Middle East Technical University
Department of Mechanical Engineering
ME 310 Numerical Methods
Spring 2023 (Dr. Cuneyt Sert)
Study Set 1

For Homework 1 submit the answers of questions (1, 2 or 3), 9, 10. Answer 3 questions in total. Their grade percentages are not known at this point. It will be decided later.

Assigned: 18/03/2023 – Due: 28/03/2018, 23:59

Homework Rules and Suggestions:

- This is an individual assignment. Everything in your report should be the result of your own work. You are allowed to discuss its solution with your classmates and teaching staff up to a certain detail on ODTUClass. You are not allowed to use a solution manual. Put the following honor pledge at the top of your homework report and behave accordingly.

  "I understand that this is an individual assignment. I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own."

  If you have exchanged ideas with other students outside ODTUClass, you need to put their names and the extent of your discussion at the beginning of your report.

- Homework submission will be allowed until 5 minutes past the due time. Late submission is not allowed unless you have a valid excuse. In such a case, it is better if you let us know about that before the submission deadline.
- Upload your report as a PDF document (not a Word document) together with all other files (such as codes) to ODTUClass. Name your MATLAB files properly. Follow MATLAB file naming rules such as "File names cannot start with a number", "They cannot contain special characters or spaces", etc.
- Also put your MATLAB codes in your report, but in doing that make sure that you format them properly to avoid line wrapping. Codes with wrapped long lines become unreadable and we cannot understand them. If the codes are very long you can shorten them by omitting noncritical parts.
- In writing your codes, follow good programming practices such as "use explanatory header lines", "explain inputs and outputs of functions", "use self-explanatory variable names", "use comments", "use empty lines and spaces for readability", "use indentation for code blocks", etc.
- Pay attention to the format of your report. Your report should look like a serious academic work, not like a high school student work. Font types and sizes, page margins, empty spaces on pages, equations, figures, tables, captions, colors, etc. are all important to give the desired "academic work feeling". Language used is also important. Reports with poor use of English cannot get a good grade.
- Do not provide an unnecessarily long report, with useless details or wasted spaces in pages. The shorter your report, the better it is, as long as it answers the questions properly. There are about 90 students, and we can spend only about 10 minutes to grade each report. For this, your report should be easy to read and understand. Also we should be able to find the results and judge their correctness easily. We should not get lost in your report. The more we struggle to understand your report, the lower your grade will be. Use figures and tables cleverly for this purpose.
- Reports with only figures, tables and codes, but no comments or discussions will not get a good grade. Even when a question does not specifically ask for a discussion, you better write some comments on its key points and your key learnings.
- Figures and tables should be numbered and should have captions (at the bottom for figures and at the top for tables). Their titles should be self-explanatory, i.e., we should be able understand everything about the table or figure just by reading its title. They should all be referred properly in the written text (such as "… as shown in Fig. 3" or "… (See Table 2)").
- Do not use Appendices in your report. Do not put your codes in Appendices.
- You can have a numbered reference list at the end of your report. In that case, you need to refer to the references in the text.
- If you are inexperienced in programming, converting an algorithm into a code and writing it in a bug-free way can be time consuming and frustrating. This is not something that can be done at the last minute. You are advised to start working on the assignments as soon as they are assigned.

Reading Assignments:

Self-learning is an important skill. Not everything can be discussed in lectures. You need to learn certain things by yourself.

For those of you who haven't got a chance to buy the textbook yet, I extracted the pages of the following reading assignments and uploaded them to ODTUClass.

Warning: Part 1 of the textbook and the 4 chapters included in it make use of the "falling parachutist problem", which mainly uses "numerical differentiation". Many concepts are discussed through this problem. If you do not want to read the textbook thoroughly starting from page 1, but will read only the below assigned parts, you may get puzzled when you see references to this problem. Be warned.

R1) Read section PT1.1.2 Numerical Methods and Engineering Practice (page 4 of 8th edition).

R2) Read about the Scarborough criteria and learn why and how it is used in Section 3.3 (page 63 of 8th edition) of the textbook.

R3) Read Section 4.2 Error Propagation (page 99 of 8th edition).

R4) Read the Epilogue section of Part 1 (page 112 of 8th edition). Part 1 includes the first 4 chapters and its epilogue is at the end of Chapter 4. What we call "Chapter 1 Introduction" in our lectures is the first part (first 4 chapters) of the textbook.

Questions:

Q1. (Solve only one of Q1, Q2, Q3) IncrementalSearch.m code available at the course web site performs root finding using a single increment. It is also possible to perform another search inside the interval in which a root is detected. For this, the increment can be taken as a fraction of the initial increment. Modify the code such that it works in the following way

```
IncrementalSearch(f, xMin, xMax, dx, nRoot, nSearch, fraction)
```

where the last two new arguments are

`nSearch:` number of searches we need to perform to find each root (positive integer)
`fraction:` the amount we want to lower the previous increment when we start a new search (real number between 0 and 1)

A sample call of the function is as follows

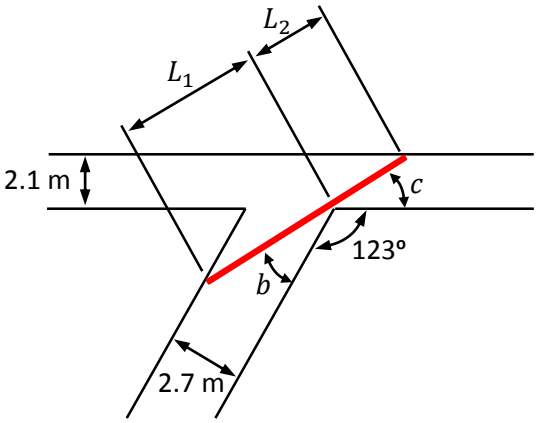
```
f = @(x) sin(x.^2) + exp(-0.5*x);
IncrementalSearch(f, 0, 100, 0.01, 5, 3, 0.1)
```

Here we want to find the first 5 roots of function f in the interval [0, 100]. The starting increment is 0.01. For finding each root, we will perform 3 searches and the increment of each search will be 0.1 times of the previous search.

To elaborate, start from 0 and perform a search with increment 0.01. When you detect the first root, perform another search in the detected interval with increment 0.001. Repeat this once again in the new interval with the increment 0.0001. After the 3rd search, finalize searching for the first root with a linear interpolation. Then start searching for the second root with the user specified initial increment of 0.01. Perform 3 similar searches and one final interpolation to locate the 2nd root. Repeat this 5 times to locate all 5 roots. Note that all the parameters given in this explanation are just examples. Your code should work with any set of meaningful inputs. We'll test it with several different cases.

Hint: Instead of writing the whole algorithm inside one function, you can write a new function that calls the original incremental search function (or a modified version of it) as many times as needed.

Q2. (Solve only one of Q1, Q2, Q3) Two intersecting mine shafts meet at an angle of 123°. The top shaft has a width of 2.1 m, and the bottom one is 2.7 m wide. We want to move a ladder in between the shafts and want to calculate the longest ladder that can turn at the intersection. We neglect the thickness of the ladder, and assume the ladder is not tipped as it is maneuvered around the corner.



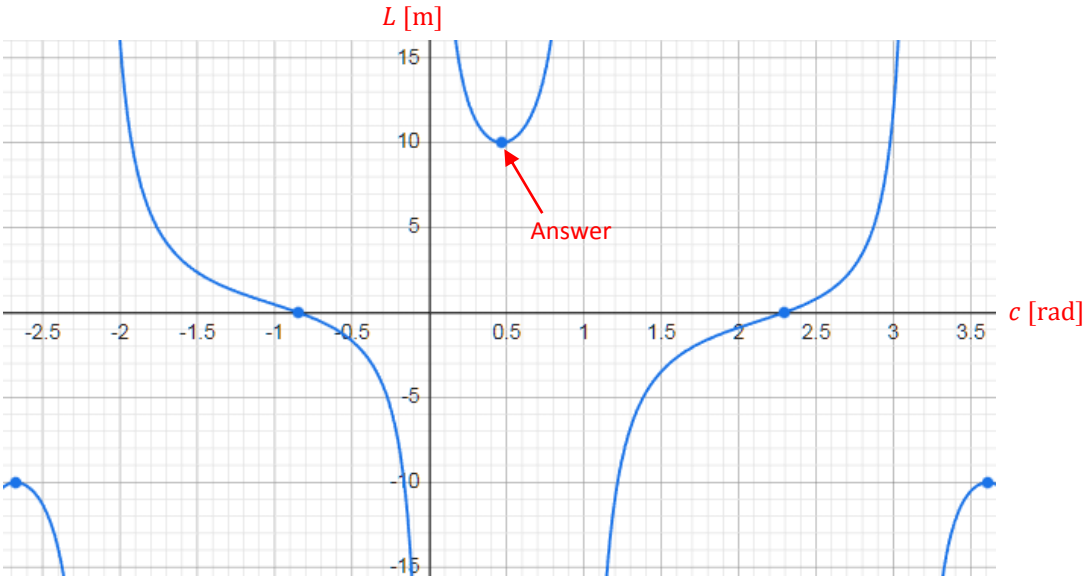
Length of the ladder can be formulated as

$$L = L_1 + L_2 = \frac{2.7}{\sin(b)} + \frac{2.1}{\sin(c)}$$

using $b = \pi - 123\frac{\pi}{180} - c$ (use radians for angles)

$$L = \frac{2.7}{\sin(\pi - 123\frac{\pi}{180} - c)} + \frac{2.1}{\sin(c)}$$

This equation gives the ladder length L as a function of angle c. Its plot is given below. It has several discontinuities due to the cyclic sine functions appearing in the denominators. The first local minimum L shown in the figure is the maximum possible ladder length that we want to find.



a) Plot this function and zoom in to find the c angle that minimizes L, and the corresponding minimum L value (which is the maximum possible ladder length). Obtain these values with at least 3 digits of accuracy.

Draw the mine shafts to the correct scale (just use paper, pencil and ruler) and place the ladder at the calculated c angle to see if its length is equal to the determined length or not.

b) This is an optimization problem. It is about finding the minimum of a function. It can numerically be solved using incremental search. Modify the IncrementalSearch.m code such that it detects not the sign change of the function, but whether the function is increasing or decreasing. Start the search from a meaningful small positive c value. Starting from zero does not make sense and it is also not possible because $L(0)$ goes to infinity. At first, L will show a decreasing trend. You can detect this by comparing f1 and f2 values in the code. This trend will change at one point, indicating that the function goes through an optimum point (minimum in this case). Note that we are interested in finding only the first optimum. Use a small enough increment to determine the minimum value with 3 digits of accuracy, and compare your solution with the graphical one.

c) MATLAB has the built-in `fminbnd` function to find the minimum of a given function in a given interval. You can read about it here. https://www.mathworks.com/help/matlab/ref/fminbnd.html. Use it to determine the minimum value of $L(c)$.

d) Does this problem have an analytical solution? Find it and show the details of your calculation.

Q3. (Solve only one of Q1, Q2, Q3) Bubble sort is an easy-to-program, but inefficient sorting technique. The idea behind the sort is to move down through an array comparing adjacent pairs and swapping the values if they are out of order. To sort the array completely, it may need to pass through it many times. It's explained in the following short video. Please watch it. https://www.youtube.com/watch?v=yIQuKSwPlro&t=205s&ab_channel=KCAng

Implement this algorithm in a MATLAB function, which takes a row array as an input and returns back the sorted row array. First test it with the array given in the video. Then test it with different arrays. If you want, you can use the `rand` and `randi` functions of MATLAB to generate test arrays of random real numbers and integers.

Note that MATLAB has a built-in sort function called `sort`. There is also a function called `sorted`, that checks whether an array is sorted or not. You can use them to test your code.

Speed of sorting can be an issue for large arrays. Test the speed of your code using a large random array of real numbers. You can do this as follows.

```
a = rand([1000000,1]);
tic; myBubbleSort(a); toc
```

It is a good idea to do the time measurement a few times (for the same array) and take the average of the measured times. This is especially important if the array is short and the run times are very short.

Repeat the time measurement with MATLAB's built-in sort function and compare times. Do this with arrays of several lengths (from very small ones to very large ones). Plot the variation of run times of both functions with the array length on the same figure. You may need to use logarithmic axes to get an easy to read figure. Discuss how the performances of your function and the built-in function change with array length. We expect to see that the built-in function performs much faster because we know that bubble sort is inefficient, but it is a good experience to really code and see this.

Warning: You can easily find several codes on the internet that performs bubble sort. Please write your own code. Do not cheat.

Q4. Write a MATLAB code to evaluate e^{-10} using the following two equations with 30 terms. Print out the result and the absolute and relative percent true errors after adding each term. Discuss the results and the behaviors you see in detail in terms truncation and round-off errors.

$$e^{-x} = 1 - x + \frac{x^2}{2} - \frac{x^3}{3!} + \cdots \qquad \text{and} \qquad e^{-x} = \frac{1}{e^x} = \frac{1}{1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots}$$

Q5. The following mathematical expression simplifies to $x^2/x^2 = 1$.

$$\frac{(x+y)^2 - 2xy - y^2}{x^2}$$

Calculate this expression using MATLAB for the following numbers using single and double precision. Comment on the results and the cause of the behavior you see. Which operation is responsible for the accuracy loss?

x	y
0.01	1000
0.001	1000
0.0001	1000
0.00001	1000
0.000001	1000
0.0000001	1000

Q6. What is machine epsilon? What does MATLAB's `eps` function return and what does it mean?

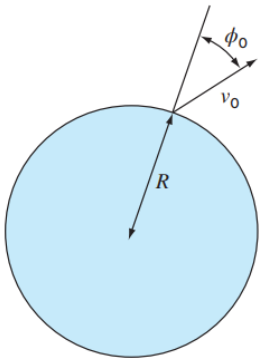
Q7. a) Do you think Heron's formula for finding square root of a number always converges? Can you find a case where it does not converge? Is the initial guess important here?

b) Do a web search to find alternative algorithms for square root evaluation.

Q8. Derive the 0th, 1st , 2nd, …, 8th order Maclaurin series expansions of the function $f = x^2 e^{-x}$ and plot them in the interval [0, 2] together with the original function.

Q9. Derive the 2nd order Taylor series expansion of the function $f = \sin(x) + 2\cos(y)$ about point (0.5, 0.5) and evaluate it at point (0.51, 0.51). Also calculate the absolute and relative percent true errors of this calculation.

Q10. (Work on the "R3 reading assignment" before solving this question) A missile leaves the ground with an initial velocity v_0 forming an angle ϕ_0 with the vertical. The maximum desired altitude is αR, where R is the radius of the earth. The laws of mechanics can be used to show that



$$\sin(\phi_0) = (1+\alpha)\sqrt{1 - \frac{\alpha}{1+\alpha}\left(\frac{v_e}{v_0}\right)^2}$$

where v_e is the escape velocity of the missile. It is desired to fire the missile and reach the design maximum altitude within an accuracy of ±2%. Determine the range of values for ϕ_0 if $v_e/v_0 = 2$ and $\alpha = 0.25$.

Q11. (Work on the "R3 reading assignment" before solving this question) Evaluate and interpret the condition number of the function $f = e^x$ at $x = 0.1, 1, 5$ and 10.