Introduction to Matlab

Kristian Sandberg
Department of Applied Mathematics
University of Colorado

Goal

The goal with this worksheet is to give a brief introduction to the mathematical software Matlab. After completing the worksheet you should know how to start Matlab, how to use the elementary functions in Matlab and how to use Matlab to plot functions.

What is Matlab?

Matlab is widely used in all areas of applied mathematics in education and research at universities and in the industry. Matlab stands for MATrix LABoratory and the software is built up around vectors and matrices. This makes the software particularly useful for linear algebra but Matlab is also a great tool for solving algebraic and differential equations and for numerical integration. Matlab has powerful graphic tools and can produce nice pictures in both 2D and 3D. It is also a programming language (similar to C) and is one of the easiest programming languages for writing mathematical programs. Matlab also has some tool boxes useful for signal processing, image processing, etc.

How to start Matlab

Mac: Double-click on the icon for Matlab.

PC: Choose the submenu "Programs" from the "Start" menu. From the "Programs" menu, open the "Matlab" submenu. From the "Matlab" submenu, choose "Matlab".

Unix: At the prompt, type `matlab.` When you want to quit Matlab when using a Unix text based interface, type `exit.`

The Matlab environment

Note: From now on an instruction to press a certain key will be denoted by < >, e.g., pressing the enter key will be denoted as <enter>. Commands that should be typed at the prompt, will be written in `courier` font.

The Matlab environment (on most computer systems) consists of menus, buttons and a writing area similar to an ordinary word processor. There are plenty of help functions that you are encouraged to use. The writing area that you will see when you start Matlab, is called the *command window*. In this window you give the commands to Matlab. For example, when you want to run a program you have written for Matlab you start the program in the command window by typing its name at the prompt. The command window is also useful if you just want to use Matlab as a scientific calculator or as a graphing tool. If you write longer programs, you will find it more convenient to write the program code in a separate window, and then run it in the command window (discussed in Intro to programming).

In the command window you will see a prompt, on Macs and PCs this prompt is >> . You type your commands immediately after this prompt. Once you have typed the command you wish Matlab to perform, press <enter>. If you want to *interupt a command* that Matlab is running, type <ctrl> + <c>.

Useful functions and operations in Matlab

Using Matlab as a calculator is easy.

Example: Compute 5 sin(2.5$^{3\text{-pi}}$)+1/75. In Matlab this is done by simply typing

```
5*sin(2.5^(3-pi))+1/75
```

at the prompt. Be careful with parantheses and don't forget to type `*` whenever you multiply!

Note that Matlab is *case sensitive*. This means that Matlab knows a difference between letters written as lower and upper case letters. For example, Matlab will understand `sin(2)` but will not understand `Sin(2)`.

Here is a table of useful operations, functions and constants in Matlab.

Operation, function or constant	**Matlab command**
+ (addition)	`+`
- (subtraction)	`-`
× (multiplication)	`*`
/ (division)	`/`
|x| (absolute value of x)	`abs(x)`
square root of x	`sqrt(x)`
e^x	`exp(x)`
ln x (natural log)	`log(x)`
$\log_{10}$ x (base 10 log)	`log10(x)`
sin x	`sin(x)`
cos x	`cos(x)`
tan x	`tan(x)`
cot x	`cot(x)`
arcsin x	`asin(x)`
arccos x	`acos(x)`
arctan x	`atan(x)`
arccot x	`acot(x)`
n! (n factorial)	`gamma(n+1)`
e (2.71828...)	`exp(1)`
π (3.14159265...)	`pi`
i (imaginary unit, sqrt(-1))	`i`

Exercises

Compute the following expressions using Matlab:

- 3cos(pi)
- 1+1+1/2+1/6+1/24-e
- ln (1000+2^{pi-2})
- $e^{i\ pi}$
- The number of combinations in which 12 persons can stand in line. (Hint: Use factorials.)

Variables in Matlab

We can easily define our own variables in Matlab. Let's say we need to use the value of 3.5sin(2.9) repeatedly. Instead of typing `3.5*sin(2.9)`over and over again, we can denote this variable as x by typing the following:

```
x=3.5*sin(2.9)
```

(Please try this in Matlab.) Now type

```
x+1
```

and observe what happens. If we do not want to see the result of a calculation we can use semi-colon to surpress the print-out on the screen even though Matlab still performs the command in "the background". If you defined x as above, now type

```
y=2*x;
```

```
y
```

and observe what happened. In many cases we want to know what variables we have declared. We can do this by typing `whos`. If you want to erase all variables from the Matlab memory, type `clear`. To erase a specific variable, say `x`, type `clear x`. To clear two specific variables, say `x` and `y`, type `clear x y`, that is separate the different variables with a space.

Vectors and matrices in Matlab

We create a vector in Matlab by putting the elements within [] brackets.

Example: `x=[ 1 2 3 4 5 6 7 8 9 10]`

We can also create this vector by typing `x=1:10.` The vector (`1 1.1 1.2 1.3 1.4 1.5`) can be created by typing `x=[ 1 1.1 1.2 1.3 1.4 1.5 ]` or by typing `x=1:0.1:1.5.`

Matrices can be created according to the following example. The matrix A=? is created by typing

```
A=[1 2 3 ; 4 5 6 ; 7 8 9 ],
```

i.e., rows are separated with semi-colons. If we want to use a specific element in a vector or a matrix, study the following example:

Example:

```
x=[10 20 30]
```

```
A=[ 1 2 3 ; 4 5 6 ; 7 8 9]
```

```
x(2)
```

```
A(3,1)
```

Here we extracted the second element of the vector by typing the variable and the position within parantheses. The same principle holds for matrices; the first number specifies the row of the matrix, and the second number specifies the column of the matrix.

If the matrices (or vectors which are special cases of a matrices) are of the same dimensions then matrix addition, matrix subtraction and scalar multiplication works just like we are used to.

Example: Type

```
x=[ 1 2 3]

y =[4 5 6]

a=2

x+y

x-y

a*x
```

and observe what happens.

If want to apply an operation such as squaring each element in a matrix we have do use a dot . before the operation we wish to apply. Type the following commands in Matlab.

```
x=1:10

x.^2

A=[1 2 3 ; 4 5 6 ; 7 8 9 ]

A.^2

A^2
```

and observe the result. Note the dot and what it accomplishes! All built-in functions such as `sin, cos, exp` and so on automatically act elementwise on a matrix. Type

```
y=[0 1/4 1/2 3/4 1]

y=pi*y

sin(y)
```

and observe the result.

How to plot with Matlab

There are different ways of plotting in Matlab. The following technique is probably the most useful one.

Example: Plot sin(x^2) on the interval [-5,5]. To do this, type the following:

```
x=-5:0.01:5

y=sin(x.^2)
```

```
plot(x,y)
```

and observe what happens. The following commands are useful when plotting:

Graphing functions	**Matlab command**
Label the horizontal axis.	`xlabel('text')`
Label the vertical axis.	`ylabel('text')`
Attach a title to the plot.	`title('text')`
"Keep plotting in the same window."	`hold on`
Turn off the "keep-plotting-in-the-same-window-command".	`hold off`

Note that all text must be put within `' '`. The last two commands (`hold on` and `hold off`) are best explained by trying them next time you plot.

Exercises

- Plot sin(x) on the interval [-pi,pi] using spacing 0.5, 0.1 and 0.01 between the points where you will sample the function. (This will change the resolution). Experiment with the `hold on` command.
- Attach labels to the axis of the previous plot and give a title to the graph.
- Plot 5 cos(x^2+1) on [-2pi,2pi]. Note that the squaring operation will require you to use the dot `.` in order for the squaring operation to act on each element individually. However, the addition operation (+) automatically acts on elements individually.