Lab: Introduction

1 Installing MATLAB

We will be learning to work with MATLAB (matrix laboratory) in this course and using this program to explore mathematical modeling in biology. If you have already successfully installed MATLAB on your computer please skip to the next section. MATLAB is available to William and Mary students through the Technology Support Center.

1.1 MATLAB installation for Windows

To follow these instructions you will need to have internet access.

1. Log into your **WM home** page
2. Scroll down the section labeled **Resources** until you reach **Technology Support**
3. Open the link labeled **Technology Support Center (TSC)**
4. In the **FAST NAVIGATION** box open the link titled **Software**
5. Now open the **Windows** link
6. Under the **Math and & Statistics Software** open **MATLAB r2007b for Students**
7. Choose the **Download Now** option and follow the instructions

1.2 MATLAB installation for Mac OS

To follow these instructions you will need to have internet access.

Contact the **Technology Support Center** for an appointment to have them install the student version of MATLAB for you. The Technology Support Center is open Mondays through Fridays from 8:00 a.m. until 5:00 p.m. and located in Jones Hall room 7. Please schedule your appointment by calling 221-4357. You must take your laptop and your **Mac OS X installation disk** to your appointment.

2 MATLAB Introduction: Basic Operations

After you have opened MATLAB you will see a window divided into three sections. The top left section allows you to see your current working directory and the lower left section shows your command history. The right side if this window is where we will begin. The MATLAB command line is denoted by two “greater than” symbols

```
>>
```

and we will also use this same notation when we wish you to type commands into the command line. To start, we will work with the most basic commands. When you wish for a command to be executed use the **return** or **enter** key.

2.1 Basic Commands: Help

The **help** command is one of the most useful commands when learning to use MATLAB. This command can be used in many ways.

- First, typing **help** will give you many options to choose from and may help you when you are unsure of a command, but know what type of math usually uses the command.
- Second: If you are unsure how a MATLAB function works, for instance the square root function, type **help sqrt**. This will result in an explanation of the desired function, lists several related functions and an occasional example.
- Third: This is perhaps the MOST IMPORTANT use of the help command. MATLAB has many predefined operations, however you can overwrite them. This means you could accidently overwrite a function that you need to use! For example, lets say I want to call my parameter “j”. When I type in **help j** I see this is already a defined function (imaginary number designation). This implies that ”j” may not be the best name for a parameter.

Let’s try some of these commands:

```
>> help
>> help sqrt
>> help j
>> help pi
>> help PI
```

You may have noticed that **pi** has already been designated as the irrational number we have grown to love (or hate), but **PI** has not yet been designated by MATLAB. MATLAB is case sensitive which is a good thing to remember when you are defining your own parameters and variables.

2.2 Basic Commands: Clear

There are a few functions which allow you to erase anything you have stored in memory, including figures as well as command line output. The most notable example is with **clear**. To clear everything including anything you have previously stored in memory type

```
>> clear all
```

If you only want to clear certain items a more specific command can be used.

```
>> clc
>> clf
>> clear a
```

These commands will clear your command window, figures and the parameter “a”, respectively.

2.3 Basic Commands: Who

In a few moments we will be defining some parameters. In this case we will only define three parameters, but you can imagine a situation where you have to use many parameters. You may have guessed that the **help** command will not remind you of your parameters, but there are two commands that will. First lets define parameter a to be 3, parameter b to be -1 and parameter c to be 2.

```
>> a = 3
>> b = -1
>> c = 2
```

Now assume you have forgotten what you called your parameters, or you want to define another parameter but do not want to overwrite any of your previous parameters. By typing **who** you will find a list of your parameters and by using **whos** you can even see your parameters’ sizes and other attributes.

```
>> who
>> whos
```

Now we are ready to start working with MATLAB.

3 MATLAB Introduction: Mathematical Operations

In the most basic sense, working with MATLAB is similar to working with a graphing calculator. Eventually we will start using MATLAB to preform operations which would be difficult on a graphing calculator, but for now let’s work with some basic math operations.
With MATLAB, the following order of operations is preserved:

- Parentheses
- Exponents
- Multiplication / Division
- Addition / Subtraction

To enter numbers and basic math functions use the traditional symbols: addition (+), subtraction (-), multiplication (*), division (/) and exponents (∧) .

```
>> -1*3∧2+ 4/5 - 0.23
>> (-1*3)∧2+4/(5-0.23)
>> 3/5;
```

Notice when a **semicolon (;)** is placed at the end of the command, MATLAB will not display the answer. This will be useful when we set parameters values and when we eventually learn to program and do not wish to see the answer at every step. To enter more than one command line at a time, use **shift return** to move down to the next line. When you are finished hit **return**

```
>> 3∧(-1)
3/4
7/4
```

3.1 MATLAB functions

As we have alluded to, MATLAB has many built in functions. Here is a list of a few useful functions:

- For the value pi use **pi**
- For square roots use **sqrt()**
- For imaginary numbers use either **i** or **j**
- For exponentials with "e" use **exp()**
- For natural logarithms use **log()**
- For base 10 logarithms use **log10()**
- For a random number generator use **rand()**
- For trig functions use the three letter abbreviation in lower case. Example: **sin()**
- For inverse trig functions place the letter "a" before the function. Example: **asin()**
- For a list of more functions look under **help elfun** (elfun = elementary function)

Here are some examples to explore:

```
>> 2*pi

>> sqrt(4)
>> -sqrt(4)
>> sqrt(-9)
>> 3+4i
>> 3+4j
>> (1+i)*(2-i)
>> 3/(2-5i)

>> exp(1)
>> log(exp(2))
>> log10(1000)

>> rand(1)
>> rand(1)
>> rand(1)

>> sin(pi/2)
>> asin(1)

>> help elfun
```

You can use the **up arrow** to retrieve commands you have already done. Try this now: In the command line press the up arrow a few times and see what happens!

3.2 Symbolic Math in MATLAB

Symbolic math is useful to work with algebraic equations and formulas using letter representation in place of numbers. It is useful to use mathematical software to help with symbolic math. The MATLAB package offered through W&M will perform symbolic manipulations, but this may not be the case if you buy your own MATLAB software.

To work with symbolic math, we need to define variables and we should also make sure they have not been previously defined as something else! Please remember not to overwrite any built-in MATLAB functions when naming variables. If you are uncertain if the name you would like to use is already taken by MATLAB use the **help** command. If you are uncertain if the name you would like to use is already being used by yourself, use the **who** command. Variable names can not begin with a number or contain spaces, but may include letters, numbers and underscores.

For the next few examples, let's use the letters a,b,c,m,x, y and z. To begin we should make sure all these letters have been reset. You have two options: **clear all** or **clear** followed by the letter. The **clear all** command will clear all the variables you have worked with where as the **clear** command will only clear the variable listed after it. We will use both these methods throughout this section.

In order to manipulate variables (without giving them numerical values) we need to use the **sym** and **syms** commands. There are two different methods for defining variables. For example, if we want to work with the function

$$y = x^2,$$

we could define this function as a variable of x by using the following commands:

```
>> clear all
>> syms x
>> y = sym(x ∧ 2)
```

We could also use the **syms** command in a slightly different way to get the same result.

```
>> clear all
>> y = sym('x ∧ 2')
```

Now that we have a function, we can evaluate it for many different values for x. For example use MATLAB to find the function value y when $x = -1$, $x = 2$, and $x = 2i$. To do this we will need to use the **eval** command. In this example we will suppress some of the outputs.

```
>> x = -1;
>> eval(y)

>> x = 2;
>> eval(y)

>> x = 2*i;
>> eval(y)
```

You can also combine functions to make new ones. Let's define function z to be:

$$z = 2x$$

To put this in MATLAB we could use:

```
>> z = sym('2*x')
```

Now, if for some reason we wanted to define another couple of functions:

$$\begin{aligned} a &= x^2 - 2x \\ b &= \frac{x^2}{2x} \end{aligned}$$

we would not need to redefine everything, though we could. Instead we could use what we already have. For this example we do not want to use **clear all** as that would delete our previously defined functions!

```
>> clear a b
>> a = sym(y-z)
>> b = sym(z/y)
```

In case you prefer to view fractions in a more typical style use the command **pretty**. This command can also be used to check your work and make sure you have placed all the parentheses in the correct places.

```
>> pretty(b)
```

There are many different MATLAB functions to manipulate formulas. Experiment with the following functions: **solve**, **simplify**, **expand**, **factor**, **collect**, and **simple**. You may want to look these function up under the **help** menu. The main command to solve algebraic equations is **solve**. Suppose we wanted to solve the following equation for the variable x and label it X,

$$ax - b = 0.$$

We would type

```
>> clear a b x X
>> syms a b x X
>> X = solve('a*x-b', 'x')
```

MATLAB assumes the equations are set equal to zero, which implies you will need to be sure and set your equations to zero as well. MATAB can solve a variety of equations. For example we know the solutions for x to

$$ax^2 = b$$

are $\pm\sqrt{\frac{b}{a}}$. To solve this with MATLAB you must make sure the equation is set equal to zero. For this example, type

```
>> clear a b x X
>> syms a b x X
>> x = solve('a*x∧2-b', 'x')
```

We can also use this software to solve systems of equations, though the notation is a bit different. Here is how to solve the system

$$\begin{aligned} ax + by &= c \\ dx + fy &= g \end{aligned}$$

for variables x and y (labeled X and Y, respectively).

```
>> clear a b c d f g x X y Y
>> syms a b c d f g x X y Y
>> [X,Y]=solve('a*x+b*y-c', 'd*x+f*y-g', 'x,y')
```

4 Summary

We have now covered some basic MATLAB commands and terminology. One of the best ways to learn MATLAB is to use it. Experiment with the software and see what else you can discover about this tool.

5 Practice Problems

1. Use the **help** command to learn about the command **isprime**.
   (a) What does the command **isprime** do?
   (b) Describe how to use the **isprime** command.
   (c) Are the following numbers prime?
      i. 233
      ii. 753
      iii. 809
      iv. 983
2. Which of the following would make good names for parameters and variables?
   (a) time
   (b) rat
   (c) delta 2
   (d) delta2
   (e) dog#3
   (f) 5days
3. Try and use MATLAB as you would a calculator and find the values for
   (a) $\frac{143}{45} - tan^2(\frac{2\pi}{3})$
   (b) $log_{10}2500$
   (c) $4 - 0.5ln300$
4. Define the following parameters: $a = -5 + i$, $b = \frac{37}{2}$, and $c = 3$. Find the values for:
   (a) a^c
   (b) $\sqrt{b - ac}$
   (c) change the value of parameter a to 5 and find $\sqrt{b - ac}$ (Hint: in MATLAB you can use the up arrow to work with some of your recently executed commands.)
5. Factor the polynomial $x^3 - 7x - 6$ using symbolic MATLAB and the **factor** command.
6. Solve the following system of equations for x, y and z.

$$\begin{aligned} 2x - y + z &= 4 \\ \frac{1}{2}x + 2y &= 8 \\ x + y - z &= 6 \end{aligned}$$