

Introduction to MATLAB

Arturo Donate

Introduction

- What is MATLAB?
- Environment
- MATLAB Basics
- Programming
- Toolboxes
- Comparison
- Conclusion

What is MATLAB?

- Matrix laboratory
 - programming environment
 - high-performance language
 - Windows, OSX, Linux/UNIX
- Matrix/Vector computations
 - linear equations, eigenvectors, etc...
 - LINPACK, EISPACK, LAPACK, BLAS, etc...

What is MATLAB?

- Uses:

- math & computation
- algorithm development
- modeling and simulation
- data analysis and visualization
- application development

What is MATLAB?

- Toolboxes

- image processing

- filtering, transforms, analysis, enhancement

- statistics

- linear models, probability dist., HMMs

- optimization

- max/minimization, least squares, line fitting

What is MATLAB?

- Toolboxes

- fixed-point

- fixed-point data type & arithmetic

- others:

- symbolic math

- signal processing

- virtual reality

Environment

The image displays the MATLAB R2007a environment. The main window is titled "MATLAB" and shows the "Current Directory" as "/Users/arturodonate/Documents/stereo". The workspace contains several variables:

Name	Value	Min	Max
ans	[1.5 0.45 0.45 1 1.5]	0.45	1.5
dd	<111x131 double>	0	1
disp	<111x131 double>	0	10.5
iml	<480x640 uint8>	19	255
imr	<480x640 uint8>	17	255
left	<15589x2 double>	1	131
pts	<15589x3 double>	0	Inf
region	<139x151 uint8>	40	135
right	<15589x2 double>	0	125...

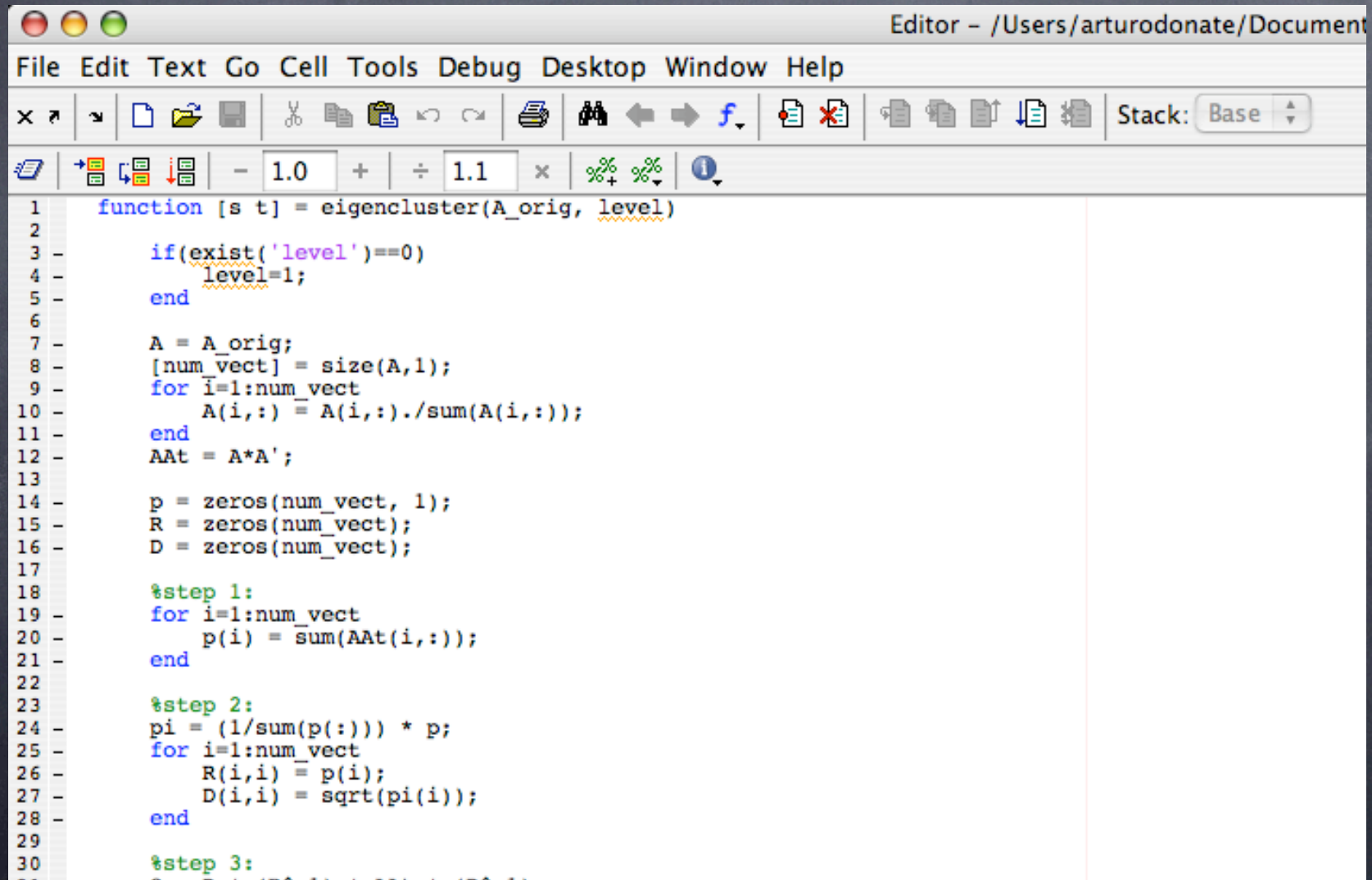
The Command Window shows the following output:

```
0.8048 0.8048 0.8048 0.8048 0.8000
0.8095 0.8048 0.8048 0.8048 0.8048
0.8143 0.8095 0.8095 0.8095 0.8048
0.8143 0.8143 0.8143 0.8143 0.8143
0.8190 0.8190 0.8190 0.8238 0.8190
0.8286 0.8238 0.8238 0.8238 0.8238
0.8333 0.8286 0.8286 0.8286 0.8286
0.8333 0.8333 0.8333 0.8333 0.8333
0.8381 0.8333 0.8333 0.8333 0.8333
0.8381 0.8381 0.8381 0.8381 0.8381
0.8429 0.8429 0.8429 0.8429 0.8429
0.8667 0.8476 0.8667 0.8667 0.8667
0.8667 0.8667 0.8714 0.8714 0.8714
0.8714 0.8714 0.8762 0.8762 0.8762
0.8762 0.8762 0.8810 0.8810 0.8810
0.8810 0.8810 0.8857 0.8857 0.8810
0.8857 0.8857 0.8857 0.8857 0.8857
0.8905 0.8905 0.8905 0.8905 0.8905
0.8952 0.8905 0.8905 0.8952 0.8952
0.8952 0.8952 0.8952 0.8952 0.8952
0.9000 0.9000 0.9000 0.9000 0.9000
0.9000 0.9000 0.9000 0.9000 0.9000
0.9048 0.9048 0.9048 0.9048 0.9048
0.9095 0.9095 0.9095 0.9095 0.9095
0.9143 0.9143 0.9143 0.9143 0.9143
0.9190 0.9190 0.9190 0.9190 0.9190
0.9238 0.9190 0.9238 0.9238 0.9238
0.9286 0.9238 0.9286 0.9286 0.9286
0.9286 0.9286 0.9333 0.9333 0.9333
0.9333 0.9333 0.9333 0.9333 0.9381
0.9381 0.9381 0.9429 0.9429 0.9429
0.9429 0.9429 0.9429 0.9429 0.9429
0.9429 0.9429 0.9476 0.9476 0.9476
0.9476 0.9476 0.9571 0.9619 0.9619
0.9619 0.9619 0.9619 0.9619 0.9619
0.9619 0.9619 0.9667 0.9667 0.9667
0.9667 0.9667 0.9667 0.9667 0.9714
0.9667 0.9667 0.9714 0.9714 0.9714
0.9714 0.9714 0.9714 0.9714 0.9762
0.9714 0.9762 0.9762 0.9762 0.9762
0.9762 0.9810 0.9810 0.9810 0.9810
0.9810 0.9857 0.9857 0.9810 0.9857
0 0 0 0 0
```

The Command History shows the following commands:

```
min(left(:,2))
max(left(:,2))
max(left(:,1))
[disp pts] = stereo(left, right);
imagesc(disp), colormap gray
[disp pts] = stereo(left, right);
imagesc(disp), colormap gray
max(disp(:))
min(disp(:))
dd = disp;
dd = disp ./ max(disp(:))
dd = disp ./ max(disp(:));
imagesc(dd), colormap gray
figure, imagesc(iml), colormap gray
figure, imagesc(region), colormap gray
colormap jet
figure, imagesc(iml)
c
??? Undefined function or variable 'c'.
figure, imagesc(region), colormap gray
figure, imagesc(iml), colormap gray
figure, imagesc(iml), colormap gray
```

Environment



The image shows a screenshot of a MATLAB editor window. The title bar reads "Editor - /Users/arturodonate/Document". The menu bar includes "File", "Edit", "Text", "Go", "Cell", "Tools", "Debug", "Desktop", "Window", and "Help". The toolbar contains various icons for file operations, editing, and navigation. Below the toolbar, there are zoom controls showing a zoom level of 1.0 and a scroll bar. The main area displays the following MATLAB code:

```
1 function [s t] = eigencluster(A_orig, level)
2
3     if(exist('level')==0)
4         level=1;
5     end
6
7     A = A_orig;
8     [num_vect] = size(A,1);
9     for i=1:num_vect
10        A(i,:) = A(i,)./sum(A(i,:));
11    end
12    AAt = A*A';
13
14    p = zeros(num_vect, 1);
15    R = zeros(num_vect);
16    D = zeros(num_vect);
17
18    %step 1:
19    for i=1:num_vect
20        p(i) = sum(AAt(i,:));
21    end
22
23    %step 2:
24    pi = (1/sum(p(:))) * p;
25    for i=1:num_vect
26        R(i,i) = p(i);
27        D(i,i) = sqrt(pi(i));
28    end
29
30    %step 3:
31    Q = R * (D^-1) * (sum(AAt,1) + 22) * (D^-1);
```


Basic Console Commands

- help

- ls

- who, whos

- clear, pack

- load, save

- eval

- disp

- what, type

- lookfor, which

- exit, quit

Basic Math Commands

Operations

- mean, median, mode
- sum, abs
- sin, sinh, asin, etc...
- sqrt, log, exp
- floor, ceil, round
- hist, plot

Constants

- $\pi = 3.14159\dots$
- $i, j = \sqrt{-1}$
- realmin, realmax
- Inf, NaN

Vectors and Matrices

• Creating

- $M = \text{ones}(10,10)$
- zeros , rand , randn ,
 eye , magic)
- $M = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$
- $M(1,1) = 23$
- $M = [\text{eye}(6)\ \text{rand}(6)]$

• Accessing

- $M(x, y)$
- $M(x, :)$, $M(x, i:j)$

• Deleting

- $M = []$

Vectors and Matrices

- Operations

- + - * / ^ \

- any, all

- diag

- rank

- eig

- svd

- trace

- prod

Vectors and Matrices

- Sample expressions:

- $A = B * C$

- $num = abs(3 + 4i)$

- $vector = sin(1:50)$

- $B = [A' C']$

- $B(:,2) = []$

- $determ = det(C)$

- $index = find(C > 0)$

- $x = \sim isprime(y)$

- $num = numel(x)$

- $[x y] = size(C)$

Vectors and Matrices

- Colon notation

- $1 : 50$

- $1 : 2 : 50$

- $A(:, 2)$

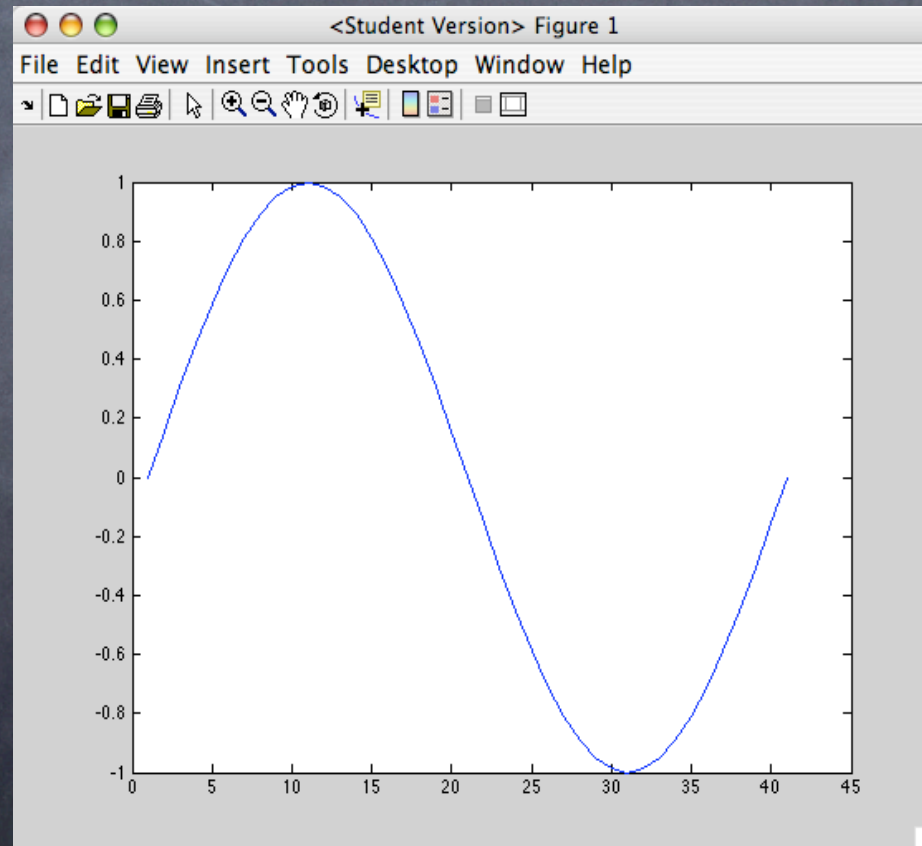
- $A(1:3, 2)$

- $A(2:4, :)$

- $\sin(0 : \pi/20 : 2*\pi)$

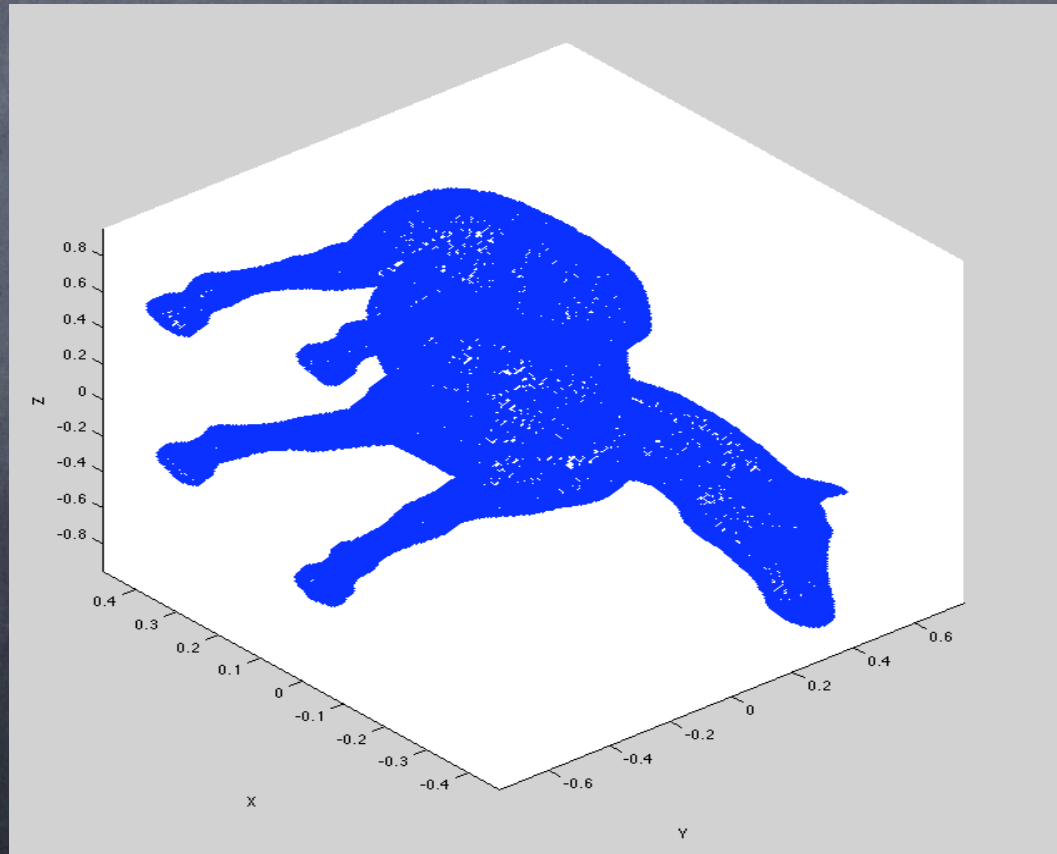
Graphics

- 2D Example: `plot(sin(0 : pi/20 : 2*pi))`

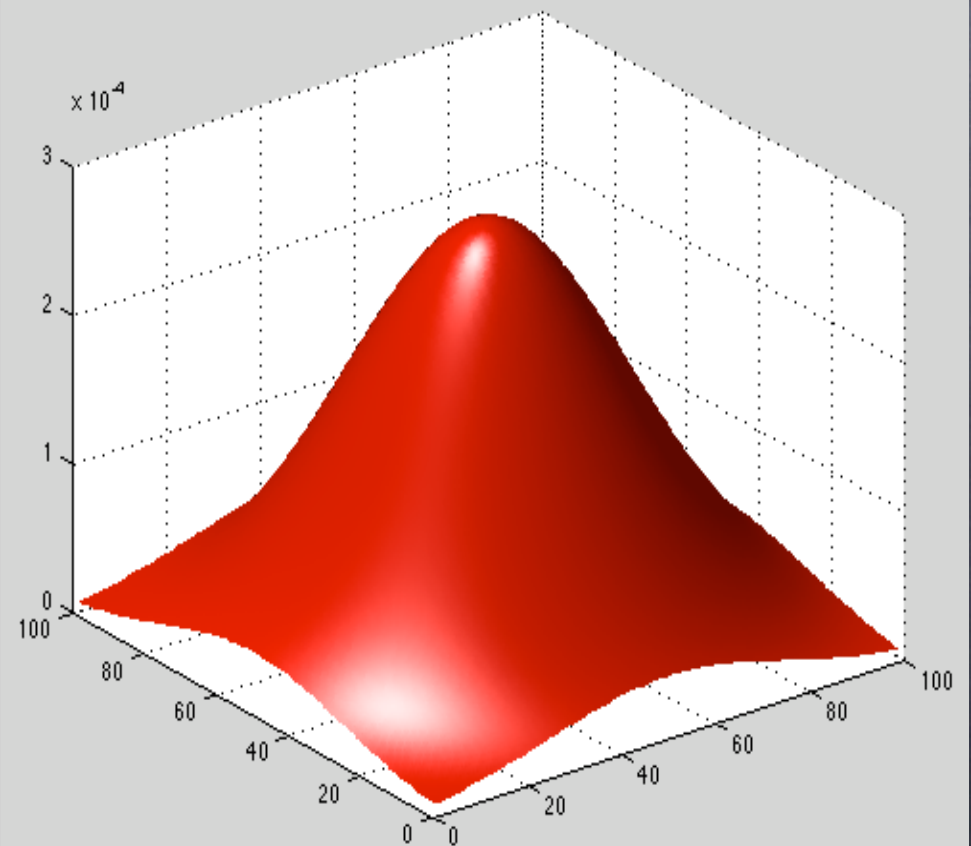
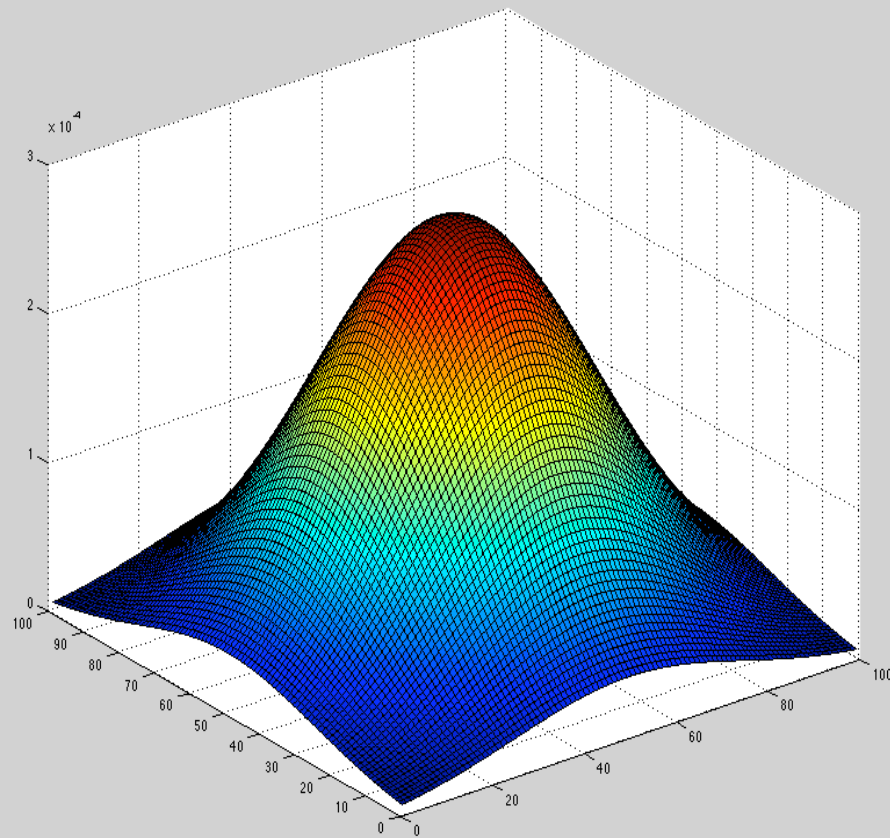


Graphics

- 3D example - plot 3D point cloud



Graphics



Programming

• Functions

- function [X Y] =
func_name(arg1,
arg2)

- func_name.m

- primary vs sub-
functions

- nested function
declaration/definition

- private functions

- function overloading
(int vs double)

- global variables

- pre-allocation

- vectorization

Programming

- script vs function
- if, elseif, else
- for, while
- switch, case
- continue, break
- return

Programming

- Relations

- $\sim a$

- $a | b$

- $a == b$

- $a | \sim a$

- $a \sim = b$

- $a \leq b$

- $a \& b$

Programming

- Data Structures

- Matrix

- basic data type

- Cell

- multi-dimensional matrices

- String

- character array

- Structure

- type with various fields

Simple Example: SVD

- Create a function that:
 - takes a matrix as input
 - make sure the matrix is square
 - computes the SVD decomposition
 - returns the first singular value
 - `answer = my_func(matrix)`

Simple Example: Sort

- Create a function that:
 - takes a list as input
 - sorts the list
 - returns a list with sorted values
 - `answer = my_func(matrix)`

Image Processing

- Toolbox used for image processing, computer vision, and signal processing tasks
 - image transformation
 - registration
 - filtering
 - image analysis/enhancement/de-blurring
 - segmentation
 - etc...

Image Processing

- Noise reduction example
 - salt & pepper noise
 - mean, median filter
 - `filter2(fspecial('average', 3), image) / 255`
 - `medfilt2(image, [3 3])`

Image Processing

- Color segmentation example
 - `cform = makecform('srgb2lab')`
 - `lab_img = applycform(image, cform)`
 - `ab = double(lab_img(:,:,2:3))`
 - `nrows = size(ab, 1)`
 - `ncols = size(ab, 2)`
 - `ab = reshape(ab, nrows*ncols, 2)`

Image Processing

- `[cluster_idx, cluster_center] = kmeans(ab, 3, 'distance', 'sqEuclidean', 'Replicates', 3);`
- `label = reshape(cluster_idx, nrows, ncols, 1);`

Image Processing

- Texture segmentation example
 - `E = entropyfilt(I);`
 - `Eim = mat2gray(E);`
 - `BW1 = im2bw(Eim, 0.8);`
 - `BWao = bwareaopen(BW1, 2000);`
 - `nhood = true(9);`

Image Processing

- `closeBWao = imclose(BWao, nhood);`
- `roughmask = imfill(closeBWao, 'holes');`
- `I2 = I; I3 = I;`
- `I2(roughmask) = 0;`
- `I3(~roughmask) = 0;`

MATLAB Clones

- Packages: Scilab, Octave, Rlab
- Similar:
 - matrix is basic data type
 - complex number support
 - built-in mathematical functions
 - powerful library
 - user-defined functions

MATLAB Clones

- Scilab

- www.scilab.org
- best support & docs
- good compatibility

- Octave

- [www.gnu.org/
software/octave](http://www.gnu.org/software/octave)

- Unix-like

- most compatible

- Rlab

- rlab.sourceforge.net
- attempts to improve syntax/semantics
- least compatible

Conclusions

- interactive programming environment
- high performance language
 - algorithm design
 - modeling & simulation
 - analysis & visualization
- linear algebra
- toolboxes

References

- MATLAB:
www.mathworks.com
- MATLAB Tutorial:
www.math.ufl.edu/help/matlab-tutorial/
- MATLAB Comparison:
www.dspguru.com/sw/opendsp/mathclo2.htm