Zvládnutie základov MATLAB-u
- Zvládnutie narábania s obrazmi ako aj tvorby jednoduchých algoritmov
- Projekty sú zamerané na vybrané problémy, ktoré má študent naprogramovať a vysvetliť dosiahnutý výsledok z hľadiska použitých metód spracovania obrazu

Úvod do MATLAB-u



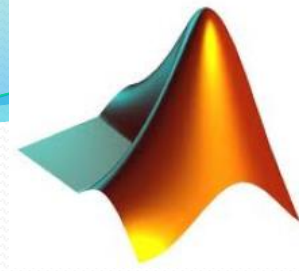
- Interaktívny nástroj na analýzu, návrh a riešenie problémov
- Matematické funkcie pre lineárnu algebru, štatistiku, Fourierovu analýzu, filtrovanie...
- Funkcie na vizualizáciu 2D a 3D dát
- Nástroje na vytvorenie GUI

Úvod do MATLAB-u



- **Využitie:** spracovanie signálu, spracovanie obrazu, testovanie a meranie, finančné modelovanie a analýza, výpočtová biológia...
- Toolboxy rozširujú prostredie MATLAB na riešenie tried problémov z konkrétnych oblastí

Úvod do MATLAB-u



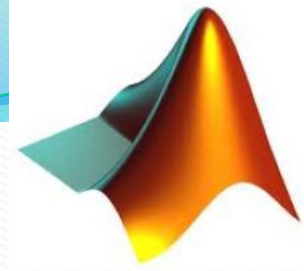
- Function list

<http://www.mathworks.com/help/matlab/functionlist.html>

- Tutorial

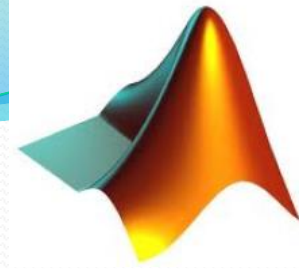
http://www.mathworks.com/help/pdf_doc/matlab/getstarted.pdf

Toolboxy



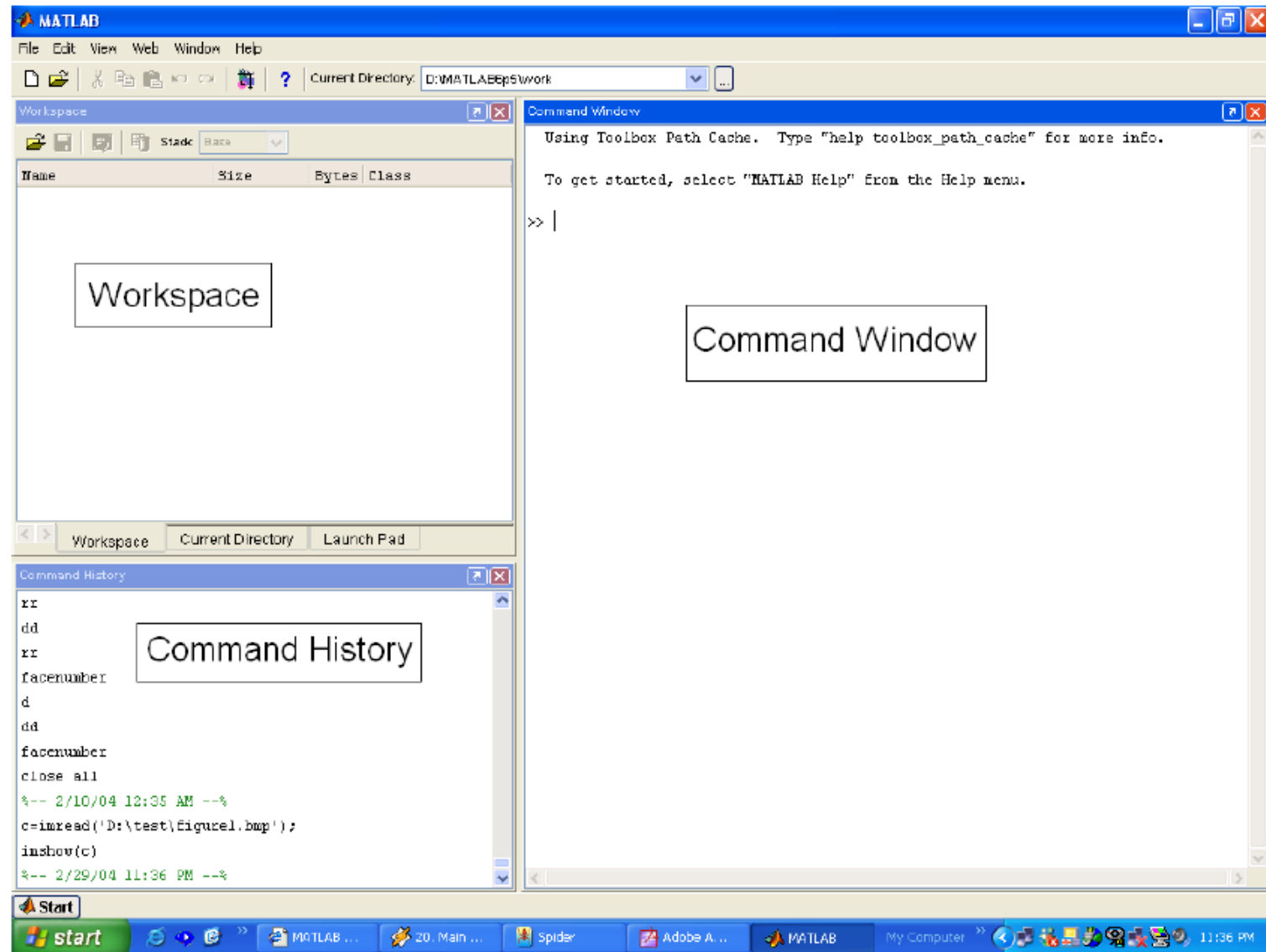
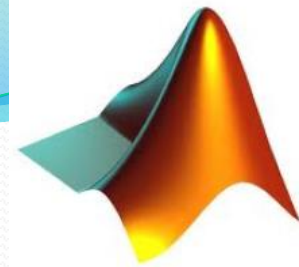
- Image Acquisition Toolbox
- Image Processing Toolbox
- Computer Vision System Toolbox
- Statistics and Machine Learning Toolbox
- Neural Networks Toolbox
- ...

Image Processing Toolbox

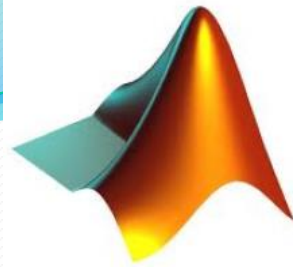


- **Image Analysis**
 - segmentácia, matematická morfológia, extrakcia príznakov, detekcia hrán
- **Spracovanie obrazu**
 - zvýšenie kontrastu, prevod medzi farebnými modelmi

Prostredie MATLAB-u

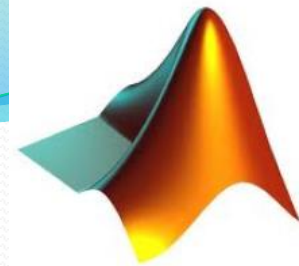


Prostredie MATLAB-u



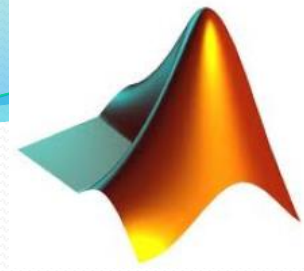
- Command window
 - písanie príkazov, výstupy, chyby
- Workspace
 - premenné, ich hodnoty a typy
- Command History
 - použité príkazy sa dajú „drag and drop“ do command window

Help v MATLAB-e



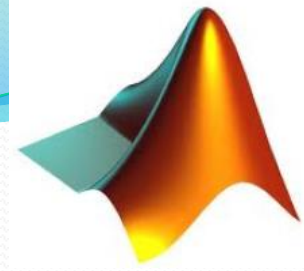
- help command
- lookfor keyword
- F1

Command window



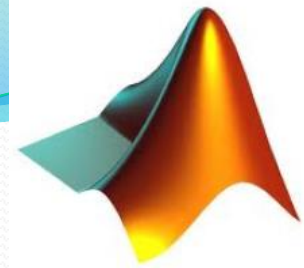
- $3 + 4 - 7$
- $k = 3 + 4 - 7$
- k
- $k;$
- $3^2 * 4$
- $2 + 2 / 1 + 1$

Command Window



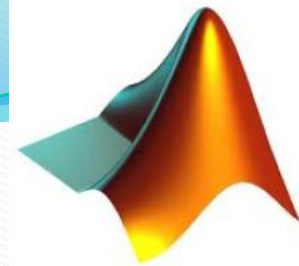
- 1/0 (Inf)
- 0/0 (NaN)
- **MATLAB je Case Sensitive!**
- K a k sú rôzne premenné
- 15 miest, ale ukazuje len 5
- format long / format short

Command Window



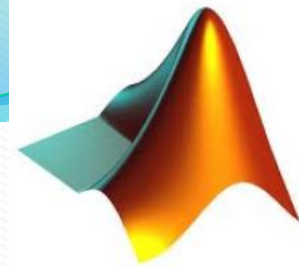
- MATLAB má množstvo vstavaných funkcií
- `sin`, `cos`, `tan`, `asin`, `acos`
- `sin(pi/2)`
- `log`, `log10`, `log2`
- `log10(100)`

Vektory v MATLAB-e



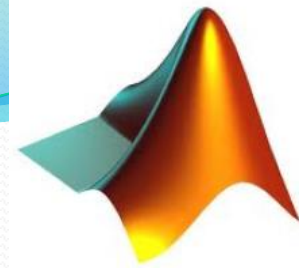
- $V = [1, 2, 3, 4]$
- $V = [1 \ 2 \ 3 \ 4]$
- $V = [1; 2; 3; 4]$
- $V = \text{start: step: end}$
- $V = 2:2:9$ $v = [2, 4, 6, 8]$
- $V = 2:5$ $v = [2, 3, 4, 5]$

Vektory v MATLAB-e



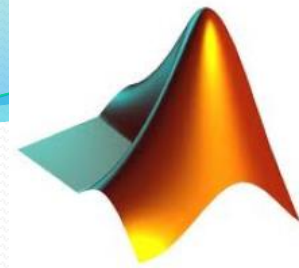
- $V = \text{linspace}(1,5,10)$
- $V(4) = 0$
- $V(5:7) = 0$
- $V(1:2:7) = 0$

Matice v MATLAB-e

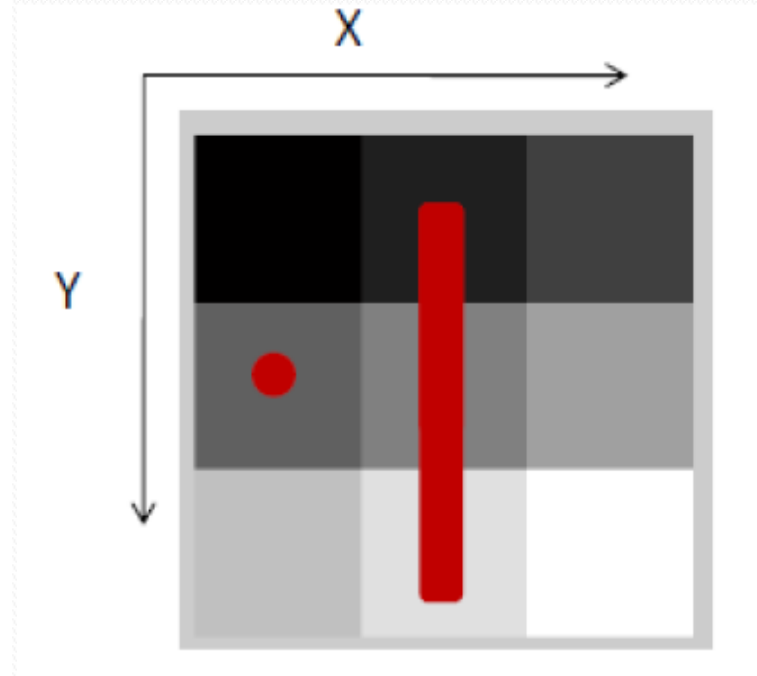


- **Vytvorenie:**
- $A = [1, 2, 3; 4, 5, 6; 7, 8, 9]$ rozmeru 3x3
- **Špeciálne matice:**
- $p = \text{zeros}(3,3) == \text{zeros}(3);$
- $o = \text{ones}(3,3) == \text{ones}(3);$
- $r = \text{rand}(3,3) == \text{rand}(3);$ rovnomerné rozdelenie
- $r1 = \text{randn}(1,10);$ normálne rozdelenie
- $k = \text{magic}(3);$

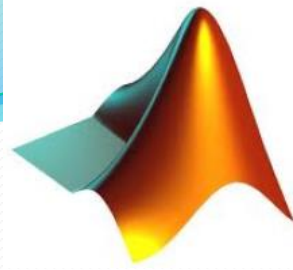
Matice v MATLAB-e



- 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

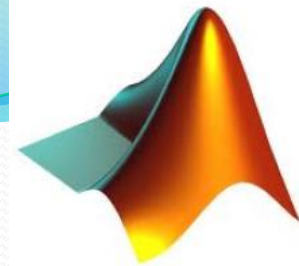


Operácie



- maticové:
 $+$, $-$, $*$, $/$, $^$
- Medzi prvkami:
 $.*$, $./$, $.^$, $\text{sqrt}()$, $\text{sin}()$, $\text{cos}()$, ...
- $\text{size}(A)$ – rozmery
- $\text{sum}(A)$ – suma po stĺpcoch
- $\text{sum}(\text{sum}(A))$ – suma všetkých prvkov
- $\text{sum}(A(:))$

Operácie II



- `>> A+A`

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

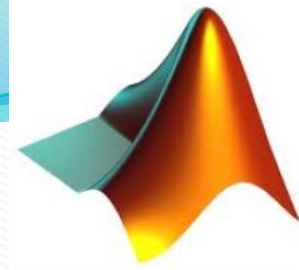
- `>> A*A`

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

- `>> A.*A`

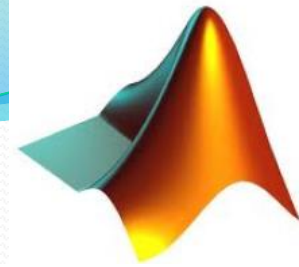
```
ans =     1  4  9  
        16 25 36  
        49 64 81
```

Indexovanie matice



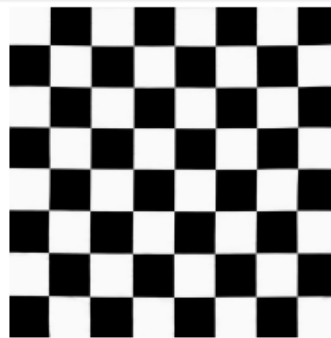
- Indexovanie jedným indexom:
 - Začíname vľavo hore, prejdeme dole po stĺpci a potom prejdeme na vrch ďalšieho stĺpca a pokračujeme `[r,c] = ind2sub([rows, cols], idx)`
- Indexovanie dvoma indexami:
 - Klasický spôsob, ktorý už bol prezentovaný (aj s využitím intervalov alebo celých riadkov či stĺpcov) `idx = sub2ind([rows, cols], r, c)`

Úloha na indexáciu 1



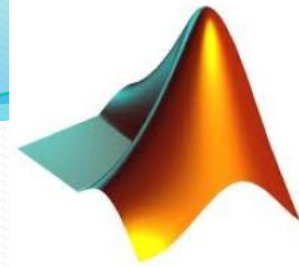
Zadanie

Vygenerujte maticu pomocou `rand(8)`. Premente všetky prvky ktoré by boli na šachovnici na čiernom políčku na 1. Následne premente všetky prvky menšie ako 0.3 na 0.



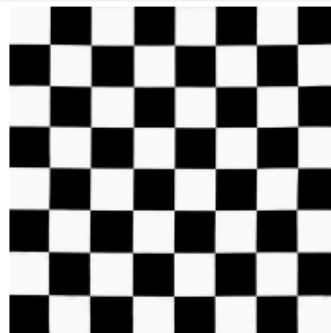
HINT: indexovať vieme aj pomocou podmienky, napr. `R(R==1)` vráti tie prvky z `R`, ktorých hodnota je 1.

Úloha na indexáciu 1



Zadanie

Vygenerujte maticu pomocou `rand(8)`. Premente všetky prvky ktoré by boli na šachovnici na čiernom políčku na 1. Následne premente všetky prvky menšie ako 0.3 na 0.



Riešenie napr:

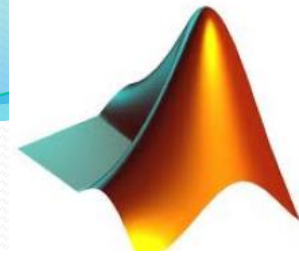
```
R = rand(8)
```

```
R(1:2:7, 2:2:8) = 1
```

```
R(2:2:8, 1:2:7) = 1
```

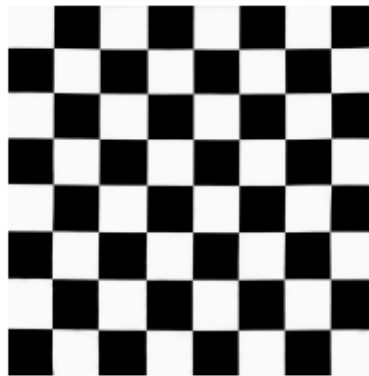
```
R(R < 0.3) = 0
```

Úloha na indexáciu 2

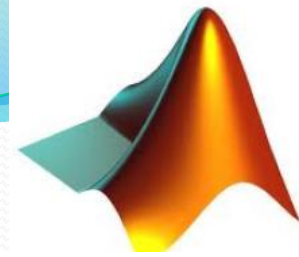


Zadanie

Vygenerujte maticu pomocou `magic(8)` a z nej vytvorte maticu 8x4 len z prvkov na bielych políčkach.

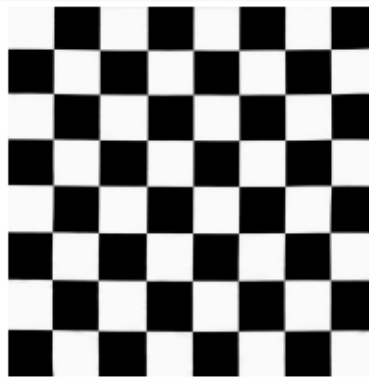


Úloha na indexáciu 2



Zadanie

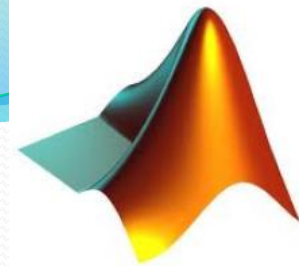
Vygenerujte maticu pomocou `magic(8)` a z nej vytvorte maticu 8x4 len z prvkov na bielych políčkach.



Riešenie napr:

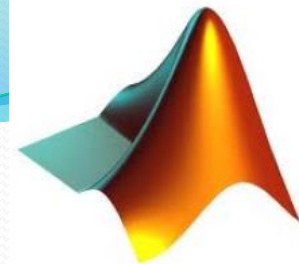
```
A = magic(8)
s = [1 0;0 1]
I = repmat(s,4)
B = reshape(A(I == 1),[8 4])
```


Názvy premenných



- Začína písmenom
- Bez diakritiky a medzier
- Rozlišuje veľkosť písmen
- Odlišné od názvov príkazov a preddefinovaných premenných
- exist meno

Logické operátory



- Logické operátory

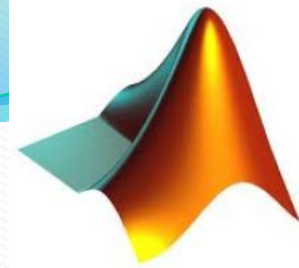
`==, <, >, ~=, ~, ...`

- `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)		

Riadiace príkazy



- flow control:

- if
- switch
- for
- while
- break

IF expression

statements

ELSEIF expression

statements

ELSE

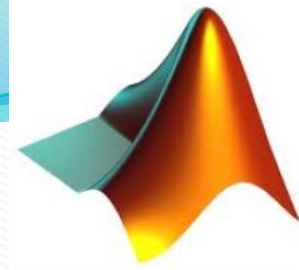
statements

END

FOR variable=expr
statements

END

Timing



- tic; prikazy; toc;

- V sekundách

- V m-file

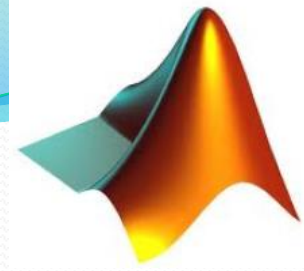
t0 = cputime

....príkazy, výpočty

t1 = cputime

fprintf('vypočet trval %g', t1- t0)

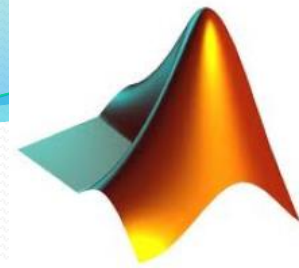
Alokácia premenných



```
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



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

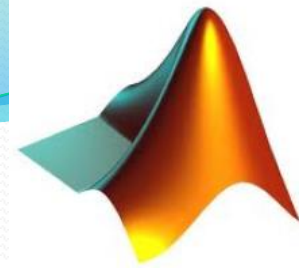
1.

```
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 vektorizácia



• 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.

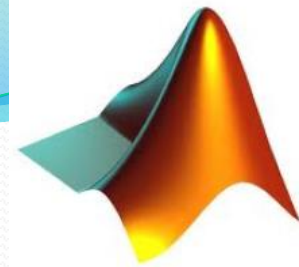
```
v = zeros(1,1000);  
for p = 1:1000  
    v(p) = p / (sin(p) + 2); end
```

 0,16 s
3.

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

 0,0083 s

Obrazy



- binárne: {0,1}

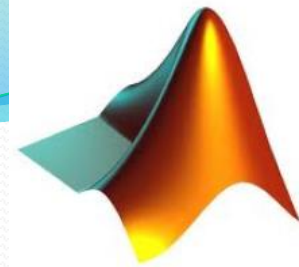
`im2uint8(I)` `im2double(I)`

- šedotónové: uint8, double ...
- RGB: $m \times n \times 3$

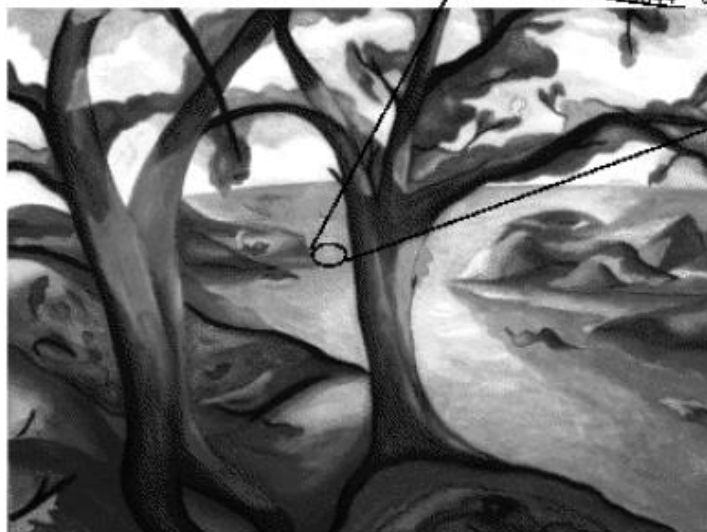


1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1

Obrazy II

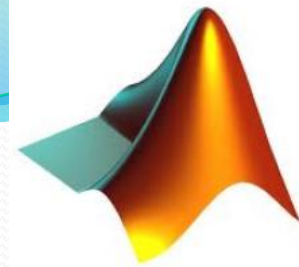


- binárne: {0,1}
- šedotónové: uint8, double ...
- RGB: $m \times n \times 3$



0.2251	0.2563	0.2826	0.2826	0.4		
0.5342	0.2051	0.2157	0.2826	0.3822	0.4391	0.4391
0.5342	0.1789	0.1307	0.1789	0.2051	0.3256	0.2483
0.4308	0.2483	0.2624	0.3344	0.3344	0.2624	0.2549
0.3344	0.2624	0.3344	0.3344	0.3344	0.3344	0.3344

Obrazy III

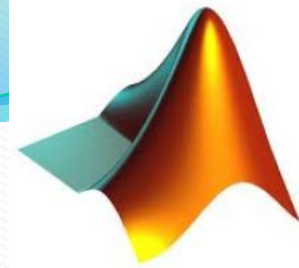


- binárne: $\{0,1\}$
- šedotónové: uint8, double
- **RGB: $m \times n \times 3$**

0.5804	0.2902	0.0627	Blue	0.4196	0.2902	0.2902	0.482
0.5804	0.0627	0.0627	0.0627	0.2235	0.2588	0.2588	0.2588
0.5176	0.1922	0.0627	Green	0.1922	0.2588	0.2588	0.2588
0.5176	0.1294	0.1608	0.1294	0.1294	0.2588	0.2588	0.2588
0.5176	0.1608	0.0627	0.1608	0.1922	0.2588	0.2588	0.2588
0.5490	0.2235	0.5490	Red	0.7412	0.7765	0.7765	902
0.5490	0.3882	0.5176	0.5804	0.5804	0.7765	0.7765	196
0.490	0.2588	0.2902	0.2588	0.2235	0.4824	0.2235	0.2235
0.2235	0.1608	0.2588	0.2588	0.1608	0.2588	0.2588	0.2588
0.2588	0.1608	0.2588	0.2588	0.2588	0.2588	0.2588	0.2588

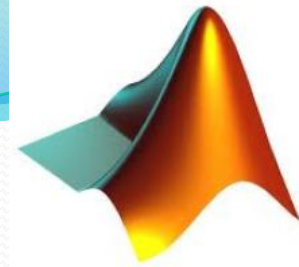


Import a export obrazu



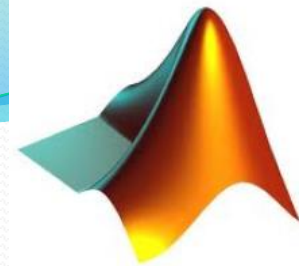
```
img = imread('apple.jpg');  
dim = size(img);  
figure;  
imshow(img);  
imwrite(img, 'output.bmp', 'bmp');
```

Import a export obrazu II

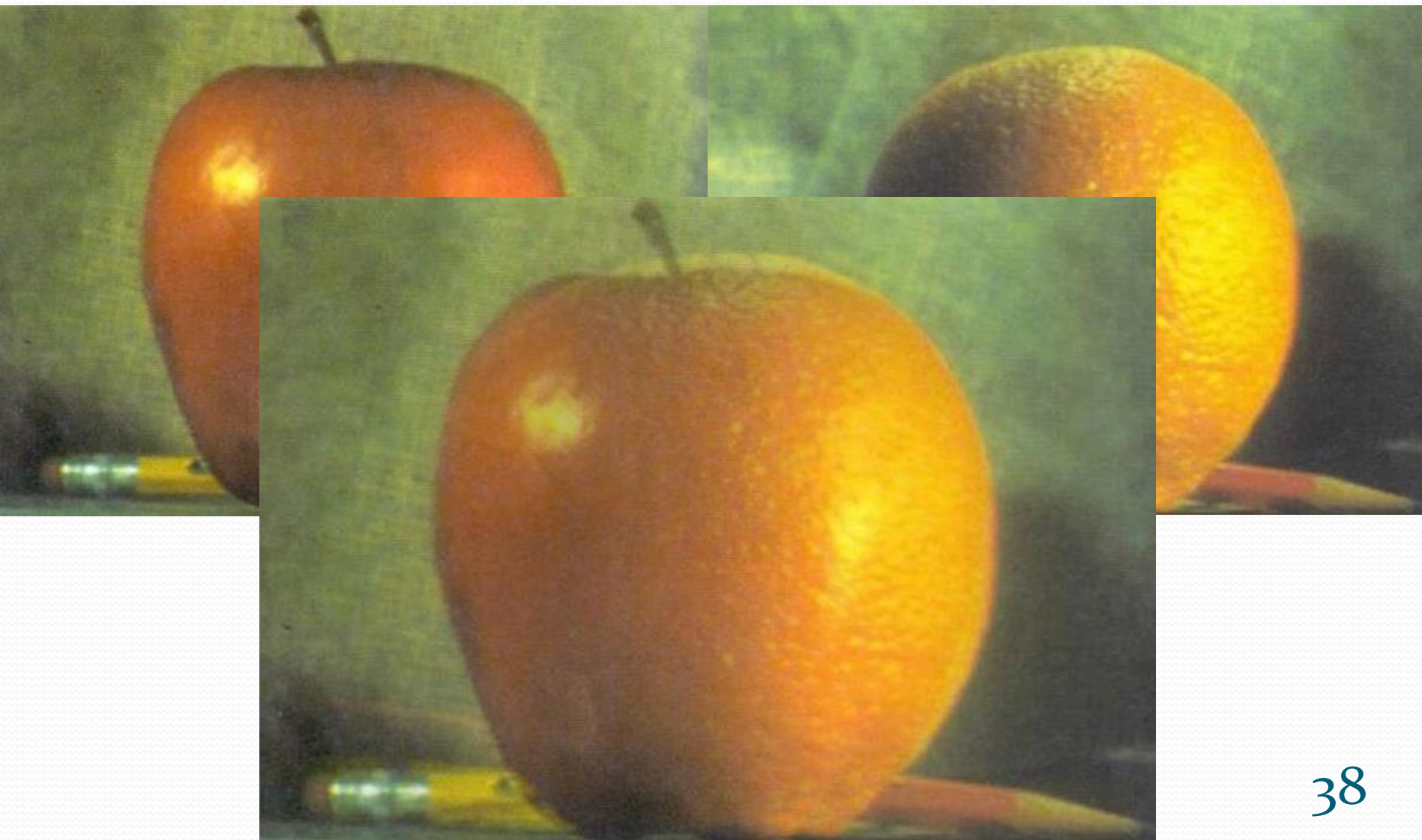
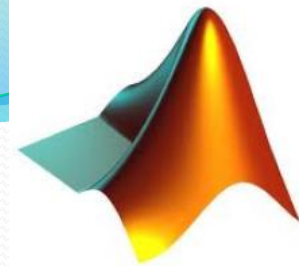


- Štandardné obrazy sú dostupné cez stránku
<http://www.sccg.sk/~ftacnik/standardne%20obrazy.docx>
- Všetky sú čiernobiele

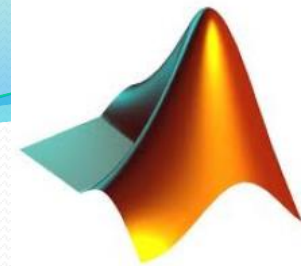
Sčítanie dvoch obrazov



Sčítanie dvoch obrazov II



Sčítanie dvoch obrazov III



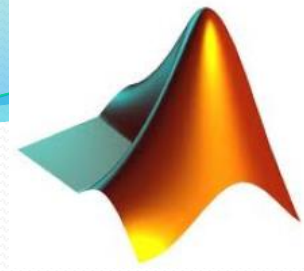
- **Hrubá sila**

- for i = 1:size(apple,1)
 for j = 1:size(apple, 2)
 for k = 1:size(apple, 3)

$$\text{output}(i,j,k) = (\text{apple}(i,j,k) + \text{orange}(i,j,k))/2$$

 end
 end
end

Sčítanie dvoch obrazov IV



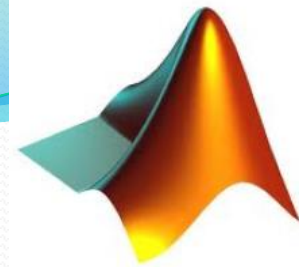
- **Maticový prístup**

tic

```
output = (apple + orange)/2;
```

toc

Funguje za podmienky, že rozmery oboch obrazov sú rovnako veľké.

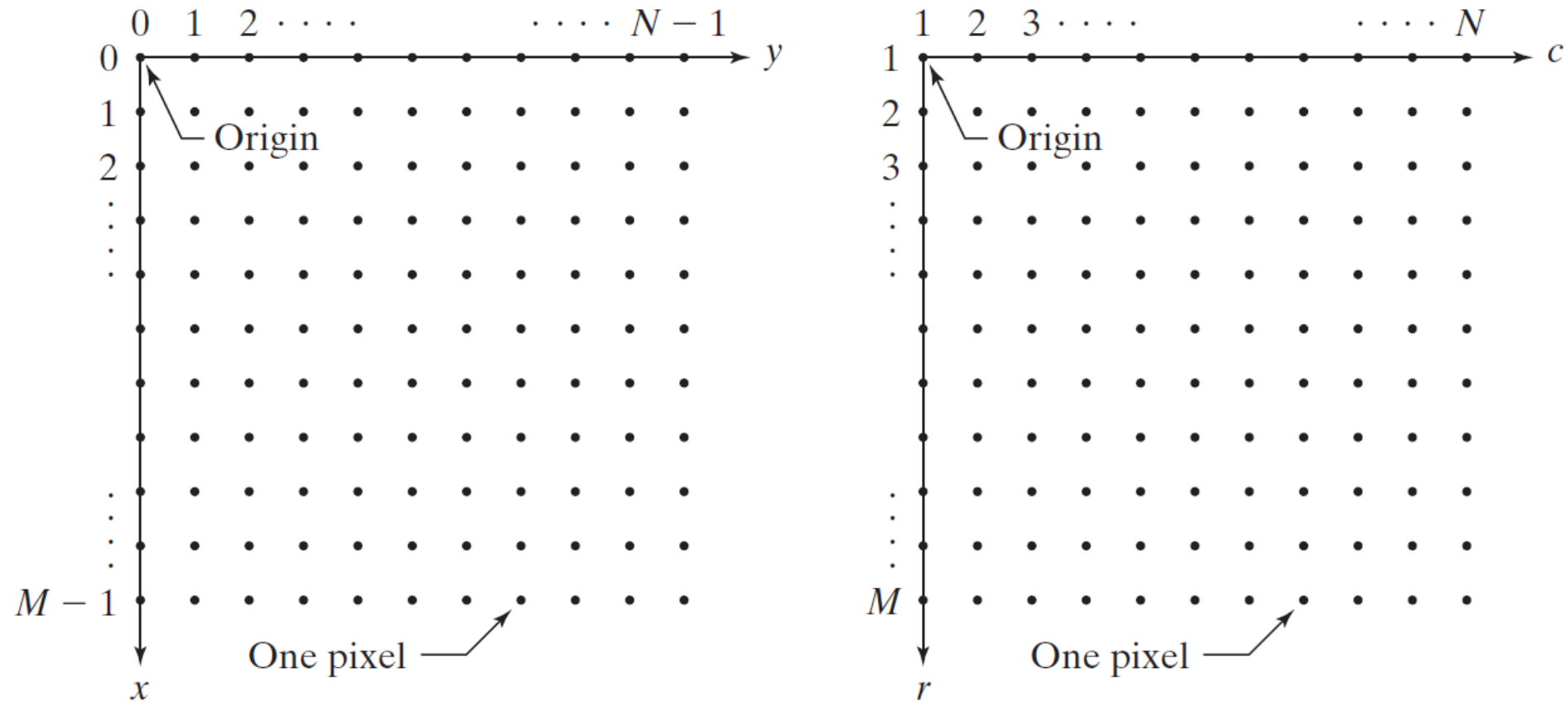


Optimalizácia výkonu

- Rýchle vektorové a maticové operácie
- Pomalé cykly
- Ako vektorizovať kód:

https://uk.mathworks.com/help/matlab/matlab_prog/vectorization.html

Súradnicový systém



Obraz ako matica

$$f = \begin{bmatrix} f(1, 1) & f(1, 2) & \dots & f(1, N) \\ f(2, 1) & f(2, 2) & \dots & f(2, N) \\ \vdots & \vdots & & \vdots \\ f(M, 1) & f(M, 2) & \dots & f(M, N) \end{bmatrix}$$

Triedy (typy) dát obrazu

Name	Description
double	Double-precision, floating-point numbers in the approximate range $\pm 10^{308}$ (8 bytes per element).
single	Single-precision floating-point numbers with values in the approximate range $\pm 10^{38}$ (4 bytes per element).
uint8	Unsigned 8-bit integers in the range [0, 255] (1 byte per element).
uint16	Unsigned 16-bit integers in the range [0, 65535] (2 bytes per element).
uint32	Unsigned 32-bit integers in the range [0, 4294967295] (4 bytes per element).
int8	Signed 8-bit integers in the range [-128, 127] (1 byte per element).
int16	Signed 16-bit integers in the range [-32768, 32767] (2 bytes per element).
int32	Signed 32-bit integers in the range [-2147483648, 2147483647] (4 bytes per element).
char	Characters (2 bytes per element).
logical	Values are 0 or 1 (1 byte per element).

Zmena triedy dát

```
f =
```

```
    -0.5    0.5  
    0.75    1.5
```

Performing the conversion

```
>> g = im2uint8(f)
```

yields the result

```
g =
```

```
    0    128  
   191   255
```

Zmena triedy dát

- Obrázok f je RGB obrázok
- Príkaz $g = \text{rgb2gray}(f)$ z neho urobí šedoúrovňový obrázok s kladnými hodnotami od 0 do 255
- Príkaz $k = \text{double}(g)$ konvertuje obrázok g na hodnoty s dvojitou presnosťou
- Príkaz $t = \text{uint8}(k)$ konvertuje obrázok k na obrázok s kladnými hodnotami od 0 do 255

M-file

- Ukladajú sa ako meno.m a editujú sa v editore MATLABu
- Súčasťou m-file je:
 - The function definition line
 - The H1 line
 - Help text
 - The function body
 - Comments

Riadok definície funkcie

- Píše sa nasledovne:

```
function [outputs] = name(inputs)
```

- Napr. funkcia súčtu a súčinu 2 obrazov

```
function [s, p] = sumprod(f, g)
```

- Volanie funkcie:

```
>> [s, p] = sumprod(f, g);
```


Riadok H1

- Je to textový riadok hneď po riadku definície funkcie, ktorý obsahuje jednoduchý komentár o činnosti funkcie, napr.

```
%SUMPROD Computes the sum and product of two images.
```

- Riadok H1 sa objaví po zadaní príkazu

```
>> help function_name
```

Ostatné časti M-filu

- **Help text** nasleduje za riadkom H1 do prvého vykonateľného príkazu, pričom každý riadok začína %
- **Telo funkcie** obsahuje MATLAB-ovský kód
- **Komentárové riadky** – všetky, ktoré sa začínajú % a nie sú súčasťou H1 a Help

Aritmetické operátory

Operator	Name	Comments and Examples
+	Array and matrix addition	$a + b$, $A + B$, or $a + A$.
-	Array and matrix subtraction	$a - b$, $A - B$, $A - a$, or $a - A$.
.*	Array multiplication	$Cv = A.*B$, $C(I, J) = A(I, J) * B(I, J)$.
*	Matrix multiplication	$A*B$, standard matrix multiplication, or $a*A$, multiplication of a scalar times all elements of A .
./	Array right division [†]	$C = A ./ B$, $C(I, J) = A(I, J) / B(I, J)$.
.\	Array left division [†]	$C = A .\ B$, $C(I, J) = B(I, J) / A(I, J)$.
/	Matrix right division	A/B is the preferred way to compute $A * \text{inv}(B)$.
\	Matrix left division	$A \backslash B$ is the preferred way to compute $\text{inv}(A) * B$.
.^	Array power	If $C = A.^B$, then $C(I, J) = A(I, J)^{B(I, J)}$.
^	Matrix power	See <code>help</code> for a discussion of this operator.
.'	Vector and matrix transpose	$A.'$, standard vector and matrix transpose.
'	Vector and matrix complex conjugate transpose	A' , standard vector and matrix conjugate transpose. When A is real $A.' = A'$.
+	Unary plus	$+A$ is the same as $0 + A$.
-	Unary minus	$-A$ is the same as $0 - A$ or $-1 * A$.
:	Colon	Discussed in Section 2.8.1.