Problem Solving For Engineering Transfer

ENS 1300 - Spring 2020

Due Date	Lab 1: Intro to ENS 1300	Last Modified
3 February 2020		29 February 2020

Course Website

ENS 1300 lab assignments will follow material that can be found on the course website. You should read through the material for each lab carefully before proceeding with the associated assignment, as the instructions and/or examples provided in the assignment pdf documents will be minimal.

Deliverables

Each ENS 1300 lab assignment will have exactly **one** final deliverable. We will use a technique called zipping to package multiple lab files into a single file. You will complete all work for a lab in a single folder called `Lab##`, where the pound signs will be replaced by the lab number. Upon completion of each lab assignment: **(1)** be sure that all the deliverable(s) specified at the end of each problem are contained within the `Lab##` folder, **(2)** zip `Lab##`, name it `Lab##_LastnameFirstname.zip`, and **(3)** upload `Lab##_LastnameFirstname.zip` to the dropbox.

Directory Structure

Create a new directory (aka folder) called `ENS1300`. You will organize and store all files for this course in this directory. In order to avoid losing your work, it would be wise to back up this folder periodically. Create the subdirectories `Templates`, `Examples`, `Projects`, `Labs`, and `Labs/Lab01`. Your directory structure should now look like this:

```
ENS1300
|-- Examples
|-- Labs
|   |-- Lab01
|-- Projects
|-- Templates

5 directories, 0 files
```

Now open MATLAB, make `ENS1300/Templates` the working directory, and then create a new script with the following contents:

```
% Name:
% Section:
% Assignment:
% Description:
% Date:
clear; close all; clc; format compact; format short;

%% Initialize

%% Read

%% Execute

%% Print
```

Lines 1-4 will be referred to as the header section of ENS 1300 MATLAB script files. Input your name and section on lines 1 and 2, respectively. Save the script as `ENS1300/Templates/script.m`. If student Foo Bar were in section 2, their script template file for ENS 1300 would look like this:

ENS1300/Templates/script.m

```
% Name: Foo Bar
% Section: 2
% Assignment:
% Description:
% Date:
clear; close all; clc; format compact; format short;

%% Initialize

%% Read

%% Execute

%% Print
```

Now you have a starting point for each script that you will submit for this class. Zip your entire `ENS1300` directory and name it `ENS1300_LastnameFirstname.zip`. Move the zip file to `ENS1300/Labs/Lab01`. Foo Bar's directory structure would now look like this:

```
ENS1300
|-- Examples
|-- Labs
|   |-- Lab01
|       |-- ENS1300_BarFoo.zip
|-- Projects
|-- Templates
    |-- script.m

5 directories, 2 files
```

Deliverable(s): `ENS1300_LastnameFirstname.zip`

Graded Item	Point Value
Zip file named as specified	5
All files and directories present	20
Total	25

Problem 1.1

Before beginning: Make a copy of `ENS1300/Templates/script.m` and save it as `ENS1300/Labs/Lab01/springProb.m`. Set the MATLAB working directory to `ENS1300/Labs/Lab01`.

A set of springs have the given loads and stiffnesses shown in Table 1 (data also available in csv format here). Copy and paste the data set into the `ENS1300/Labs/Lab01/springProb.m` script and define it as a matrix called `data`. By using indexing, separate the matrix into column vectors containing spring forces and stiffnesses. Name the two column vectors `springForce` and `springStiffness`, respectively. Using the specified variable names where applicable, complete the following tasks:

	Task	Variable Name(s)
(a)	Fill in the header section.	
(b)	Find the total number of springs.	`numSprings`
(c)	Calculate the deflection of the 15^{th} spring.	`springDeflection`
(d)	Calculate the total potential energy in the springs.	`springEnergy`
(e)	Calculate the average stiffness.	`meanStiffness`
(f)	Calculate the median deflection.	`medianDeflection`

Spring equations:

$$F = kx$$

$$U = \frac{1}{2}kx^2$$

$F =$ Force (N)

$U =$ Potential energy (J)

$k =$ Spring stiffness $\left(\frac{\text{N}}{\text{m}}\right)$

$x =$ Spring deflection (m)

Example: The header in Foo Bar's `ENS1300/Labs/Lab01/springProb.m` file might look something like this:

ENS1300/Labs/Lab01/springProb.m

```
% Name: Foo Bar
% Section: 2
% Assignment: Lab 1, Problem 1
% Description: Simple spring problem
% Date: 1 January 1983
```

Deliverable(s): `springProb.m`

Graded Item	Point Value
File appropriatly named	3
Adequate script header	3
Use of specified variable names	4
Correct calculations (b)-(f)	15
Total	25

Table 1: Data for Problem 1.1

Spring Force (N)	Stiffness (N/m)
23	349
19	284
34	325
38	259
40	230
21	359
18	312
32	343
40	353
18	270
30	322
15	272
17	295
23	327
18	283
23	236
33	311
20	317
33	358
29	344
15	206
21	354
24	320
24	217
30	237
28	257
37	214
38	258
25	337
22	327
19	287
33	327
27	252
39	253
31	227
26	282
19	252

Problem 1.2

The data in Table 2 gives segments of some traveled path. Make another copy of the script template and save it as `ENS1300/Labs/Lab01/pathProb.m`. In this new script, complete the following tasks:

	Task	Variable Name(s)
(a)	Fill in the header section.	
(b)	Extract the values from Table 2 and define them as a matrix.	`data`
(c)	Using indexing, separate `data` into two column vectors; one for distance and the other for angle.	`distance`, `angle`
(d)	Calculate the horizontal and vertical components for each segment of the path.	`xDist`, `yDist`
(e)	Sum the horizontal components.	`xTot`
(f)	Sum the vertical components.	`yTot`
(g)	Calculate the magnitude of the resultant distance.	`resMag`
(h)	Calculate the angle (CCW from +x) of the resultant distance vector.	`resAng`

Table 2: Data for Problem 1.2

Distance (ft)	Angle (CCW from +x)
19	281
33	274
30	233
34	162
27	204

Deliverable(s): `pathProb.m`

Graded Item	Point Value
File named as specified	3
Adequate script header	3
Use of specified variable names	4
Correct calculations (c)-(h)	15
Total	25

Problem 1.3

Make another copy of the script template and save it as `ENS1300/Labs/Lab01/equationProb.m`.

Consider the following equation:

$$A = \frac{e^{-c/(2x)}}{\ln(y)\sqrt{zd}}$$

Where:

$c = 2$
$d = 4$
$x =$ a row vector of values from 1 to 10 in increments of 1
$y =$ a row vector of values from 0.1 to 0.55 in increments of 0.05
$z =$ a row vector of 10 equally spaced values in the range 18 to 45, inclusive.

Write a script that will:

	Task	Variable Name(s)
(a)	Correctly initialize given variables.	`c`, `d`, `x`, `y`
(b)	Calculate A.	`A`
(c)	Calculate average value of A.	`meanA`

Deliverable(s): `equationProb.m`

Graded Item	Point Value
File named as specified	3
Adequate script header	3
Use of specified variable names	4
Correct (a)-(c)	15
Total	25

Submission Details

If your name were Foo Bar, your directory structure should now look like this:

```
ENS1300
|-- Examples
|-- Labs
|   |-- Lab01
|       |-- ENS1300_BarFoo.zip
|       |-- equationProb.m
|       |-- pathProb.m
|       |-- springProb.m
|-- Projects
|-- Templates
    |-- script.m

5 directories, 5 files
```

Now zip your `Lab01` directory and name it `Lab01_LastnameFirstname.zip`. This zip file is what you will upload to the dropbox to be graded. You are free to store this file anywhere you like, but it is recommended to stay organized and consistent. Here are a few good options:

Recommended: Store your zip files in `ENS1300/Submissions`.

```
ENS1300
|-- Examples
|-- Labs
|   |-- Lab01
|       |-- ENS1300_BarFoo.zip
|       |-- equationProb.m
|       |-- pathProb.m
|       |-- springProb.m
|-- Projects
|-- Submissions
|   |-- Lab01_BarFoo.zip
|-- Templates
    |-- script.m

6 directories, 6 files
```

Store your zip files in `ENS1300/Labs`.

```
ENS1300
|-- Examples
|-- Labs
|   |-- Lab01
|   |   |-- ENS1300_BarFoo.zip
|   |   |-- equationProb.m
|   |   |-- pathProb.m
|   |   |-- springProb.m
|   |-- Lab01_BarFoo.zip
|-- Projects
|-- Templates
    |-- script.m

5 directories, 6 files
```