CS/MATH 375, Fall 2024 — HOMEWORK # 1
Due : Friday, Aug. 30 at 10:00 pm on Canvas

Instructions

- **Report:** In general, your report needs to read coherently. That is, start off by answering question 1. Fully answer the question, and provide all the information needed to understand your answer. If Matlab code or output is part of the question, include that code or output (e.g., screenshot) alongside your narrative answer. If discussion is required for a question, include that. Overall, your report is your narrative explanation of what was done, your answers to the specific questions, and how you arrived at your answers. *Your report should include your Matlab scripts, code output, and any figures.*

- **What to hand in:** Submission must be one **single PDF** document, containing your entire report, submitted on Canvas.

- **Partners:** You are allowed to (even encouraged) to **work in pairs**. If you work with a partner, only one member of the group should need to submit a report. On Canvas, both partners should join a group (numbered 1 through 15). Then either member can upload the report for the entire group. Groups of more than 2 students are not allowed.

- **Typesetting:** If you write your answers by hand, then make sure that your handwriting is readable. Otherwise, I cannot grade it.

- **Plots:** All plots/figures in the report must be generated in Matlab and not hand drawn (unless otherwise specified in the homework question).

  In general, make sure to (1) title figures, (2) label both axes, (3) make the curves nice and thick to be easily readable, and (4) include a legend for the plotted data sets. The font-size of all text in your figures must be large and easily readable.

Reading: Read the "Matlab Tutorial" posted to the Canvas page and the second tutorial posted in the "Slightly less basic basics" section at the bottom of the "Matlab Tutorial".

Problems:

1. Suppose `z=[10 40 70 90 20 30 50 60]`. What does this vector look like after each of these commands? Assume that the commands are done sequentially. You do not need to submit any code for this problem.

   (a) `z(1:3:7)=zeros(1,3)`

   (b) `z([3 4 1])=[]`

2. (a) Use the `linspace` function to create vectors identical to the following created with colon notation:

      i. `t=1:4:25`

      ii. `x=-11:1`

   (b) Use colon notation to create vectors identical to the following created with the `linspace` function:

      i. `v=linspace(-10,-8,6)`

      ii. `r=linspace(0,1,5)`

3. Given that `t=0:0.1:1; y=sin(pi*t);`, write a single-line MATLAB code that returns each the following.

   (a) $\sum_{k=1}^{N} t_k$ (use the built-in Matlab `sum` function)

   (b) $\sum_{k=1}^{N} t_k y_k$

   (c) $\sum_{k=1}^{N} t_k^2$

4. Write a MATLAB script to plot the four functions x, $\exp(x)$, x^2, x^3 over the interval $0 \le x \le 1$ using `plot`, `semilogy`, `semilogx`, and `loglog`.

   For each type of plot, all four curves should be shown on the same figure/axis. Each function should have its own distinct color. Each plot should be clearly labeled, use large fonts, have a title and labels on each axis, and have a legend to describe each curve. Submit your figures and comment on the advantages / disadvantages of the different plotting commands.

5. (a) Write a MATLAB function called `my_mean` which takes four arguments: a function name `fun`, a number `a`, another number `b` satisfying `a<=b`, and a positive integer `N`. The function `my_mean` should return an approximate value for

$$\int_a^b f(x)\, dx,$$

   where $f(x)$ represents the function `fun`. The approximate value should be calculated via the formula

$$\int_a^b f(x)\, dx \approx \frac{h}{2}(f(x_1) + f(x_N)) + \sum_{j=2}^{N-1} h f(x_j),$$

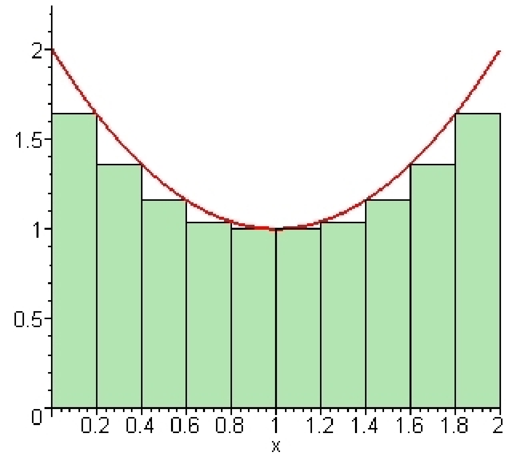

Figure 1: Example Riemann sum for approximating an integral, where the function $f(x)$ is in red, and the approximate integral is comprised of the green rectangles.

where $h = (b-a)/(N-1)$ and $x_j = a + (j-1)h$. This formula approximates the integral by calculating a simple Riemann sum (i.e., computes the area of rectangles under the curve) as depicted in Figure **??**.
An example call to the function would look like

$$\texttt{my\_mean}(@\texttt{sin}, 0, 2, 100).$$

It is recommended that you use the above `linspace` function or colon notation to create your vector of x_j points, and that you use the `sum` function for computing the approximate integral.
Note: As an aside, the @ symbol above is used to create a "function handle" for the built in function `sin`. This can be a useful construction and I recommend reading this this help document on function handles.

(b) Matlab lets you easily create more complicated functions from simpler, built-in ones like `sin`, `log` or `sqrt`. Write another function, `my_fun`, which returns the value of $x\,e^x$ given an input x. The function `my_fun` should be a function of `x` only. The input `x` could be a vector.

(c) Use `my_mean` and `my_fun` to compute an approximation to

$$\mathcal{M} = \int_{-1}^{1} xe^x \, dx.$$

Compute approximations for $N = 10, 20, 40, 80, 160, 320, 640$, and 1280. In this case the exact solution can be computed with pen and paper. Show that the exact value is

$$\frac{2}{e}.$$

Plot the absolute error between $\mathcal{M}$ and your approximations vs. N. You may want to use `loglog` for the plot. **Make sure to use** `xlabel` **and** `ylabel` **to label your plot**.

(d) Print a table listing, in each row, N, the corresponding approximation of $\mathcal{M}$, and the absolute error between the approximation and $\mathcal{M}$. Do you see a pattern? What is the type/rate of convergence for your `my_mean` algorithm?