

Fall 2020 Engineering 7 Midterm Exam

Questions and Instructions

Logistics

- You may start to work with your midterm exam on 8:00 am Sunday October 11th, and you must finish it and submit your report and MATLAB code in a zip file online before Sunday October 18th 11:59 pm.
- You are supposed to submit four M-files for four Problems. For example, you may name them as *MyProblem1.m*, *MyProblem2.m*, *MyProblem3.m*, and *MyProblem4.m*. For each problem all the functions, test case example inputs, execution line commands, and plotting commands in one problem are within one single M-file. This means each file will be an independent script starting with the commands to run the test cases and followed by the definition(s) of you function(s), as shown in lecture demos;
- You are supposed to write a short midterm exam report that includes a *brief* written summary of your work and findings, which includes the outputs of your calculation for testing examples, graphs or pictures, accuracy or error analysis, etc.;
- You must complete this midterm exam by yourself only, i.e. You must write your own MATLAB scripts that implement the test case(s) provided for each problem. Any collaboration is prohibited, and it will be viewed as plagiarism. We have a code to check the similarity between codes from different students. If two MATLAB codes from two individuals are identically the same, they will both receive failing grades;
- You are not allowed to use MATLAB built-in functions: *find*, *sort*, *sum* to do this midterm exam, and
- The deadline for students with disability is Wednesday (October 21th) 11:59 pm.

1 Sort and (bar) plot a matrix

<code>function [row_sort, col_sort, fig] = myMatrixPlotSort(myMatrix)</code>		
Input	Type	Description
<code>myMatrix</code>	4x5 double	Unsorted input matrix.
Output		
<code>row_sort</code>	4x5 double	Matrix sorted with ascending rows.
<code>col_sort</code>	4x5 double	Matrix sorted with ascending columns.
<code>fig</code>	1x1... <code>matlab.ui.Figure</code>	Figure with bar plot of 1D sorted values from input matrix; see Fig. 1.

Write a MATLAB function to sort a general $M \times N$ matrix in ascending order in terms of both row-wise and column-wise fashions. For example, if you are given the following matrix (test-case example),

$$\begin{bmatrix} 15 & 2 & 8 & 12 & 10 \\ 11 & 9 & 3 & 14 & 5 \\ 6 & 4 & 1 & 7 & 13 \\ 7 & 11 & 14 & 5 & 7 \end{bmatrix}$$

your code has to be able to sort the matrix row-wise as

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 6 & 7 & 7 & 7 \\ 8 & 9 & 10 & 11 & 11 \\ 12 & 13 & 14 & 14 & 15 \end{bmatrix}$$

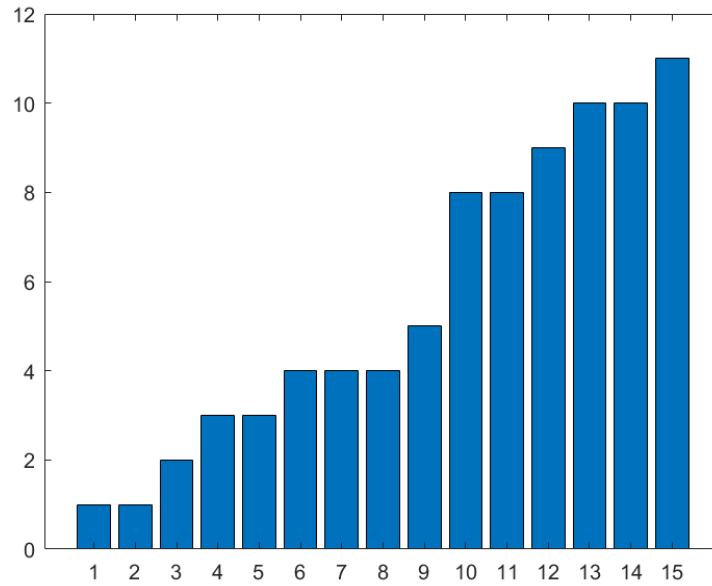


Figure 1: Example of bar plot for sorted matrices.

and column-wise as

$$\begin{bmatrix} 1 & 5 & 7 & 10 & 13 \\ 2 & 5 & 7 & 11 & 14 \\ 3 & 6 & 8 & 11 & 14 \\ 4 & 7 & 9 & 12 & 15 \end{bmatrix}$$

Your function should also return a figure that graphs the sorted values from the input array as a 1D bar plot; see Fig. 1.

1.1 Test Case

Use the example matrix provided above.

1.2 Report

- Include the bar plot for the test case.
- You may discuss and compare computing times of different sorting algorithms, but it is not required.

2 Riemann sum

function [R,e,fig] = myRiemann(a,b,npts,R_exact)		
Input	Type	Description
a	1x1 double	Lower bound of plot and integration.
b	1x1 double	Upper bound of plot and integration.
npts	1x1 double	Number of points for plot and Riemann sum.
R_exact	1x1 double	Exact integration value for error calculation.
Output		
R	1x1 double	Result of numerical integration.
e	1x1 double	Error of numerical integration.
fig	1x1... matlab.ui.Figure	Plot of piece-wise continuous function.

Consider the piece-wise continuous function

$$f(x) = \begin{cases} x^2 & , x \leq 1 \\ x^3 & , x \geq 1 \end{cases}$$

Write a MATLAB function that will:

- (1) Plot the function in the range $[a, b]$ on linear axes with a title that includes the values of a and b ;
- (2) Integrate the function from a to b by using the Riemann sum, i.e.

$$R = \int_a^b f(x)dx = \sum_{i=1}^n f(x_i^*)\Delta x_i$$

where $a = x_0 < x_1 < x_2 < \dots < x_n = b$, and $x_i^* \in [x_{i-1}, x_i]$ (Note that there are $n+1$ points here, and $\Delta x_i = (b - a)/n$). Choose x_i^* as the midpoints, i.e. $x_i^* = \frac{1}{2}(x_{i-1} + x_i)$; and

- (3) Calculate R and compare the value with the exact solution, i.e. calculate

$$e = \left| \frac{R - R_{exact}}{R_{exact}} \right|$$

2.1 Test Cases

Test your code using $a = -3$ and $b = 3$ first with $n = 10$ and then with $n = 1000$.

2.2 Report

- Include your plot of the continuous, piece-wise function.
- Describe the result and error of the integration for each value of n .
- What is the computation complexity of your code for this problem, using Big-O notation?

3 Count appearances

function [locs] = mySearch(myArray,target)		
Input	Type	Description
myArray	1xn double	1D input array of any length.
target	1x1 double	Target value to find and count.
Output		
locs	1xm double	Locations in the input array where the target values are found.

Write a general binary search function that can find a target value with one or multiple appearances in an array. Your function should include a sort before the binary search so that the input can be arrays of arbitrary order.

3.1 Test Cases

Use the following array to test your code:

$$A = [-2, -1, 0, 1, 1, 2, 3, 3, 4, 4, 4, 5, 8, 8, 9, 10, 10, 11];$$

and test the targets: 1,3, 4, 8, and 10 separately. For this array, your output for the target value 1 should be [4, 5].

3.2 Report

In the report, you can describe the key steps that you add to make your binary search find a target with multiple positions.

4 Collatz conjecture

function [y,k]=myCollatz(n)		
Input	Type	Description
n	1x1 double	Starting number, any positive integer.
Output		
y	1x1 double	Output value of the given function $f(n)$
k	1x1 double	Total recursive iterations to get to this output value.

Write a MATLAB function implementing the following mathematical function,

$$f(n) = \begin{cases} \frac{n}{2}, & \text{if } n \text{ is even;} \\ 3n + 1, & \text{if } n \text{ is odd;} \end{cases}$$

where n is a natural number or positive integer.

Your MATLAB function should perform this operation repeatedly through recursion, beginning with any positive integer, n , and taking the result at each step as the input at the next. That is

$$a_i = \begin{cases} n, & \text{for } i = 0 \\ f(a_{i-1}), & i = 1, 2, 3, \dots, k \end{cases}$$

where k is a finite value, and n can be any positive integer.

4.1 Test Case

For the test case, please choose $a_0 = n = 1234567890$. Find how many iteration steps, k , are needed to find $a_k = 1$.

4.2 Report

In the report, give the number of iteration steps required for the test case. You may experiment with other values and describe if you can find any pattern but it is not required.

Remark:

Once you finish writing your code, provided it is correct, if you can find a positive integer n that does not converge to 1, you basically disprove that the famous *Collatz conjecture* is false. You will not only get A+ in the class (forget the rest of HWs and final), and you will also be the winner of the next Fields Medal ! In other words, you will become world-famous instantaneously, this is because the Collatz conjecture “*is an extraordinarily difficult problem, completely out of reach of present day mathematics.*” It has been more than 80 years, no one has been able to prove it or disprove it.