**Lab One**
**Computational Physics**
**Kigali Institute of Science and Technology**
December 2011

The purpose of this lab is for you to gain additional experience with matlab. The best way to learn matlab is to try it out, experiment, and to keep practicing. If you already have some experience with matlab, please help other students who are just starting out.

There is nothing to hand in from today's lab. The goal is to explore and experiment. But save your work from this week. Next week you will need to do work that leads to a small report.

**Using the Matlab Editor.** Open the matlab editor and create a new file. I will show you how to do so, or you can ask another student. Make sure you know how to save the file. At the end of the lab, I would suggest emailing the matlab file to yourself, and/or saving it on a flash drive.

**Comments.** A *comment* in matlab file begins with a `%`. Anything that comes after the `%` is ignored by matlab. Comment are for you, not the computer. Put comments in your matlab code so that you can understand what it does. Always put comments on the top of matlab file with your name(s), the date, and a brief description of the code. Always use comments when you define a new variable. E.g.

```
m = 0.3 %mass of electron, measured in nm
```

**Review of Simple Plotting:** The plot command is of the form `plot(xdata, ydata)`. The `xdata` needs to be in the form of a list. The `start;step-size;end` command makes a list. Suppose you want to plot $\sin(x)$ from $x = 0$ to $x = 10$. Try the following and observe what happens.

```
x1 = 0:2:10
plot(x1,sin(x1))
```

Then try:

```
x2 = 0:1:10
plot(x2,sin(x2))
```

And now try:

```
x3 = 0:0.2:10
plot(x3,sin(x3))
```

So when making a plot you need to choose a step size that makes the plot look sufficiently smooth. How small a step size you need will depend on how wiggly or steep the function is.

**Labeling axes:** You can label axes for a plot as follows.

```
t = 0:0.01:10;
N = 4*exp(-0.5*t);
plot(t,N)
xlabel('time in hours')
ylabel('number of atoms')
title('radioactive decay')
```

**Multiple plots on the same axes:** Often we want to plot two functions on the same axes so we can compare them. Here is an example of how to do this:

```
x = 0:0.01:2*pi;
y1 = sin(x)
y2 = cos(x)
plot(x,y1,x,y2)
```

**Defining a Function:** In order to use matlab's built-in function solver, we need to learn how to define our own functions. Here is an example to illustrate this:

```
f = inline(x^2 - 9)
```

We can now evaluate the function at any $x$. For example, try typing `f(4)`. Then try `f(5)`.

**Using `fzero` to find roots:** Now that we have the function defined, we can use matlab's built-in command for finding the roots (also called zeros) of a function. Try the following:

```
fzero(f,2.5)
```

The 2.5 in the above command tell matlab where to start looking for the zero. Try other values. Can you have matlab find the other root?

**Exercise:** Make matlab find the root of the equation $g(z) = \cos(z) - z$. Confirm that the number you get makes sense by plotting $\cos(z) and (z)$, as follows: `plot(x,cos(x),x,x)`.

**Exercise:** Define a function $d = \frac{1}{2}at^2$. Set $a = 9.8$. Then use matlab to evaluate $d(1)$, $d(2)$, and $d(3)$. Use comments when you define the function.

**Some evaluation practice:** Have matlab evaluate the following expressions:

1. $3\log(3)$
2. $(\frac{1}{7})^{5/7}$
3. $\sqrt{5}$

**Precision and Infinities:** Have matlab evaluate the following expressions:

1. $\tan(0)$
2. $\tan(\pi/4)$
3. $\tan(\pi/2)$
4. $\tan(\pi)$
5. $\tan(3\pi/2)$
6. $\tan(2\pi)$

Are any of the results surprising?