PHYS 1002—Lab 3
Data Analysis with MATLAB:
The simple pendulum

Kelsey Jordahl

February 10, 2005 (revised)

Objective

We are going to use MATLAB to analyze data taken in a simple pendulum experiment. We will present results graphically as well as quantitatively. Error analysis will show some of the limitations in the data, and how to get around them.

1 Introduction to MATLAB

MATLAB is a powerful data analysis package available on FDU lab computers. Many scientists and engineers prefer MATLAB to spreadsheet programs (such as Microsoft Excel) for scientific computing. MATLAB has many built-in analysis and graphics tools, and is easy to extend and program with customized functions.

First, start up MATLAB (Start menu→All Programs→MATLAB 6.5→MATLAB 6.5). The basic interface of MATLAB is the command prompt. Individual operations are performed on the prompt. When you start up MATLAB, you will be given a prompt like this:

```
  To get started, select "MATLAB Help" from the Help menu.

>>
```

MATLAB can represent numbers as well as more complicated data structures. We will be using vectors, which to MATLAB are list of numbers. MATLAB can also deal with matrices (rectangular arrays of numbers) but we won't be needing them for this lab.

We can set a variable:

```
>> x=1
```

If we end a line with a semicolon, it will not print the result:

```
>> x=3;
```

However, the command was still executed and we can see that the value of `x` has changed:

```
>> x

x =
    3
```

Now try a vector list:

```
>> y=[1 2 3 4 5]
```

You can do the same thing in shorthand with:

```
>> y=1:5
```

MATLAB can perform mathematical operations on vectors as well as scalars (simple numbers)

```
>> x*3

ans =
     9

>> y*3

ans =
     3    6    9    12    15
```

There are many built-in functions that can perform more complex operations. Try some of these:

```
>> max(y)
>> min(y)
>> mean(y)
>> sqrt(x)
>> x*y+3
```

You can save your entire workspace for later (make sure you are in a directory for which you have write permission):

```
>> who
>> save junk
>> clear
>> who
>> load junk
>> who
```

MATLAB also has very powerful graphics capabilities:

```
>> x=-10:10;
>> y=x.^2;
>> plot(x,y)
```

and you can plot more than one thing at once:

```
>> y2=(x.^3)/10;
>> plot(x,y,x,y2)
```

Finally, for a taste of what MATLAB is really capable of, type

```
>> demo
```

and try a few of the demos (especially under "Graphics").

2 Data analysis of the simple pendulum

We will use data taken from a simple pendulum experiment with an ultrasonic range meter which measured displacement.

First, change to the directory where the experimental dataset is located (I'll show you how to do this). Next, load the data and see what's there:

```
>> clear
>> load data
>> whos
```

Plot the position vs. time. Label the plot and the axes.

```
>> plot(t,x)
>> xlabel('Time (seconds)')
>> ylabel('Displacement (meters)')
>> title('Simple pendulum data')
```

Find the velocity of the pendulum.

```
>> dt=0.025;
>> v=diff(x)./0.025;
>> plot(t,v)
```

Why doesn't this work? Try this:

```
>> v=interp1(t(1:599)+0.025,diff(x)./0.05,t);
>> plot(t,v)
```

Now plot x versus v. Describe how the motion of the pendulum changes (both the position and velocity) through each quadrant of the graph. Is it what you expected?

How many full cycles does it make?

Now try to find and plot the acceleration.

Next we will find the length of pendulum L and an equation for its motion. Remember that the period is given by

$$T = 2\pi\sqrt{\frac{L}{g}} \tag{1}$$

Start by finding the zero crossings of the displacement:

```
>> z=findzeros(t,x)
>> plot(t,x,z,0);
```

How can we find the period from these points?

Plot theoretical and experimental curves for the motion of the pendulum. For this you will need to set the values of x_0, f and ϕ from the experimental data and use it in the equation

$$x(t) = x_0 \cos(2\pi f t + \phi) \tag{2}$$

Translated into MATLAB, we can plot our theoretical curve with

```
>> xt=x0*cos(2*pi*f*t+phi);
>> plot(t,x,t,xt)
```

How do we find the values for x_0, f and ϕ?

When you are satisfied that you have a good fit, go on to the next section.

3 Error analysis

Calculate mean error.

```
>> err1=mean(x-xt)
```

Why is this not a good estimate of error?

Calculate mean absolute error.

```
>> err2=mean(abs(x-xt))
```

Calculate RMS error.

```
>> err3=sqrt(mean((x-xt).^2))
```

How does this differ from mean absolute error? Why?

The usual way to find the best parameters to fit a theoretical model (such as equation [2]) to experimental data is to minimize the error, most often the RMS error. Did your estimate of the parameters x_0, f and ϕ provide the lowest possible error? Did any of your classmates find parameters that gave a lower error? Can you find parameters to reduce the error?