Biorhythms

Background

Biorhythm theory originated in Europe toward the end of the last century. According to this theory each person has a set of life rhythms or life cycles which begin on the day a person is born. There are three such rhythms known as the physical, emotional, and intellectual cycles. Each of these cycles takes the form of a sine wave with the following periods: physical cycle (23 days), emotional cycle (28 days), intellectual cycle (33 days)

In the case of the physical cycle, the theory proposes that in the first half, or positive portion, of the cycle, one's physical well-being is enhanced as opposed to the second half, or negative portion, of the physical cycle. Similar reasoning is applied to the emotional and intellectual cycles. The worst days are the critical days when any one of the three rhythms switches from positive to negative or vice versa.

Procedure

This is an individual project**. You are encouraged to discuss verbally the project with other students. You can brainstorm together solution approaches, and you can teach each other how to do things with Matlab. However, allowed collaboration ends with this verbal discussion. At no time can you copy work others have done, or have someone else do any of the work for you, or do any of the work for someone else.** Everything in the Matlab files themselves must be your work, and your work alone**. If you need more help, ask your proctor or instructor for assistance.**

Create a directory entitled **uuuuup5**, where **uuuuu** are the characters of your cougarnet username. As you work on this project, save all of the specified files in this directory. This should be set as your current working directory. When finished, you will turn in the complete directory using the dropfolders method.

Begin by creating a script file named **Main.**m. This file should begin with a "title" comment that includes your name, email address, and proctor number. This information should also be displayed in the command window when your script is run. Additional comments should briefly describe the project, the important parameters, and the required user-defined functions. The computations will be divided into the tasks described below. Preceding each task, provide the task number, e.g., "**Task 1**", followed by suitable explanatory comments. The task number and explanations should also be displayed in the command window when your script file is run. This documentation is required! In other words, both your script and your output must be easy to read with each section well delineated. Many of the tasks will require you to define functions, which will then be invoked by **Main**. Each of these functions must include appropriate comments, and must be saved in **uuuuup5**.

Be sure to put **clear, close all** and **clc** as the first executable instructions in your script.

On the course website, in the Projects/Project5 folder you will find a plain text file, zzp5data.txt. This file contains values that you will need to use in your script. Copy the file to your p5 directory. However, your solution must be kept sufficiently general so that if these values are changed the script will still execute correctly--without anybody making additional changes. We will execute your script with a different set of values! We will not provide any interactive input!

Task 1: (Read and display the data) Using the load function, load the data from the file zzp5data.txt into a matrix called **indata**. The data consists of a 3 column matrix of dates. The first column is the year, the second column is the month, and the third is the day. Display this in a neat table with appropriate title and column headings

Task 2:(Julian day number) The calculation of a person's biorhythm requires knowing the number of days since they were born. This may be determined by calculating the Julian day number of the date of birth and the Julian day number of the required day and subtracting them (required day minus birthday).
The Julian day number for a given day may be calculated by the following formula (valid for all dates AD):
`Julian day = 1721060 + 365y + 31(m - 1)+ d + [y'/4] - [y'/100] + [y'/400] - x` where y = year, m = month, d = day. The square brackets denote the greatest integer function (Matlab's floor function).
If $m \leq 2$, x = 0 and y' = y - 1 while for m>2, x = [.4m + 2.3] and y' = y.
Define a Matlab function **julian**(date) which will take a 3-column date matrix and return the corresponding Julian day vector. Evaluate your function for [2008,12,14], for [1808, 3,15], and for the matrix **indata,** displaying the results clearly labeled. Note: Do NOT use the Matlab datenum function!

Task 3:(cycle values) The biorhythm cycle value is $\sin(2\pi t/p)$ where t is the difference between the current Julian day number and the Julian day number of the person's birth and p is the period in days for the particular cycle—physical, emotional, or intellectual. Define a Matlab function **cycle(**jd,jb,p) which will calculate the appropriate cycle value on a particular day based on the current Julian day (jd), the Julian day of birth (jb) and the period (p). Design your function to handle vector input. Define a 3-component vector **cycleper** which will contain the number of days in each of the three cycles for ease of reference. Define **dumbirth** = [1978,9,10] and use **julian** and **cycle** to find the physical cycle value at [2008,9,20], the emotional cycle value at [2008,2,14], and the intellectual cycle value at [2008,11,23], displaying each of them, clearly labeled. Using the first row of **indata** as the birthdate, find all three cycle values at the date specified in the second row of **indata.**

Task 4:(cycle plots). Plot the biorhythm curves on the same graph for the first part of the input data, using the first row as the birthdate, the second row as the beginning date, and the third row as the ending date. Use red for the physical cycle, blue for the emotional cycle, and green for the intellectual cycle. Note that the horizontal axis values which correspond to Julian dates are quite messy. This should be Figure 1.

Task 5:(febdays). In order to convert back from a Julian day number to [year,month,day] form in the period we will be interested in, we will need to construct a 3-column matrix in which the first column gives the year, the second the month and the third the day for each consecutive day in our period of interest. This is very tedious because we cannot simply increment days. We have to watch for reaching the end of the month (in which case we must increment the month and set the day back to one), and the end of the year(in which case we must increment the year and set both the month and day back to one). This is further complicated by the fact that months have differing numbers of days, and because of leap year, the same month (February) can have either 28 or 29 days. So to begin with, define a Matlab function **febdays**(year) that will take a given year and return the number of days in February for that year. Recall that a year is a leap year if it is divisible by 4 and not divisible by 100 or if it is divisible by 400. Evaluate **febdays**(2008), **febdays**(1900), **febdays(**1600), **febdays**(1813), and febdays(indata(2,1)), in each case displaying the result clearly labeled.

Task 6:(datefill). Taking into consideration the facts mentioned in Task 5, and using the function **febdays**, define a Matlab function **datefill**(start,stop) that given a starting date (start) in 3-component form and an ending date (stop) in 3-component form will return a 3-column matrix showing the year, month, and day for the consecutive days from start to stop. Use an explicit loop for this—no array operations or functions. You will probably find it helpful to define a vector containing the number of days in each month, using the default of 28 for February
For example, **datefill**([1997,12,30],[1998,1,5]) should return
[1997,12,30;1997,12,31;1998,1,1;1998,1,2;1998,1,3;1998,1,4;1998,1,5]. . Evaluate **datefill(**[1592,6,23],[1592,7,19]), **datefill(**[2000,2,20],[2000,3,5]), and **datefill(indata**(5,:**)**,**indata**(6,:)). Display the results of each of these, clearly labeled.

Task 7:(biorhythm) You now have the building blocks necessary to create a six column matrix of dates and biorhythm cycle values. Therefore, define a Matlab function **biorhythm**(birth,beg,end) which will return this six column table for the person whose birthday is given in the 3-component vector birth, starting with the 3-component date beg and ending with the 3-component date end. The first column of the output table is the year, the second is the month, and the third is the day where the cycle values are computed. The fourth column is the value of the physical cycle, the fifth is the value of the emotional cycle, and the sixth is the value of the intellectual cycle. **biorhythm** will contain appropriate references to **julian**, **cycle,** and **datefill** defined above**.**
Evaluate **inbio** = biorhythm(indata(1,:),indata(2,:),indata(3,:)) and **mybio** = biorhythm(**mybirth**,[2008,11,26],[2009,1,1]) ; where **mybirth** is your own birthday. Display the output in neatly labeled form as illustrated below.

```
                 Biorhythms of Myself
Year-Month-Day    Physical Cycle    Emotional Cycle    Intellectual Cycle
2008-11-26
2008-11-27
  etc.
```

Task 8: (betterplot) Define the single column vectors **inph**,**inem**,**inint** consisting of the physical, emotional and intellectual cycle values for the first three rows of the input data (the 4-th,5-th, and 6-th columns of the table **inbio** constructed in Task 7 above.) Using plot with integers 1,2,3, etc for the horizontal axis, plot these biorhythm curves on the same graph. Use large red dots for the physical values, large blue dots for the emotional values, and large green dots for the intellectual values and label the plot appropriately. Repeat this for your own rhythms (from **mybio** above). Do not connect the dots. These should be Figure 2 and Figure 3, respectively.

Task 9:(critical days). To determine the critical days for a particular cycle we need to identify when the cycle changes from positive to negative and vice-versa. Therefore define a Matlab function **crit**(z) which will determine the location(s) in the vector z of sign changes between successive components. **crit** should return a vector giving the positions(subscripts) in z where sign changes are found. If 0 occurs in the list, its position should be included. If a sign change occurs between two positions, the position to go in the list should be the one corresponding to the smaller absolute value unless it has already been included , in which case use the other. In case of a tie in magnitude, return the first one. Thus crit([-5,-4,1,2,-3,-1,1]) should return [3,4,6] while crit([0,2,3,-1,3,-1,0,2]) should return [1,4,5,6,7]. Evaluate these two for your function **crit**. to be sure it is working properly,displaying the results. Then use **crit** to determine the critical days for all three cycles for the first input person and for you, based on **inbio** and **mybio**. Display these in a clearly labeled manner, eg.

```
 For the input person, the critical days for the physical cycle are
    1777-12-26
    1778-1-7
```

etc.

Turn In Your Project

Use the DropFolders method to turn in your project. Copy your entire uuuuup5 directory into the proper dropfolder. Remember that projects up to 24 hours late receive a 25 point penalty; projects later than 24 hours are not accepted.

No matter what problems you are having with your project, always turn in whatever you have completed by the due date. If you have strange last minute problems, you can send email to your professor explaining it. But don't miss the due date! Turn in whatever you have by the due date!!!