UNIVERSITÀ DEGLI STUDI
DI MILANO

Introduction to Matlab

Enrique Muñoz Ballester

Dipartimento di Informatica
via Bramante 65, 26013 Crema (CR), Italy
enrique.munoz@unimi.it

Contact

- Email: enrique.munoz@unimi.it
- Office: Room BT-43
  Industrial, Environmental, and Biometrics Informatics Laboratory
- Website: https://homes.di.unimi.it/munoz/teaching.html

Why?

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

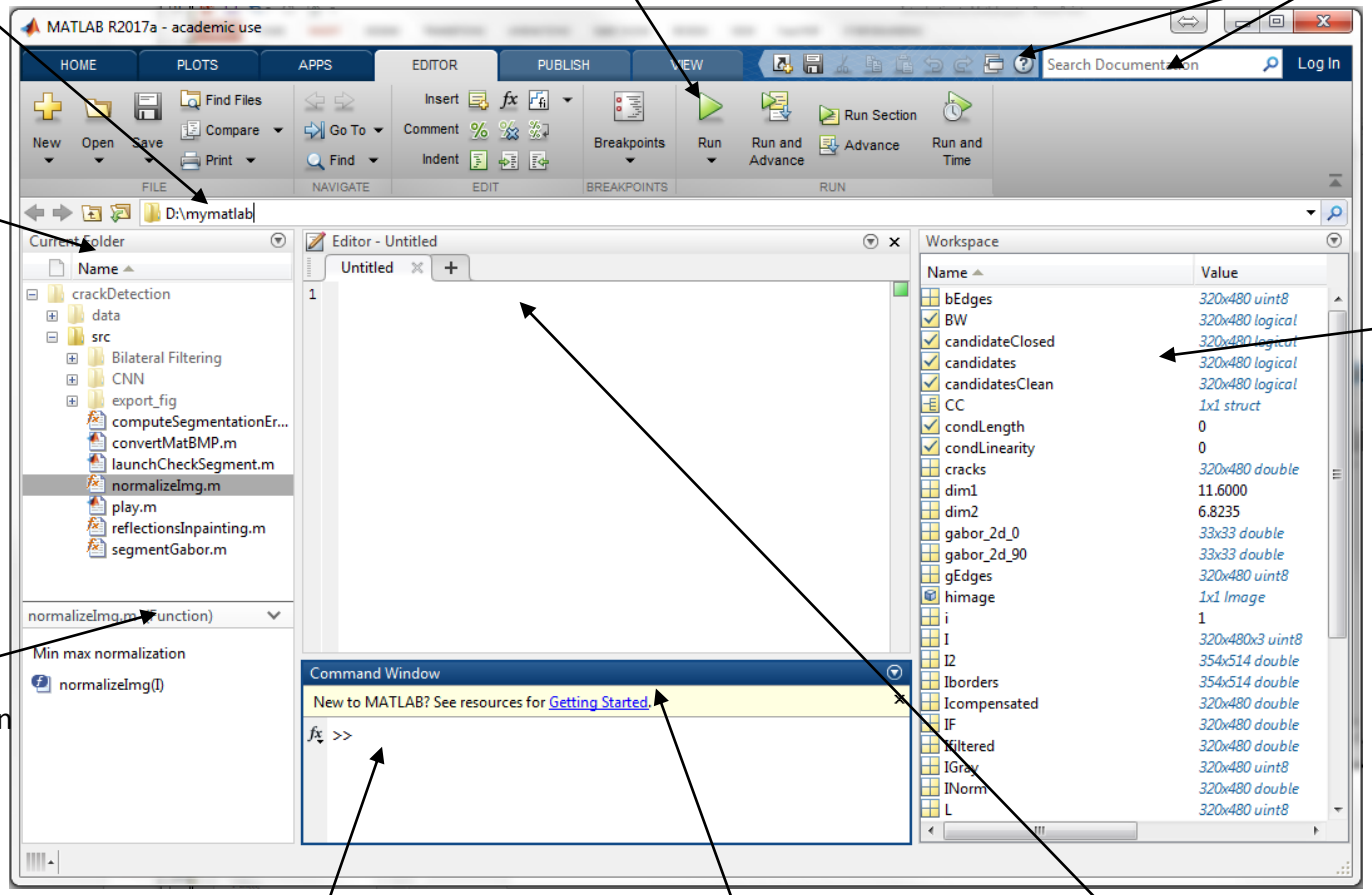
You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend MATLAB to solve particular classes of problems in these application areas.

http://www.mathworks.com

How to get Matlab

- http://www.unimi.it/ateneo/80207.htm

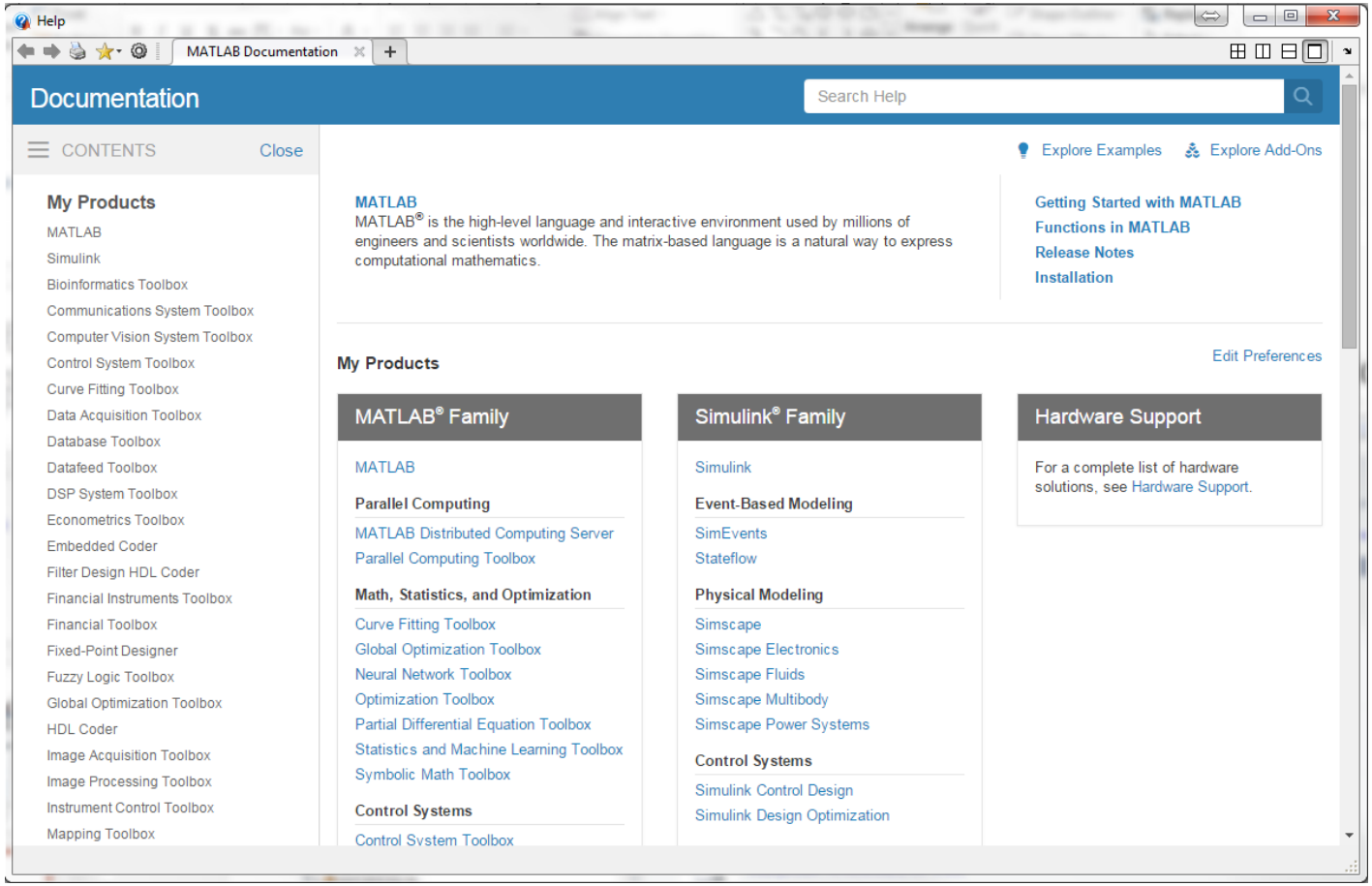
Matlab desktop



Help

VERY IMPORTANT!!!

It provides examples, demos, guides, and describes all the functions.



Arrays & matrices

Allocate arrays:

```
a = [];
a = [1, 2, 3];
a = [1; 2; 3];
a = 1:3;
a = zeros(1,3);
a = zeros(3,1);
a = ones(1,3);
a = ones(3,1);
a = ones(1,3) * 10;
a = ones(3,1) * 10;
a = b;                          (b array)
```

Arrays & matrices

Read elements of arrays:

```
b = a(1);
b = a([2,4,9]);          vector of indices
b = a(3:5);
a(1)  = 3;
a([2,4,9]) = [1,2,3];
a(3:5) = [1,2,3];
```

Number of elements appertaining to an array:

```
numberOfElements = numel(array);
```

Arrays & matrices

Two-dimensinal matrices:

The arrays are two-dimensional matrices with a dimension of size 1.

Examples

```
A = [1, 2, 3; 4, 5, 6];
A = zeros(3);                    3×3 elements
A = zeros(3,5);
B = A(2,[4,9]);
B = A([2,4,9]);                  it is possible to use a single index
B = A(:, 3:5);
```

Number of elements:

```
numberOfElements = size(A);                 returns [dim1, dim2]
numberOfElements = size(A, consideredDimension);
largestArrayDimension=length(array);
```

Arrays & matrices

N-dimensional matrices:

the read and allocation operations are similar to the previously described ones;

Examples

```
A = zeros(5,2,2);
B1= ones(5,2)*2;
B2= ones(5,2)*3;
A(:,:,1) = B1;
A(:,:,2) = B2;
c = A(1,2,1);
c = A(1,2,:);
D = A(:,2,:);
```

Useful function

squeeze

Important operators

Arithmetic operators :

+ Addition	
- Subtraction	
* Matrix multiplication	.* Multiplication
/ Matrix right division	./ Right division
\ Matrix left division	.\ Left division
^ Matrix power	.^ Power

Logical operators:

& and	| or
~ not	

Relational operators:

== Equal to	~ = Not equal to
< Less than	<= Less than or equal to
> Greater than	>= Greater than or equal to

Conditional operators

```
if a==b
        …
else
        …
end

switch a
        case {'linear','bilinear'}
                ….
        case 'cubic'
                …
        otherwise
                …
end
```

```
while a==b
        …
end

for i = 1 : N
        …
end

for i = 1 : 3 : N
        …
end
```

Files & data

File extensions:

.m = script, function files;

.mat = data files;

Save and load variables in MATLAB formatted binary files

save(fileName, ‘var1’, ‘var2’);

load(fileName);

Read and write generic files

fopen, fscanf, fprintf, etc.

Other useful notions

Functions

```
function [a, m] = additionMultiplication(v1, v2);
[a, m] = additionMultiplication(2, 3);
```

Useful commands

close all	deletes all figures
clear all	removes the variables, scripts, and functions from memory
clear variables	removes only variables
clc	clears all input and output from the Command Window display

Visualization

Text

fprintf()

One-dimensional data

plot() bar() …

Two-dimensional data

imshow() image() imagesc() …

Three-dimensional data

plot3() surf() mesh() ...

Figures and graphs

Use **HELP!!!**

Useful commands

figure	creates an empty figure
subplot(2,3,1)	creates a 2x3 grid of graphs and creates axes in position 1
hold on	retains the current graph and adds another graph to it
hold off	resets hold state to the default behavior

N.B. by default, MATLAB clears the existing graph and resets axes properties to their defaults before drawing new plots.

Exercises

1. Create a vector composed by these elements
   - (2, 4, 6, 8)
   - (1, 1/2, 1/3, 1/4, 1/5)
2. Given the array A = $\begin{pmatrix} 2 & 4 & 1 \\ 6 & 7 & 2 \\ 3 & 5 & 9 \end{pmatrix}$ provide the commands needed to
   - assign the first row of A to a vector called x1
   - assign the last 2 rows of A to an array called y
   - compute the sum over the columns of A
3. Create the matrices A = $\begin{pmatrix} 1 & 2 & 3 \\ 0.1 & 0.2 & 0.3 \\ 10 & 20 & 30 \end{pmatrix}$ B = $\begin{pmatrix} 4 & 5 & 6 \\ 0.4 & 0.5 & 0.6 \\ 40 & 50 & 60 \end{pmatrix}$ and:
   - sum A and B and store the result in C
   - subtract B from A and store the result in D
   - multiply A and B element by element and store the result in E
   - divide A and B element by element and store the result in F
   - multiply A and B in the matrix space and store the result in G

Exercises

4. Create the vectors v1=(1,2,3,4,5) and v2=(10,20,30,40,50) and:
   - sum v1 and v2 and store the result in vs
   - subtract v2 from v1 and store the result in vd
   - compute the scalar product of v1 and v2 and store the result in s.
5. Execute the following instructions:
   - A=ones(3,2); B=2*ones(2,3); A*B; A(2,3)=2; A*B
     Why there is an error message?
   - u=0:3; v=(-3:-1:0)'; W=u.*v
     Why there is an unexpected result?
6. Given the vector x = (1, 8, 3, 9, 0, 1), create a short set of commands that will
   - add up the values of the elements (Check with sum.)
   - computes the running sum (for element j, the running sum is the sum of the elements from 1 to j, inclusive. Check with cumsum.)
   - computes the sine of the given x-values (should be a vector)

Exercises

7. Write a script that, given in input a natural number *k*, compute the first k elements of the Fibonacci series, given by the recurrence formula:

$$F_0=1,\ F_1=1,\ F_i=F_{i-1}+F_{i-2} \qquad \forall i \geq 2$$

8. Write a script computing, for a natural number *k*, the ratio:

$$r_k = \frac{F_{k+1}}{F_k}$$

(where F_k are the Fibonacci numbers defined in the previous exercise). Verify that, for a large k, r_k converges to the value $(1 + \sqrt{5})/2$

9. Write a function that, given three values *a*, *b (a<=b)* and *h>0*, compute the values of the function

$$f(x)= 2\sin(8x) - \log(x^2 + 1)$$

on a grid of equally spaced points in the interval *[a,b]*, with step *h*.

Exercises

10. Given x=1:30, plot the functions sin(x) and cos(x)
- in two distinct windows
- in the same window divided by two along the x axis
- in the same plot of the same window, using different colors, and creating
- a legend

For each graph, plot the title and the axis names.

11. Load the file penny.mat (Matlab libraries). This file describes the surface of a penny. Try these functions:
- imshow()
- surf()
- mesh()
- plot3()
- pcolor()
- contour()

Exercises

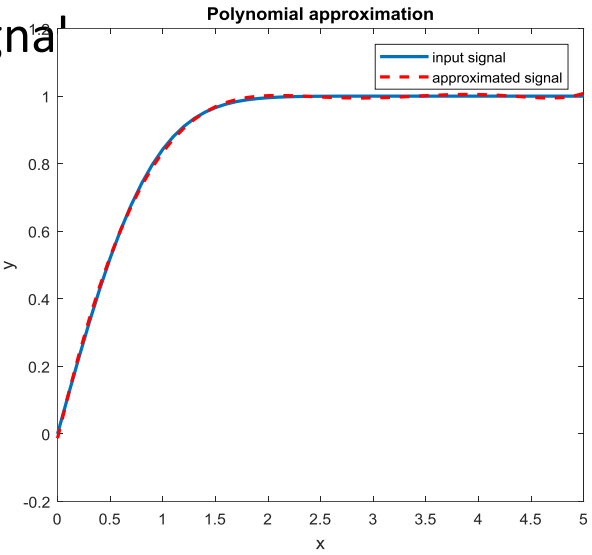
12. Obtain the least square approximation of the signal

```
x = (0: 0.1: 5)';
y=erf(x);
```

Suggestion: polyfit()

Expected results:

Order 6 approximation:
mean error = 4.386314e-03;
standard deviation error = 2.722904e-03



13. Obtain the linear approximation of a sinusoidal signal

simulate a sinusoidal signal x(n)=*a* sin(*b* * *n* – *c*) + *d*, for *b*>0

estimate a sequence $x_s(m)$ obtained by sampling the signal $x(n)$

(e.g.: if n contains 500 values, m should contain 50)

computate the signal $x_e(m+k)$, obtained by linear interpolation

(suggestion: interp1())

evaluate the accuracy

More examples

http://www.facstaff.bucknell.edu/maneval/help211/exercises.html

R. J. Braun, " Beginning Matlab Exercises", Department of Mathematical Sciences, University of Delaware, http://www.math.udel.edu/~braun/M349/Matlab_probs2.pdf

http://www.dm.unibo.it/~piccolom/didattica/num_met/Intro_Matlab.pdf

https://it.mathworks.com/examples/matlab/mw/matlab_featured-ex16585494-creating-3-d-plots?s_tid=examples_p1_MLT