Problem Solving For Engineering Transfer

ENS 1300 - Spring 2020

Due Date	Lab 3: Subfunctions, Data Types, Data Import/Export	Last Modified
17 Febuary 2020		16 February 2020

Important

Create a new directory for this lab, `ENS1300/Labs/Lab03`. Set this as your working directory in MATLAB before beginning. Upon completion: (1) be sure that all the deliverable(s) specified at the end of each problem are contained within the `Lab03` folder, (2) zip `Lab03`, name it `Lab03_LastnameFirstname.zip`, and (3) upload `Lab03_LastnameFirstname.zip` to the brightspace dropbox.

Main deliverable: `Lab03_LastnameFirstname.zip`

Graded Item	Point Value
Zip file follows specified naming convention	5
Zip file contains all required files and directories	5
Total	10

Problem 1

You are a design engineer working for company XYZ performing experiments on products that may be susceptible to vibration damage. Your measurement equipment records high resolution position values and outputs them in csv format to your labratory server, which can be accessed at the following web address: https://c-a-j.github.io/ENS1300/. As an exemplary employee concerned with your efficiency on the job, you wish to bypass the time-consuming process of visiting the server website, downloading individual csv files, opening them, and copying/pasting the values into MATLAB.

Your most recent output file is called `harmonicData.csv`, so the full URL path to it is `https://c-a-j.github.io/ENS1300/harmonicData.csv`. The first column is time in seconds and the second column is position in millimeters. Write a function called `plotProb` that accepts two inputs and returns no output. The first input should be a URL (the web address to the csv file) and the second should be an output filename. The function will use the URL to import data, generate an appropriately annotated plot but **not** display it in a figure window, and save the plot as a png file using the output filename.

Example: The following code segment creates a figure without displaying it, plots $y = x^3$, and saves the plot as a png image called `filename.png`.

```
x = -5:0.1:5;
y = x.^3;

fig = figure('visible','off');
plot(x,y)
saveas(fig,'filename','png')
```

Function Specifications:

Function Name	Purpose	Input(s)	Output(s)
`plotProb`	1: import data 2: plot data 3: save plot as png image	1: CSV file URL 2: image filename	None

Useful Built-in Functions: websave(), csvread(), dlmread(), importdata(), figure(), saveas()

Deliverable(s): `plotProb.m`

Graded Item	Point Value
Adequate comments and organization	5
Adhearing to file and function name specifications	5
Correct data import	5
Correct function input/output	5
Total	20

Problem 2

Make a copy of the `calcGrade.m` function file from Lab 2 and name it `calcGrade2.m`. Modify the function to accept a single structure input and a single structure output. The output structure should contain all the original fields as the input plus an additional field, `avgGrade`. Assume the input structure contains the following fields: `exams`, `labs`, `quizzes`, and `projects`. Calculate the average grade just as in Lab 2 and assign the value to a field called `avgGrade`.

Function Specifications:

Function Name	Purpose	Input(s)	Output(s)
`calcGrade2`	calculate average grade	structure with 4 fields (`exams, labs, quizzes, projects`)	structure with 5 fields (`exams, labs, quizzes, projects, avgGrade`)

Write a script called `calcGrade2_test.m` that imports student grade data from a JSON file at the url https://c-a-j.github.io/ENS1300/grades.json and calculates the average for one student using the calcGrade2function. The JSON file contains 500 students' grades. They will be imported as a structure using the built-in `webread()` function. (NOTE: Octave users may need to seek an alternate method.)

Example: The following lines will result in a structure called `students` that contains 500 structures. Each of the 500 student structures contains the fields `exams`, `labs`, `quizzes`, and `projects`.

Command Window

```
>> url = 'https://c-a-j.github.io/ENS1300/grades.json';
>> students = webread(url);
>>
```

From this point, you just need to extract a single student from the main structure and use it as input to the `calcGrade2()` function.

Command Window

```
>> url = 'https://c-a-j.github.io/ENS1300/grades.json';
>> students = webread(url);
>> students.student314 = calcGrade2(students.student314);
>>
```

In the command window block above, the structure `students.student314` is overwritten by the output of `calcGrade2`.

Useful Built-in Functions: webread()

Deliverable(s): `calcGrade2.m`, `calcGrade2_test.m`

Graded Item	Point Value
Adequate comments and organization	5
Adhearing to file and function name specifications	5
Correct data import	5
Correct function input/output	5
Total	20

Problem 3

Write a main function called `cart2sphere_uno()` and subfunctions `getInput()`, `convertCoords()`, and `printOutput()` that obtains a 1x3 vector of 3-D cartesian coordinates from the user via the built-in `input()` function, converts the coordinates to spherical, prints the results to the command window, and outputs the spherical coordinates as a 1x3 vector (report θ and ϕ in degrees).

Function Specifications:

Function Name	Purpose	Input(s)	Output(s)
`cart2sphere_uno`	call subfunctions	None	1: vector [r, θ, ϕ]
`getInput`	get 3-D cartesian point coordinates from user	None	1: vector [x, y, z]
`convertCoords`	convert cartesian coordinates to spherical	1: vector [x, y, z]	1: vector [r, θ, ϕ]
`printOutput`	print results to command window	1: vector [r, θ, ϕ]	None

Equations:

$r = \sqrt{x^2 + y^2 + z^2}$ = radius

$\theta = \cos^{-1}\left(\frac{z}{r}\right)$ = inclination angle

$\phi = \tan^{-1}\left(\frac{y}{x}\right)$ = azimuth angle

Example: The usage and output of your main function should be the same as follows when the user enters [3, 4, 5].

Command Window

```
>> spherical = cart2sphere_uno
Enter the cartesian coordinates of a point in
vector format [x y z]: [3,4,5]

The radius is 7.07
The inclination angle is 45.00
The azimuth angle is 53.13

spherical =
    7.0711   45.0000   53.1301
>>
```

Useful Built-in Functions: input(), fprintf()

Deliverable(s): `cart2sphere_uno.m`

Graded Item	Point Value
Adequate comments and organization	5
Adhearing to file and function name specifications	5
Adhearing to function input/output specifications	5
Correct resulting spherical coordinate matrix	10
Total	25

Problem 4

In the previous problem, a single 3-D cartesian coordinate needed to be manually typed in by the user each time the main function was called. This is acceptable if just a few conversions are needed, but imagine how time consuming that process would be for hundreds, or even thousands, of coordinates. In this problem, we are going to import and export any number of coordinates using csv (comma separated value) files.

Make a copy of the `cart2sphere_uno` function file and name it `cart2sphere`. In the new file, rename the main function to `cart2sphere` so that it matches the filename. Modify all the functions to meet the following specifications.

Function Specifications:

Function Name	Purpose	Input(s)	Output(s)
`cart2sphere`	call subfunctions	1: input filename (string) 2: output filename (string)	1: spherical coordinates (matrix)
`readInput`	read 3-D cartesian point coordinates from csv file	1: input filename (string)	1: cartesian coordinates (matrix)
`convertCoords`	convert cartesian coordinates to spherical	1: cartesian coordinates (matrix)	1: spherical coordinates (matrix)
`printOutput`	print results to command window and file	1: spherical coordinates (matrix) 2: output filename (string)	None

Equations:

$r = \sqrt{x^2 + y^2 + z^2}$ = radius

$\theta = \cos^{-1}\left(\frac{z}{r}\right)$ = inclination angle

$\phi = \tan^{-1}\left(\frac{y}{x}\right)$ = azimuth angle

Example: The usage and output of your main function should be the same as follows when this csv file is used as the input.

Command Window

```
>> ifile = 'cartesianCoordinates.csv';
>> ofile = 'sphericalCoordinates.csv';
>> sphereCoords = cart2sphere(ifile, ofile);

The average radius is 97.08
The average inclination angle is 55.31
The average azimuth angle is 45.98

Result file, sphericalCoordinates.csv, written successfully.
>>
```

The output messages in the example above are provided as a means to check your solution. The spherical coordinate matrix output from the function `cart2sphere()` is what will be checked for accuracy.

Useful Built-in Functions: websave(), csvread(), dlmread(), importdata(), fprintf()

Deliverable(s): `cart2sphere.m`

Graded Item	Point Value
Adequate comments and organization	5
Adhearing to file and function name specifications	5
Adhearing to function input/output specifications	5
Correct resulting spherical coordinate matrix	5
Correct output file written	5
Total	25