UNIVERSITÀ DEGLI STUDI
DI MILANO

Introduction to Matlab

Ruggero Donida Labati

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

Why?

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, 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 the MATLAB environment to solve particular classes of problems in these application areas.

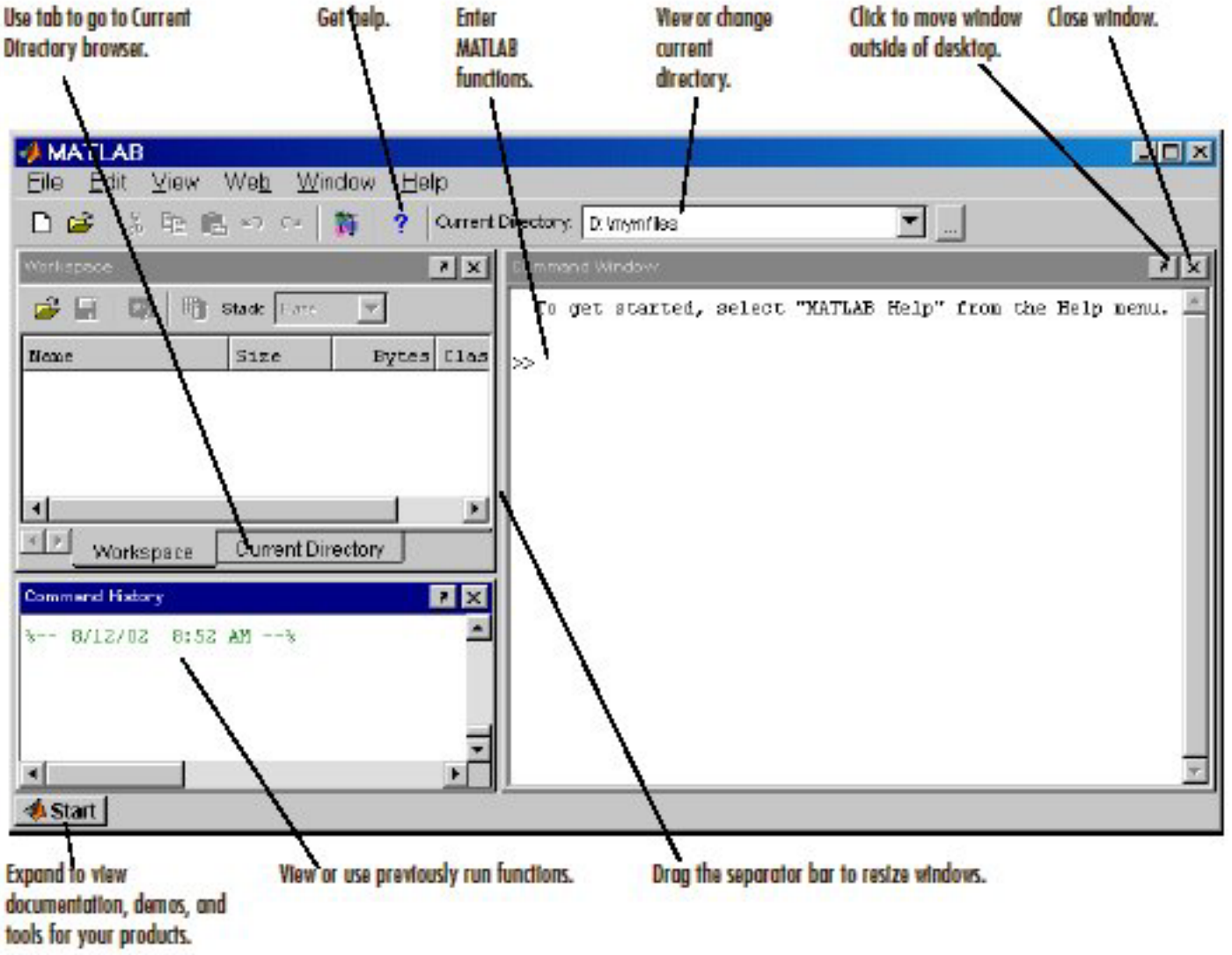
http://www.mathworks.com

How to get Matlab

Academic free license

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

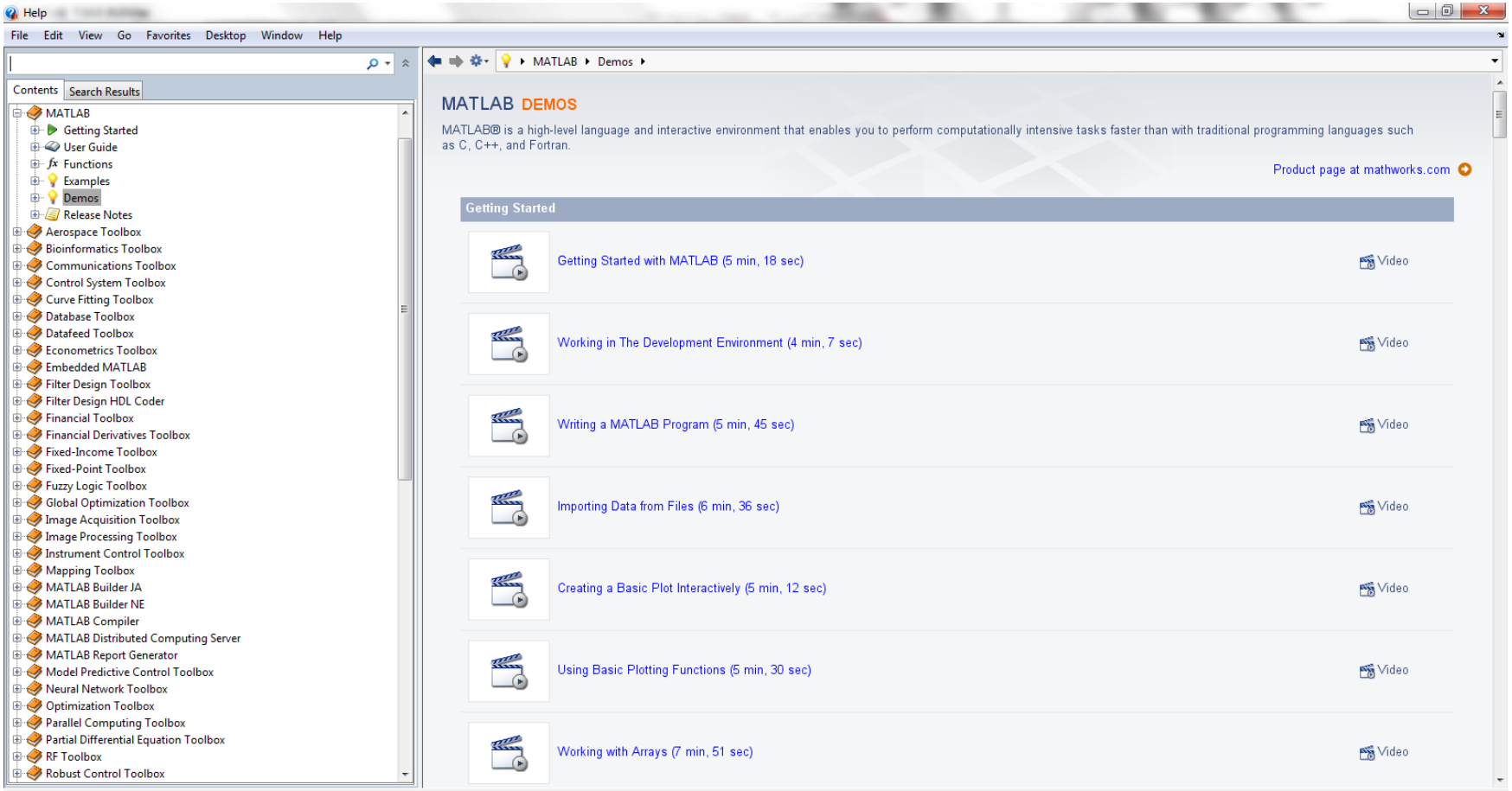
Matlab desktop



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 = length(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);
```

Arrays & matrices

N-dimensional matrices:

the read operation is similar to the previously described ones;

the allocation is more complex.

Examples

```
A = [];
B1 = zeros(5,2);
B2= zeros(5,2);
A(:,:,1) = B1;
A(:,:,2) = B2;
c = A(1,2,1);
c = A(1,2,:);
D = A(:,2,:);
```

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 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
clc	clears all input and output from the Command Window display

Visualization

Text

fprintf()

One-dimensional data

plot() plotyy() 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)	2 graphs in the same figure
subplot(2,3)	2 × 3 graphs in the same figure
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

Creare a vector composed by theese elements

- 2, 4, 6, 8
- 1, 1/2, 1/3, 1/4, 1/5

Given the array A = [2 4 1 ; 6 7 2 ; 3 5 9], 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

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)

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

Exercises

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

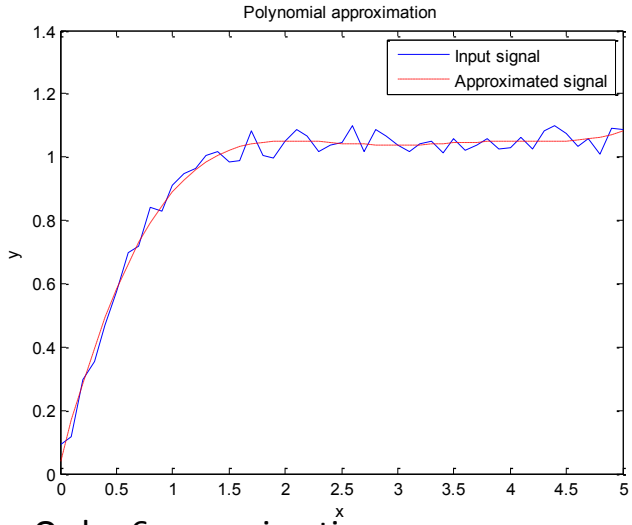
For each graph, plot the title and the axis names.

Exercises

Least square approximation of the signal

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

Excpected results



Order 6 approximation:

mean error = 0.000000;

standard deviation error = 0.029241

Suggestion: polyfit()

Exercises

2D graphs
http://www.mathworks.it/products/matlab/examples.html?file=/products/demos/shipping/matlab/penny.html

3D surfaces
http://www.mathworks.it/it/help/matlab/examples/3-d-plots.html