

Matlab Tutorial for Beginners

Quan Wang, ECSE, RPI
IEA Graduate TA for Section 3
wangq10@rpi.edu

My Office Hours

- * Location: JEC 2001
- * Monday: 4:00 – 5:00
- * Tuesday: 1:00 – 5:00
- * Friday: 1:00 – 4:00

What Matlab is?

- * It is a Programming Language for Scientific Computing
- * It is a Numerical Computing Environment where you can test or implement algorithms
- * It is a Software which runs on Windows/Mac OS/Linux

What Matlab is not?

- * It is not used to develop high efficiency executable applications (compared with C++, Java, Python)
- * Generally, it cannot be compiled (compared with C++)
- * Generally, we do not use it for symbolic computation (compared with Mathematica and Maple)

Before Using Matlab

* Get Matlab

1. Go to **Rensselaer Help Desk**

<http://helpdesk.rpi.edu/setup.do>

2. Software -> Software to install -> Matlab

3. Get the license with your RPI account

4. Download and install

Substitutes of Matlab

- * With your RPI Matlab license, you are not allowed to use Matlab off campus, unless you use VPN
- * You can also use:
 - Scilab
 - Free, cross-platform, similar to Matlab
 - Freemat
 - Free, cross-platform, similar to Matlab
 - Python
 - Free, cross-platform, different grammar, high efficiency, various libraries, popular, can be packaged to .exe files

Beginning with Matlab

- * Start Matlab, and you will see ...

File Edit View Debug Parallel Desktop Window Help

The screenshot displays the MATLAB 7.10.0 (R2010a) interface. The menu bar at the top includes File, Edit, View, Debug, Parallel, Desktop, Window, and Help. Below the menu bar is a toolbar with various icons for file operations and execution. The main workspace is divided into several panes: a Current Folder pane on the left showing a file named test1.m (0 KB); an Editor pane in the center for editing code; a Command Window at the bottom center for running commands; a Workspace pane on the right for viewing variables; and a Command History pane at the bottom right for tracking executed commands. The Command History pane shows a sequence of commands including imagesc, colormap, figure, and C3=BoundMirrorShrink, along with their execution times.

Shortcuts [How to Add](#) [What's New](#)

Current Folder: test1.m (0 KB)

Editor - /Users/Quan/Documents/MA...

Command Window: fx >>

Workspace:

Name	Value
------	-------

Command History:

```
imagesc(sum_f)
colormap gray
imagesc(ma_f)
imagesc(max_f)
colormap gray
figure;
C3=BoundMirrorShrink(medf.
imagesc(C);
colormap gray;
%-- 9/5/11 12:44 PM --%
%-- 9/5/11 2:48 PM --%
clc
```

Menu bar

The screenshot displays the MATLAB 7.10.0 (R2010a) environment. The main window is the Editor, which is highlighted with a red border. The Editor window title is "Editor - /Users/Quan/Documents/MA...". The Editor toolbar includes icons for file operations (New, Open, Save, Print), editing (Undo, Redo), and execution (Run, Stop). The Editor content area shows a single line of code: "1".

Other visible windows include:

- Current Folder:** Shows a file named "test1.m" with a size of 0 KB.
- Workspace:** An empty table with columns "Name" and "Value".
- Command History:** Lists the following commands:

```
imagesc(sum_f)
colormap gray
imagesc(ma_f)
imagesc(max_f)
colormap gray
figure;
C3=BoundMirrorShrink(medf.
imagesc(C);
colormap gray;
%-- 9/5/11 12:44 PM --%
%-- 9/5/11 2:48 PM --%
clc
```
- Command Window:** Shows the prompt "fx >>".

The bottom of the screen shows the Windows taskbar with the Start button.

Editor

/Users/Quan/Documents/MATLAB

Shortcuts How to Add What's New

Current Contents

Name	Size
test1.m	0 KB

Editor - /Users/Quan/Documents/MA...

1

Command Window

```
fx >>
```

Workspace

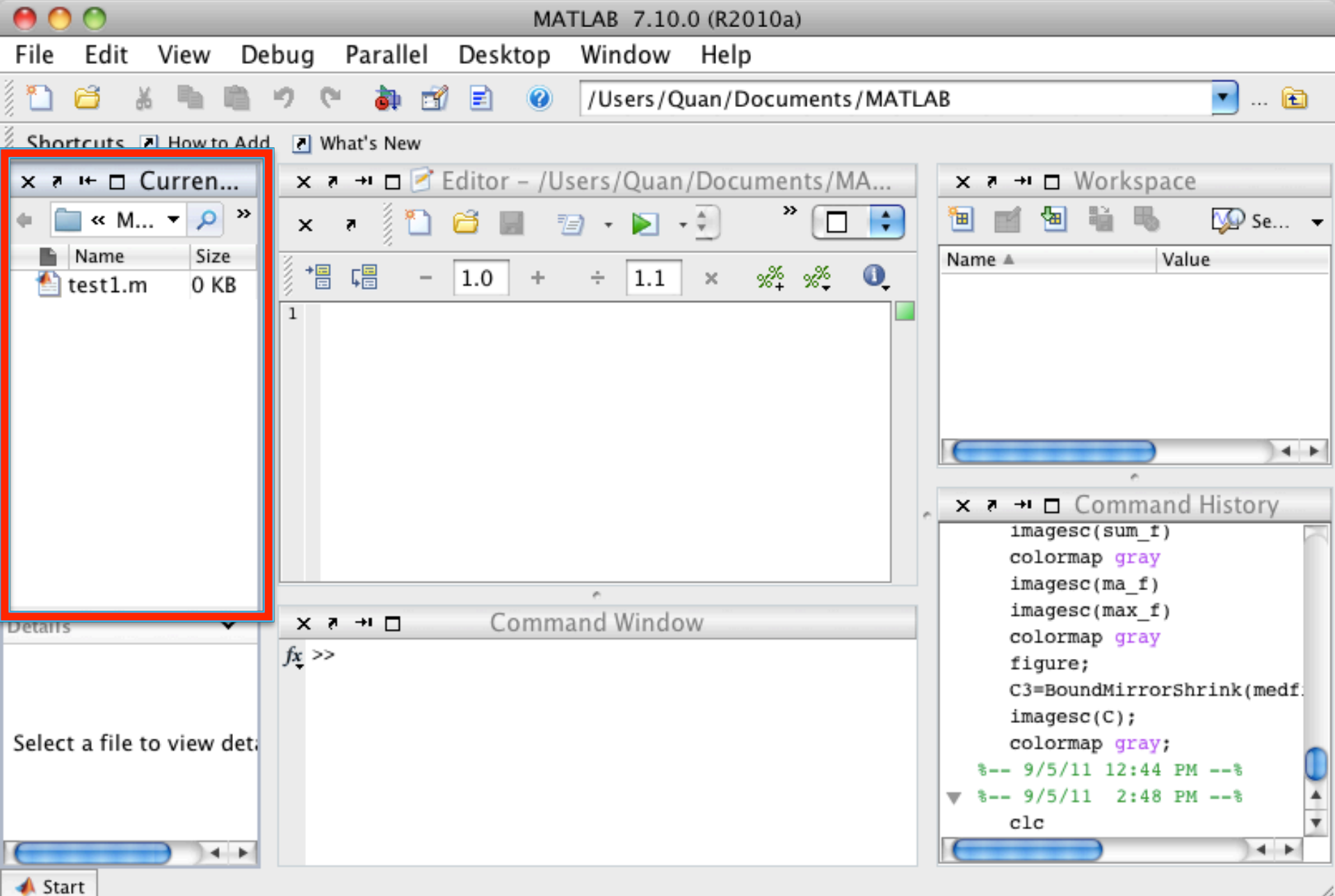
Name	Value
------	-------

Command History

```
imagesc(sum_f)
colormap gray
imagesc(ma_f)
imagesc(max_f)
colormap gray
figure;
C3=BoundMirrorShrink(medf.
imagesc(C);
colormap gray;
%-- 9/5/11 12:44 PM --%
%-- 9/5/11 2:48 PM --%
clc
```

Start

Command window



Current folder

Current Contents

Name Size

test1.m	0 KB
---------	------

Details

Select a file to view details

Editor - /Users/Quan/Documents/MA...

1

Command Window

```
fx >>
```

Workspace

Name	Value
------	-------

Command History

```
imagesc(sum_f)
colormap gray
imagesc(ma_f)
imagesc(max_f)
colormap gray
figure;
C3=BoundMirrorShrink(medf.
imagesc(C);
colormap gray;
%-- 9/5/11 12:44 PM --%
%-- 9/5/11 2:48 PM --%
clc
```

Workspace

Current Contents

Name Size

test1.m	0 KB
---------	------

Details

Select a file to view details

Editor - /Users/Quan/Documents/MA...

1

Command Window

```
fx >>
```

Workspace

Name	Value
------	-------

Command History

```
imagesc(sum_f)
colormap gray
imagesc(ma_f)
imagesc(max_f)
colormap gray
figure;
C3=BoundMirrorShrink(medf.
imagesc(C);
colormap gray;
%-- 9/5/11 12:44 PM --%
%-- 9/5/11 2:48 PM --%
clc
```

Command history

Example: Entering commands and expressions

- * In the command window, enter some commands or expressions, then press **Enter** key, see what is displayed in command window, and what happened in workspace
- * Semicolon: do not display results immediately

```
>> 1+2^3-4.5/6  
ans =  
8.2500
```

```
>> 1+2^3-4.5/6;  
>> ans  
  
ans =
```

```
8.2500
```

Variables

- * In Matlab, data can be represented by variables
- * Each variable has a case-sensitive variable name, which is a combination of letters, digits, and underscore characters, beginning with a letter
- * It is always good for you to use readable variable names

Variables

```
>> a=3;  
>> b=4;  
>> bb=2;  
>> bb2=a*b*bb
```

```
bb2 =
```

```
24
```

Anyone knows what is he doing here?

Compare two pieces of Matlab code, both of which are computing the total area of several rectangles

```
>> Width=3;  
>> Length=4;  
>> NumberOfRectangles=2;  
>> Area_AllRectangles=Width*Length*NumberOfRectangles
```

```
Area_AllRectangles =
```

```
24
```

Natural language alike variable names are much more readable!

a=a+1?

- * Understand the command `a=a+1`
- * How does the computer execute this command
 1. Look at the right-hand-side **expression**: `a+1`
 2. Compute the expression: `a+1`
 3. Look at the left-hand-side **variable**: `a`
 4. Assign the right-hand-side expression to the left-hand-side variable
- * e.g.
 1. Before command: `a=5`
 2. After command: `a=6`

```
>> a=5
```

```
a =
```

```
5
```

```
>> a=a+1
```

```
a =
```

```
6
```

Workspace Commands

- * `clc`: clear command window
- * `clear`: remove all variables from memory
- * `clear var1 var2 ...`: remove indicated variables
- * `who`: list all variables in memory
- * ...

Predefined Symbols

- * Complex numbers

- * If i and j are not defined by user, they are the imaginary units.

- * Ratio of circle's circumference to diameter

- * π

```
>> 3+i
```

```
ans =
```

```
3.0000 + 1.0000i
```

```
>> 9-4*j
```

```
ans =
```

```
9.0000 - 4.0000i
```

```
>> pi
```

```
ans =
```

```
3.1416
```

Arrays

- * Arrays can be defined using square brackets
- * Most functions take arrays/matrices as input arguments
- * e.g.

```
>> u=[0:0.1:0.3]
```

```
u =
```

```
0 0.1000 0.2000 0.3000
```

```
>> sin(u)
```

```
ans =
```

```
0 0.0998 0.1987 0.2955
```

Built-in Functions

- * $\exp(x)$
- * $\text{sqrt}(x)$
- * $\log(x)$
- * $\log_{10}(x)$
- * $\cos(x)$
- * $\sin(x)$
- * $\tan(x)$
- * $\text{acos}(x)$
- * ...

plot

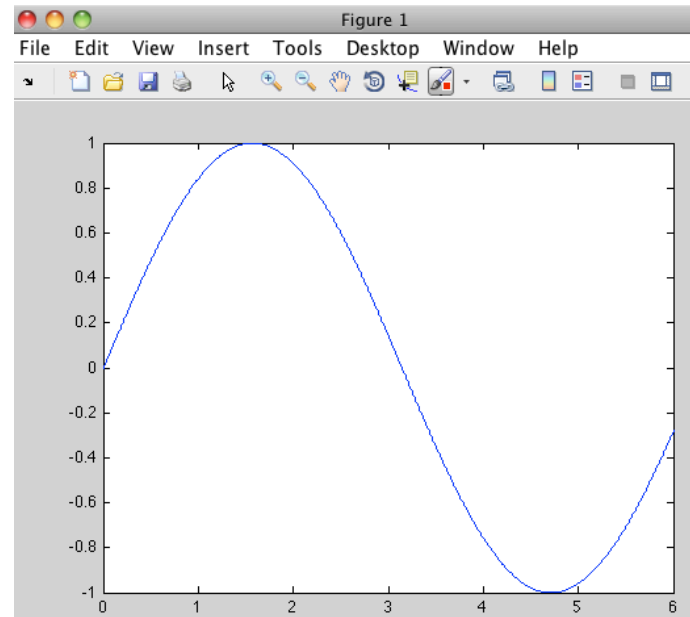
- * plot command is for drawing 2D lines:

- * e.g.

- * `>> x=[0:0.01:6];`

- * `>> y=sin(x);`

- * `>> plot(x,y)`



plot related commands

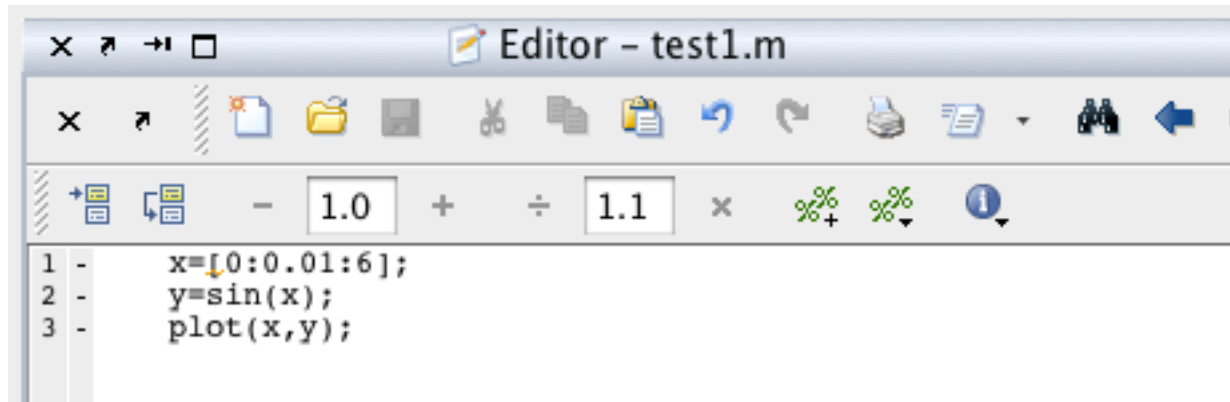
- * `figure`: create a new figure window
- * `hold on`: enable next plot to superimpose on current figure
- * `hold off`: enable next plot to erase current figure
- * `grid`: put grid lines on the plot
- * `title('text')`: add text at the top on the plot
- * `xlabel('text')`: add text to x-axis
- * `ylabel('text')`: add text to y-axis

M-files!

- * Most of the time, you do not want to type the commands line by line, again and again
- * Instead: write all commands in one file, and run it at different time
- * M-file: the script of Matlab code

Create an M-file

- * Menu bar -> File -> New -> Script
- * Write your code here
- * Save it as a .m file

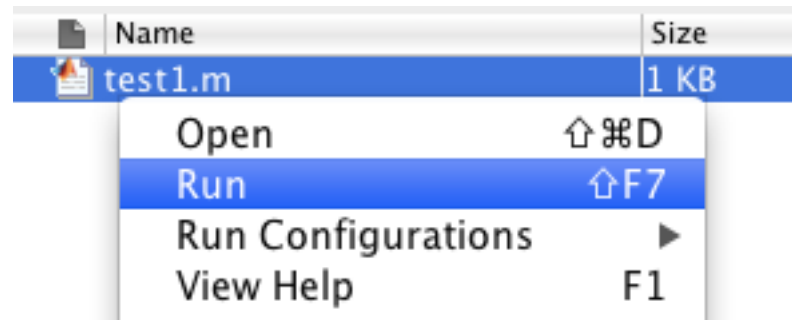


The image shows a screenshot of a MATLAB Editor window titled "Editor - test1.m". The window has a standard toolbar with icons for file operations (new, open, save, print) and editing (undo, redo, cut, copy, paste). Below the toolbar is a numeric keypad with buttons for digits 1.0 and 1.1, and mathematical operators like minus, plus, multiply, and divide. The main editing area contains three lines of MATLAB code:

```
1 - x=[0:0.01:6];  
2 - y=sin(x);  
3 - plot(x,y);
```

Run your M-file

- * One way:
 - * Right click your M-file, choose Run



Run your M-file

- * Another way:
 - * Select the code you want to run, then right click, choose Evaluate Selection


```
x=[0:0.01:6];  
y=sin(x);  
plot(x,y);
```

Evaluate Selection	⇧F7
Open Selection	⇧⌘D
Help on Selection	F1
Function Browser	⇧F1
Function Hints	^F1

Comments

- * Comments are lines which are not executed
- * Comments begin with a percent sign
- * Use two percent signs to highlight different sections of your code

Comments



A toolbar for a MATLAB calculator, showing various mathematical operators and functions. From left to right: a stack of three documents icon, a plus sign, a document with a plus sign icon, a minus sign, a text box containing '1.0', a plus sign, a division sign, a text box containing '1.1', a multiplication sign, and a percent sign with a plus sign.

```
1 %% Computing x and y
2 % x is an array
3 - x=[0:0.01:6];
4 - y=sin(x); % y is also an array
5 %% Plot a 2D line
6 - plot(x,y);
```

Debug

- * What if you are not sure where the problem of the code is?
- * Set a breakpoint, where your program will stop for you to check current variables in workspace

Debug

- * Setting a breakpoint

```
1 - x=[0:0.01:0.5];  
2 - y=sin(x);  
3 ● z=y+2;  
4
```

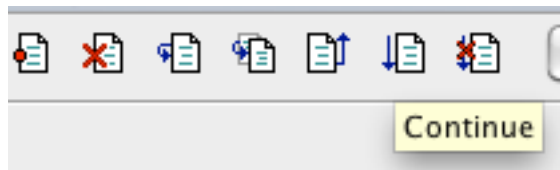
- * Run the program, and it will stop before your breakpoint (the command on the breakpoint line will not be executed)

```
1 - x=[0:0.01:0.5];  
2 - y=sin(x);  
3 ● → z=y+2;  
4
```

Name ▲	Value
x	<1x51 double>
y	<1x51 double>

Debug

* Continue running



Name ▲	Value
x	<1x51 double>
y	<1x51 double>
z	<1x51 double>

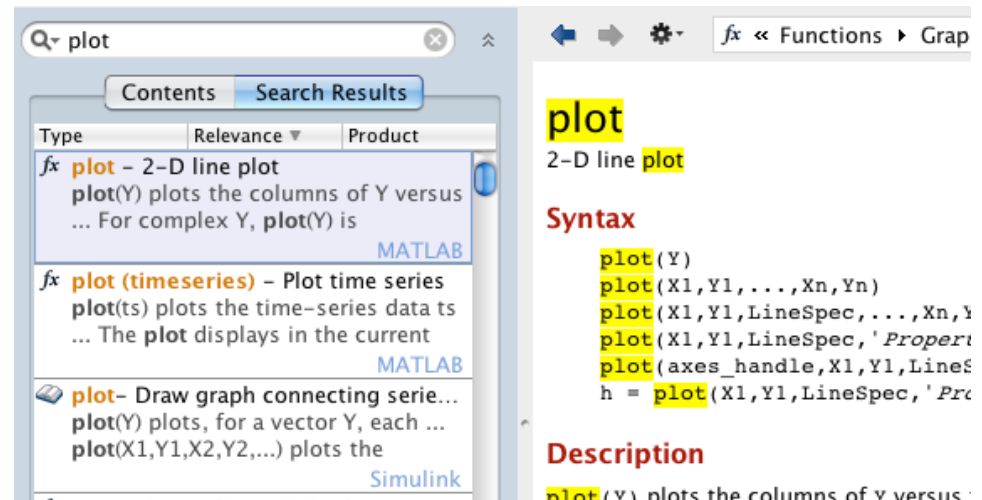
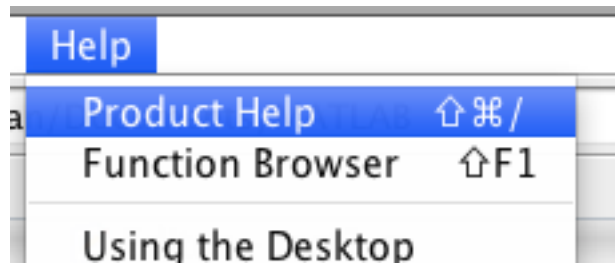
Need help?

- * Google it with key words!



Need help?

* Matlab product help!



Need help?

- * Come to ask your TA!

Homework

- * Install Matlab on your laptop
- * Play with simple expressions and commands
- * Write some M-files
- * Design and plot some 2D curves (e.g. try to draw a heart)