ENG1060 ASSIGNMENT – S2 2016

Due: 11:00PM (Sharp), Friday 21st October 2016

This assignment should be completed INDIVIDUALLY. Plagiarism will result in a mark of zero. Plagiarism includes letting others copy your work and using code without citing the source. If a part of your code is written in collaboration with classmates, say so in your comments and clearly state the contributions of each person.

NOTE: Your MATLAB code will be checked for plagiarism – DON'T RISK LOSING ALL 10 MARKS BY COPYING SOMEONE ELSE'S CODE (OR BY ALLOWING SOMEONE ELSE TO COPY YOURS)

INSTRUCTIONS

Download **`template.zip`** from Moodle and update the M-Files named **`Q1a.m`**, **`Q1b.m`** etc with your assignment code. DO NOT rename the M-Files in the template or modify **`run_all.m`**. Check your solutions to Q1 and Q2 by running **`run_all.m`** and ensuring all questions are answered as required.

SUBMITTING YOUR ASSIGNMENT

Submit your assignment online using Moodle. You must include the following attachments:

1) A **ZIP** file (**NOT .rar or any other format**) named after your student ID containing the following:
    a. Solution M-Files for assignment tasks named: **run_all, `Q1a.m`, `Q1b.m`** etc…
    b. Any additional function files required by your M-Files (**PowerFit.m** function etc.)
    c. All data files needed to run the code, **including the input data provided to you**

    Follow "**ENG1060 Assignment ZIP file instructions.pdf**" to prepare your zip file for submission.
2) An assignment cover sheet with your **name** and **ID** (do **NOT** include this in the zip file)

Your assignment will be marked in your usual computer lab session during Week 12. ***YOU MUST BE IN ATTENDANCE TO HAVE IT MARKED***. We will extract (unzip) your ZIP file and mark you based on the output of **`run_all.m`**. It is your responsibility to ensure that everything needed to run your solution is included in your ZIP file (including data files). It is also your responsibility to ensure that everything runs seamlessly on the (Windows-based) lab computers (especially if you have used MATLAB on a Mac OS or Linux system).

MARKING SCHEME

This assignment is worth 10% (1 Mark == 1%). Code will be graded using the following criteria:

1) **`run_all.m`** produces results **automatically** (no additional user interaction needed except where asked explicitly)
2) Your code produces correct results (printed values, plots, etc…) and is well written.

ASSIGNMENT HELP

1) You can ask questions in the Discussion Board on Moodle
2) Hints and additional instructions are provided as comments in the assignment template M-Files
3) Hints may also be provided during lectures
4) The questions have been split into sub-questions. It is important to understand how each sub-question contributes to the whole, but each sub-question is effectively a stand-alone task that does part of the problem. Each can be tackled individually.
5) I recommend you break down each sub-question into smaller parts too, and figure out what needs to be done step-by-step. Then you can begin to put things together again to complete the whole.
6) To make it clear what must be provided as part of the solution, I have used bold italics and a statement that (usually) starts with a verb (e.g. ***Write a function ..., Print the value...***, etc.)

Question 1 starts on the next page.

QUESTION 1 [7 MARKS]

Background

You have been asked to manage a project to install a pipeline from an offshore gas platform to an onshore gas processing plant, and to find the cheapest design that is possible. The platform is D km offshore, and L km along the coastline from the processing plant. The pipeline will follow a straight line from the platform to the shore, and hit land at a point that is a distance 'x' from that point on the shore that is closest to the platform. We assume that the coastline is a straight line. The layout is shown schematically in Fig 1.

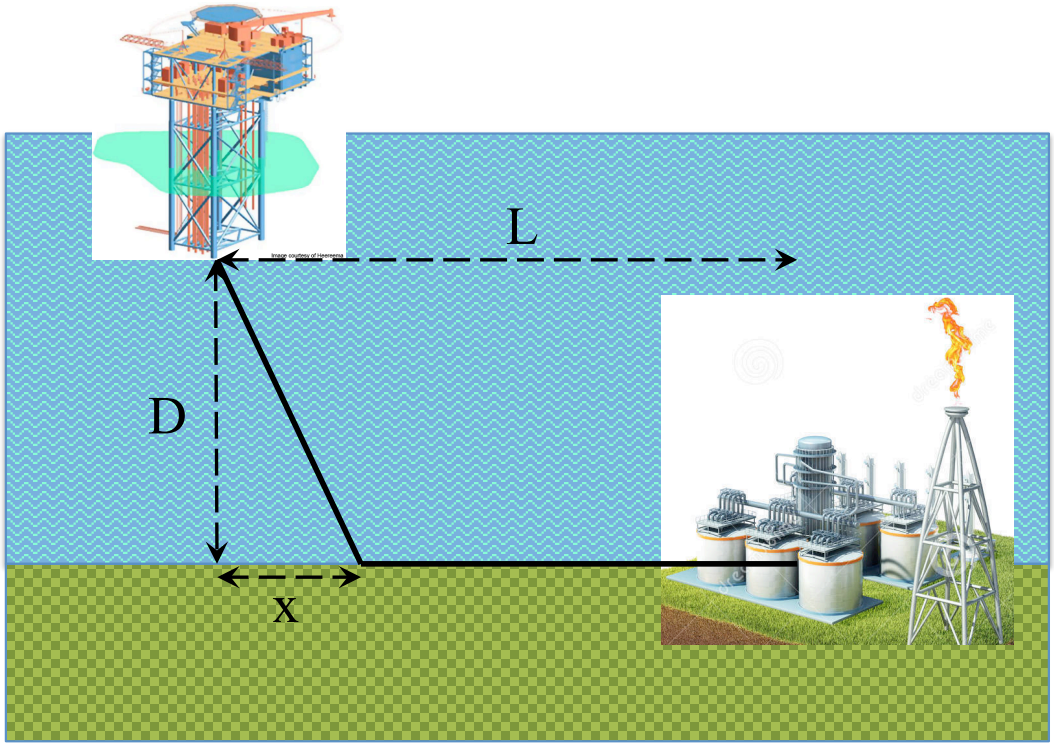

Fig 1 A schematic of the gas pipeline from an offshore platform to an onshore processing facility.

Q1a

One company has quoted the cost of laying a subsea pipeline of $\$C_{S1}$ per km (regardless of water depth) and the cost of laying onshore pipeline at $\$C_{O1}$ per km. ***Write a MATLAB function*** called **PipeCost1** that calculates the total cost of laying the pipeline as a function of 'x', the distances L and D and the costs per kilometer $\$C_{S1}$ and $\$C_{O1}$. **Note:** write the function so that 'x' can be a vector – assume all other input parameters are scalars.

If D = 40 km, L = 120 km, C_{S1} = \$2.5M per kilometer and C_{O1} = \$2.0M per kilometer, ***plot the cost*** of the pipeline (in units of \$100M) for 0 <= x <= 120km for 121 equally spaced points.

Q1b

Write a MATLAB function called **dCost1dx** that calculates the derivative of the cost with respect to the distance x (hard code this in MATLAB, do not attempt to use the symbolic toolbox). On a new figure, ***plot the derivative*** of cost w.r.t. x for 0 <= x <= 120km for 121 equally spaced points. Assume the same parameters from Q1a.

Use the Newton-Raphson method ***to determine the distance*** 'x' that gives the lowest cost pipeline and ***print the lowest cost*** (in $M) to the MATLAB command window to 3 significant figures with a suitable statement attached (i.e. ***DO NOT*** just write the cost). In your fprintf statements, add units for distance and add new lines for each printout.

(***NOTES:***

1. You should write your own Newton-Raphson code.
2. You should use a starting guess in N-R of 0.0
3. You should use an absolute precision of 0.001 (i.e., not a % precision)
4. You ***CANNOT*** easily pass a function that has multiple input parameters into another function as an input parameter. However, if you know all of the input values except x (i.e. you know D, L, C_{S1}, C_{O1}) you can define an anonymous function that you can pass as an input parameter
   As an example, if x is a vector of values, and MyFunc a function with 4 parameters (x,A,B,C) then you ***cannot*** do the following
   ```
   plot(x,MyFunc(x,A,B,C))
   ```
   However, if you set values for A, B, C, then you ***CAN*** do this
   ```
   A=val1
   B=val2;
   C=val3;
   f=@(x) MyFunc(x,A,B,C)
   plot(x,f(x))
   ```
   NOTE: Every time you change one of the values (A, B or C), then you must redefine the anonymous function

Q1c

Keeping L, C_{S1} and C_{O1} fixed as in Q1a (L = 120 km, C_{S1} = $2.5M/km, C_{O1} = $2.0M/km), vary D over the range 1 <= D <= 80 km (in increments if 1 km) and calculate the minimum cost for each D. In a new figure, with two vertically stacked sub-plots, ***plot the optimal value of 'x' as a function of D*** in the top subplot. In the second subplot below the first, ***plot the total cost*** (in units of $100M) as a function of D.

Q1d

Another company has also given a quote, where the cost of laying the onshore pipeline is $\$C_{O2}$ per km and the cost per km of the subsea pipeline depends on the water depth η (also specified in kilometres) and is written

$$C_S = C_{S2}\,(1+\alpha\eta).$$

Consider the case where the depth of the water in kilometres (η) increases linearly with distance (in kilometres) from the shore (y) like

$$\eta\,(y) = \varepsilon y$$

In this case, it is possible to show that the cost of the ***subsea portion*** of this pipeline will be

$$\text{Cost}_{SS} = C_{S2}\left(1+0.5\alpha\varepsilon D\right)\left(\sqrt{x^2+(1+\varepsilon^2)D^2}\right)$$

Write a function (called **PipeCost2**) that calculates the TOTAL cost of the pipeline from Company 2 as a function of 'x', the distances L and D, cost factors C_{S2} and C_{O2}, the slope factor ε and the depth-cost factor α.

Assume the following parameters

- L=120 km
- D=40 km
- C_{S2} = \$2.25M/km
- C_{O2} = \$1.80M/km
- α = 0.5
- ε = 0.025

In a new figure, ***plot the cost*** of the pipeline for 0 <= x <= 120km for 121 equally spaced points. ADD your plot from part 1a for comparison and add a legend to the figure. You will see the two curves cross at one value of x.

Q1e

Find the distance 'x' at which company 1 and company 2's quotes are equal. Check your answer by calculating the cost using both Pipe Cost models. ***Print the distance (in km) and cost from both Companies (in \$M)*** to the command window, accurate to 3 decimal places (and a new line for each number/value you print).

NOTE: You must do this numerically, NOT visually. You will need to re-frame this question as a root finding problem. You can use fzero to find the root, or any other root finding method you like.

Q1f

You have just received an environmental report that shows that a near-shore reef is situated between the platform and the shore. This reef is the home of a rare species of parrot fish and you are told that the only place that you will be allowed to bring the pipeline ashore is at a location given by x=60km. At this location the quote from company 2 is more expensive.

You begin negotiating with Company 2 to drop their price. They refuse to change the cost per km for the onshore section (i.e. C_{O2}=\$1.80M/km) because they will subcontract this to another party. However you believe they might be prepared to change the cost per km of the subsea section. The depth factor, α, is not negotiable, but the factor C_{S2} might be. Currently they have quoted \$2.25M/km. What value do they need to drop this to in order to be competitive with Company 1's quote for pipe landfall at 60km?

YOU MUST find this value automatically in MATLAB, not by trial and error. Again, you should frame this question as a root finding problem, and again you may use any root finding method you like (including fzero).

Print the competitive C_{S2} ***value*** to the MATLAB command window (in \$M/km to 3 decimal places).

Do you think you would be able to negotiate them down this much? ***Print*** a one sentence statement saying why Company 2 should agree to receiving a reduced fee (zero marks associated with this, but chocolates awarded to the student who provides the most entertaining (but still ethical) response. Comments to be reported by the demonstrators, and adjudicated by Murray – no correspondence entered into.)

Question 2 [3 MARKS]

Background

A residence time distribution (or RTD) is a probability density function that describes how long fluid stays in a continuous flow chemical reactor. One way of measuring an RTD is to inject a small pulse of a chemical tracer (e.g. salt, radioactive material or coloured dye) at the inflow of the rector and then measure the signal at the outflow of the reactor.

At one extreme is a so-called "plug-flow" reactor which has no mixing and in which all of the input pulse of tracer exits the reactor at the same time. Provided there is no short-circuiting, this time is given (in seconds) as

$$\tau = V/Q$$

where V is the volume of the reactor (in m^3) and Q is the flow rate (in m^3s^{-1}).

At the other extreme, a Continuously stirred tank reactor (CSTR) is one in which each element of fluid that is injected into the reactor is instantly uniformly mixed with everything else inside the reactor. The RTD for a CSTR is a negative exponential function.

Both of these extremes are idealised concepts and can never be realised in practice. In the real world, the RTD (usually) rises quickly, and then decays slowly, and in this question we investigate some different RTD's.

NOTE: You are allowed to use the fact that time vector in the following question is evenly spaced with 1 second between values.

Q2a

You have been asked to determine if several different reactors are operating with similar behaviour, but you have not been given any information on their size, or design and have only been given a set of concentration measurements as a function of time, one for each reactor.

First you must ***open the rtd data file*** ('rtd.dat') and read it into MATLAB using importdata. The first column of data is the time of the measurement (in whole seconds), and the other columns are the concentration measurements ($C(t)$) of the tracer at the exit of the reactor for an unknown number of reactors. Determine how many different reactors have been included in the file and ***print this number*** to the command window.

Plot each of the concentration versus time curves on the same figure using a different coloured line (in order, use as many as needed of blue, green, red, yellow, magenta, black, orange). Ensure your plot has a legend that uses the text headers contained in the rtd.dat file.\.

Q2b

In order to compare the curves, they must be normalized. The normalized RTD curve (often called $E(t)$) is defined as

$$E(t) = \frac{C(t)}{\int_0^\infty C(t)\,dt}$$

Write a function called SimpRule that calculates the integral of a function using Simpson's 1/3 rule. The input parameters should be a vector of (evenly spaced) times over the time range [t_1, t_2] and a vector of function values that corresponds to the time vector.

Normalise each of your concentration curves to give $E(t)$ and in a new figure, ***plot these*** for each reactor using the same colour lines as Q2a. (**NOTE:** Instead of integrating to $t = \infty$ you may integrate to the last point in the data that you read in). ***Write the normalising value*** of $\int_0^\infty C(t)\,dt$ for each reactor to the command window, one to a line.

Q2c

The mean residence time in the reactor can be determined from the following integral

$$\tau_M = \int_0^\infty tE(t)\,dt \qquad (1)$$

The mean residence time is also known as the "first moment" of the RTD. Higher order moments are also important and can be used to categorise and compare different reactors.

The second order moment is called the *variance* of the distribution and is

$$\sigma^2 = \int_0^\infty \left(t - \tau_M\right)^2 E(t)\,dt$$

It can be normalized by the mean residence time to provide a value that indicates how big the standard deviation is compared to the mean:

$$\sigma_N = \sqrt{\frac{1}{\tau_M^2}\int_0^\infty \left(t - \tau_M\right)^2 E(t)\,dt} \qquad = \left(1/\tau_M\right)\sqrt{\sigma^2} \qquad (2)$$

The third order moment is called the *skewness* of the distribution and is

$$s^3 = \int_0^\infty (t - \tau_M)^3 E(t)\,dt$$

and can be normalized like

$$s_N = (1/\tau_M)\left[\int_0^{\infty}(t-\tau_M)^3 E(t)\,dt\right]^{1/3} \qquad = (1/\tau_M)\left(s^3\right)^{1/3} \tag{3}$$

Calculate τ_M, σ_N and s_N (i.e. equation 1, 2, 3) for each of the reactors using your Simpson's Rule function and ***print them to the command window using fprintf in tabular form*** that looks like

Reactor	Mean RT	StDev_N	Skew_N
1			
2			
etc.			

(Use 2 decimal places for these values. You do NOT need the vertical and horizontal lines of the table borders.)

If two reactors have similar values of both σ_N and s_N they are operating in a similar way, even if their mean residence times are quite different (for example, they might be the same design but have different sizes and different flow rates). For the purposes of this question, we will assume that two reactors are similar if their values of σ_N do not differ from each other by more than 10% AND if their s_N also vary from each other by less than 10%. You should use the following equation to calculate the difference

$$\%Difference = 2\frac{\left|Val_1 - Val_2\right|}{\left(\left|Val_1\right| + \left|Val_2\right|\right)} \times 100\%$$

(1

Based on the results in your table, ***automatically calculate*** if any two reactors are similar and ***print a sentence*** to the command window for each pair that are, saying which pair. If reactors 1 and 2 are similar, make sure you ***do not*** also print that reactors 2 and 1 are similar.

Poor Programming Practices [-2 Marks]

(Includes, but is not limited to, poor coding style or insufficient comments or unlabeled figures, sloppy output to the command window, no line breaks, unsuppressed lines of code, etc.)

(END OF ASSIGNMENT)