Matlab Lab 1: Biorhythms and Lines

Before we get started today: Printing Script Files

Copy the following script file into the Matlab editor, and save it as `temp1.m`:

```
%% Sample Script File

%% Exercise 1
% This is a comment for Exercise 1
% You can include LaTeX in your comments: $\int_a^b f(x)\,dx$
x=[1 2 3 4 5];
y=[6 7 8 9 10];
x-y

%% Exercise 2
% The following image comes with Matlab (for testing)
load durer
image(X)
colormap(map)
axis image
```

In the editor, choose `File`, then `Publish temp1.m`. A browser window should now open with the script file and the Matlab output. Choose `Print`, then select `Print to file`, and then print. You should see a dialog box come up to enter the file name (the suffix should be `.ps`). This is the file that you will upload to CLEo.

Biorhythms[1]

Biorhythms were very popular in the 1960s. You can still find many websites today that offer to prepare personalized biorhythms, or that sell software to compute them. Biorhythms are based on the notion that three sinusoidal cycles influence our lives. The physical cycle has a period of 23 days, the emotional cycle has a period of 28 days, and the intellectual cycle has a period of 33 days. For any individual, the cycles are initialized at birth, so sine functions that start on that day are used for our model. We will also assume that the amplitude for each is 100 units.

1. Assume that t is measured in days since some fixed date in the past. Let t^* be the day (not the date) that you were born. Write the three sine functions that would represent the "Physical", "Emotional", and "Intellectual" cycles for you since you were born.

   Hint: The sine curve $A\sin(wt)$ has amplitude A and period $2\pi/w$.

[1] From Cleve Moler's "Introduction to Matlab"

2. Here are some special Matlab commands:

`datenum('01-Jan-2012')`	Gives the number of days elapsed to Jan 01, 2012.
`fix(now)`	Gives the number of days elapsed to today.
`datestr(754963)`	Converts 754,963 days elapsed to a calendar date.

   (a) How many days have you been alive? (Use Matlab, and assume that "today" is Friday, April 6th.

   (b) Compute the date it will be when you will have been alive for 20,000 days.

3. Write a script file that will plot your biorhythms for an eight week period centered around today's date. Use `legend` to label the three functions, and `datetick` so that the $x-$axis labels dates with calendar-style (I want you to use the help files for these). Here's a code fragment you can look at for assistance:

```
mybirthday='03-Mar-1963';
t0=datenum(mybirthday);
t1=fix(now);
t=(t1-28):1:(t1+28);
% These are not the correct sine functions- Use the ones you found above.
y=[ sin(2*(t-t0)/10)
    sin(3*(t-t0)/20)
    sin(4*(t-t0)/30) ];
plot(t,y);
title('My Biorhythms!');
```

4. All three cycles start at zero when you are born. How long does it take to return to that initial condition? How old were you, or will you be, on that date? Re-plot your biorhythms using the previous script. (Hint: You might find `lcm` helpful)

5. Is it possible for all three cycles to reach their maximum or minimum at exactly the same time?

What to turn in? Make a script file that answers 2, 3, and 4 (your answer to 1 will be in your solution to 3). Save it to a script file, then publish and print to file (then upload that file to CLEo).

Line of Best Fit

This next section is a bit different than the last- Here, we take a look at some data.

Our main problem: Given a set of N data points, (x_i, y_i), can we determine the "best fitting" line through those points? We can write this question as:

Find m, b so that $y = mx + b$ is the best model function for the given data.

What is "best"?

The word "best" implies that there is some measure of the error which we will minimize. What error is there? We are assuming that the data does not exactly fit the line- otherwise, we can simply use two data points and find the equation of the line.

One way to look at the problem is as a system of equations. That is, find m, b so that:

$$\begin{aligned} mx_1 + b &= y_1 \\ mx_2 + b &= y_2 \\ \vdots & \vdots \\ mx_N + b &= y_N \end{aligned}$$

The problem is that there is no solution, so we introduce an error at each data point. Let $\hat{y}_i$ be the $y-$value on the line:

$$\hat{y}_i \doteq mx_i + b$$

and y_i be the $y-$value given in the data. We define the **sum of squares error** (SSE) as a function of m, b:

$$E = (y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \cdots + (y_N - \hat{y}_N)^2 = \sum_{j=1}^{N} (y_i - \hat{y}_i)^2$$

so that

$$E(m, b) = \sum_{j=1}^{N} (y_i - (mx_i + b))^2$$

Exercises

1. By hand, show that, by setting the two partial derivatives to zero:

$$\frac{\partial E}{\partial m} = 0 \qquad\qquad \frac{\partial E}{\partial b} = 0$$

we get the following system of equations in m, b:

$$m \sum_{j=1}^{N} x_j^2 + b \sum_{j=1}^{N} x_j = \sum_{j=1}^{N} x_j y_j$$

$$m \sum_{j=1}^{N} x_j + bN = \sum_{j=1}^{N} y_j$$

2. Cramer's Rule is a quick way to solve systems of equations by hand. If we define the determinant:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

and we have the system

$$\begin{aligned} ax + by &= e \\ cx + dy &= f \end{aligned}$$

then the solution is:

$$x = \frac{\begin{vmatrix} e & b \\ f & d \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} \qquad y = \frac{\begin{vmatrix} a & e \\ c & f \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}}$$

By hand, use Cramer's Rule and the system of equations we obtained in the previous exercise to find the line of best fit (that is, find m, b for the line) using the following data:

x	-1	0	1	2
y	-4	1	3	5

Hint: You might compute these values first:

$$\sum_{j=1}^{4} x_i^2 = ?? \qquad \sum_{j=1}^{4} x_i = ?? \qquad \sum_{j=1}^{4} x_i y_i = ?? \qquad \sum_{j=1}^{N} y_i = ??$$

NOTE: The line you have found is sometimes called the least squares line, since we minimized the sum of squares error to get it.

3. Verify your answer using Matlab's plotting tools, and to give the y value for $x = 3$ (and plot the results):

   First, plot the data:

   ```
   x=[-1 0 1 2];
   y=[-4 1 3 5];
   plot(x,y,'*');
   ```

   In the plot window, go to `Tools->Basic Fitting`, then select the linear fit. You might also check the "Show Equations" box to see the slope and intercept.

   Once we have the line, we can press the arrow key at the bottom of the fitting window until we see the dialog box to enter new values of x- There, put in 3. If you check the box, we can then see the new plot when you press "Evaluate".

4. In Matlab, we could have done the previous problem using the following script.

   ```
   x=[-1 0 1 2];
   y=[-4 1 3 5];
   a=sum(x.^2); b=sum(x); c=b; d=length(x); e=sum(x.*y); f=sum(y);
   A=[a b;c d];
   n=det([a b;c d]);
   ```

```
m=det([e b;f d])/n;
b=det([a e;c f])/n;
t=linspace(-1,2);
y1=m*t+b;
plot(x,y,'*',t,y1,'g-');
```

Exercise: Change the script into a function that inputs the vectors x and y, and outputs the numbers m and b. Make the first line of your page:

```
function [m,b]=Line01(x,y)
% function [m,b]=Line01(x,y)
%  Input data in vectors x and y
%  Output slope m and intercept b so that y=mx+b (using SSE)
```

5. In many biology problems, there is a scaling relationship of the type:

$$y = cx^n$$

   (a) By hand, show that taking the logarithm of both sides produces a linear function by taking $u = \ln(x)$ and $v = \ln(y)$.

   (b) On our class website is an M-file that gives body weight (in grams) and corresponding heart rates (in beats per minute) for mammals in a large range of weights (from a chipmunk to an elephant). Is there a relationship of the type $y = cx^n$?

   To get you started, download the file `Mammals.m`. Then in Matlab, type:

```
Mammals
u=log(w);
v=log(r);
plot(u,v,'*');
```

   Use the result of the previous exercise to find c, n so that $r = cw^n$. Use this result to find the predicted heart rate for someone that weighs 191 pounds (which is about the average adult male weight) and 164 pounds (about the average adult female weight). Are these numbers (your answers, not the weights) reasonable?

What to turn in? The answer to Question 1 and the first part of Question 5 can be on a piece of paper (turn it in the old fashioned way). Solutions for 2 and 3 need not be turned in, but you should do them before going on to 4. For question 4, upload your finished file Line01.m. For question 5, answer part (b) in a script file, then publish it and upload the result.