MCEN 3030 Midterm II

Spring 2020, Midterm 2

INSTRUCTIONS:

- Open notes, open book. Calculators, computer, MATLAB are allowed. Notes = anything you have written yourself, or any printed material that has been distributed in class or the course web site (sample codes, assignments and solutions).
- Receiving assistance from or sharing of exam material with another person is strictly prohibited.
- The exam has two main parts, MATLAB-based questions and short-answer questions.
  - MATAB: Before the exam starts, download the MATLAB code template from the course web site:
    `https://spot.colorado.edu/~henzed/MCEN3030_s2020/exams/midterm2_XXXXXX.m`
    where XXXXXX is your MEID (no hyphens). **All answers to these questions must be included in your code file itself.**
  - Short-answer questions: these are available on Canvas, just like a ready quiz, under Midterm2 Short Questions
- At the end of the exam (9 pm), upload your MATLAB m-file to the Canvas dropbox and submit all of your short-answer question responses. I will leave these open an extra 20 minutes to allow time for students to deal with connection issues, uploading, etc. Please keep in mind though they will close precisely at 9:20 and that it takes a few minutes to upload, submit, etc. If all else fails, as a last resort you can email me a copy of your code or answers to short-answer problems, but these must be received by 9:20.
- In this file, only include code for questions labeled **MATLAB QUESTION** that are included in this pdf (Question 1 - 3). The remaining 25 points comes from short-answer questions that you complete directly on Canvas.
- I will be available on Zoom to answer questions: https://cuboulder.zoom.us/my/davenhenze

1. MATLAB	25	
2. MATLAB	25	
3. MATLAB	25	
4. Canvas	25	
Total	100	

1. [25 pts, **MATLAB QUESTION**]

In this problem we will explore a new Newton Cotes method. This method is shown in composite (multi-application) form in the figure below for the case of 8 intervals. It is used to integrate from a to b using a set of n evenly distributed points, x_1 through x_n, where interestingly the endpoints $x_1 = a$ and $x_2 = b$ as well as points $x_5, x_9, x_{13}, \ldots$ are not used. The total number of intervals must be a multiple of 4 for this method to work (i.e., the total number of data points must be one larger than a multiple of 4). In composite form, this method has a truncation error that is proportional to the fourth power of the interval spacing, h.

The integration estimated can expressed mathematically as

$$\int_a^b f(x)dx \approx 4h/3 \left[\left(2 \sum_{i=2,4,6,8,\ldots}^{n-1} f(x_i) \right) - \left(\sum_{i=3,7,11,15,\ldots}^{n-2} f(x_i) \right) \right]$$

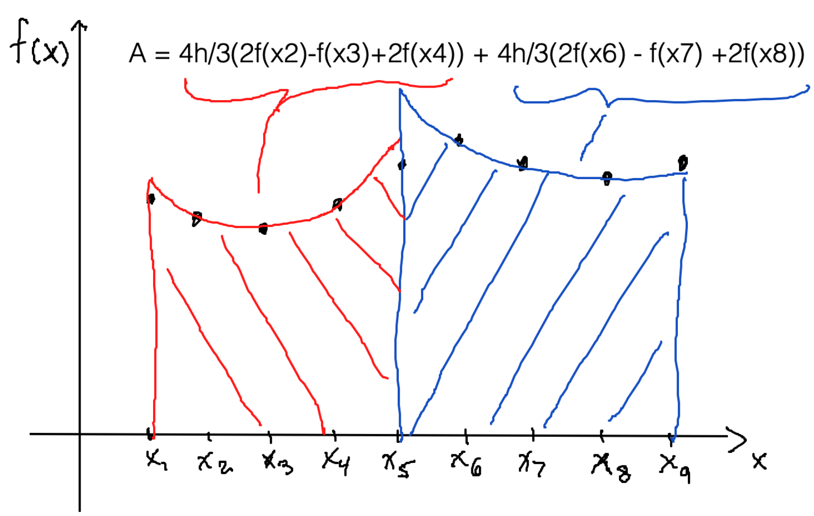
where $a = x_1$, $b = x_n$, and $h = (b-a)/(n-1) = x_{i+1} - x_i$. Pay close attention to the bounds on the summations in this formula. Note: if you are not able to get part (a) to work correctly, you should still proceed with parts (b) - (d) using your newNC routine as if it did work.

(a) Complete the function `newNC`. Code is provide that will allow you to test this for a simple case (integrating x^3) for which this method should produce the exact answer, since the truncation error scales like $f''''(x)$.

(b) Use your `newNC` function to estimate the integral of $\int_a^b \ln(x)x^3 dx$ over the bounds provided in the template file, using 4 intervals.

(c) Estimate the integral from part (b) again, but this time use 8 intervals.

(d) Without using a value of the true integral, estimate the error in your solution for part (c).

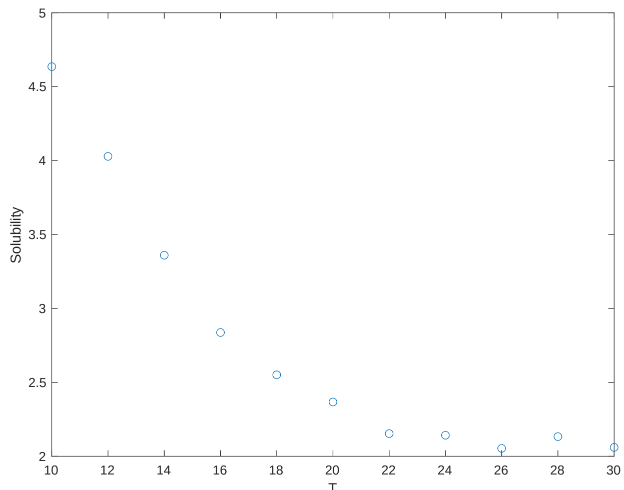


2. [25 pts, **MATLAB QUESTION**] The solubility, s, of a contaminant in jet fuel depends on temperature as

$$s = be^{m/T}$$

Several experiments have provided the data shown in the figure below. You may make use of any built-in MATLAB commands for this problem with the exception of `nlinfit`.

(a) Using the data pre-loaded in the vectors `T` (in units of °C) and `s`, make a plot of the data, which should look like this (use any shape and size markers you want, not connected by any line):



(b) Calculate and display the values of m and b.

(c) Add a plot of your regression function $s(T)$ to the plot as a solid line, including a legend.

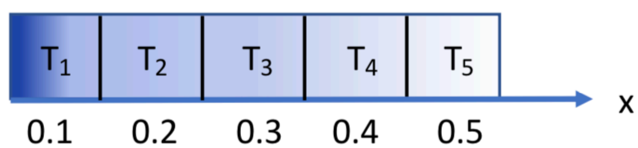
(d) Estimate the solubility at T = 21 C.

3. [25 pts, **MATLAB QUESTION**] Temperatures in a 1D cooling rod have been measured at the different x positions shown in the figure below at a specific instant in time. The temperature rate of change, $\frac{\partial T(x,t)}{\partial t}$, is given by the heat equation

$$\frac{\partial T(x,t)}{\partial t} = \mu \frac{\partial^2 T(x,t)}{\partial x^2}$$

Evaluate the instantaneous temperature rate of change (K/s) in the center of the rod using finite difference methods. In other words, calculate $\mu \frac{\partial^2 T(x,t)}{\partial x^2}|_{x=0.3}$ using finite difference methods applied to the data for $T(x)$ that is provided in the MATLAB template file. Assume that the thermal conductivity μ is 1.0 m^2/s.

Provide an answer that is accurate to within 1.0 K/s. Also provide a quantitative estimate of the uncertainty in your answer. State in the comments of your code what method you are attempting to use.



4. [25 pts, **Short-answer questions**] See Canvas

Table 6-1: Finite difference formulas.

First Derivative		
Method	Formula	Truncation Error
Two-point forward difference	$f'(x_i) = \dfrac{f(x_{i+1}) - f(x_i)}{h}$	$O(h)$
Three-point forward difference	$f'(x_i) = \dfrac{-3f(x_i) + 4f(x_{i+1}) - f(x_{i+2})}{2h}$	$O(h^2)$

Table 6-1: Finite difference formulas.

Two-point backward difference	$f'(x_i) = \dfrac{f(x_i) - f(x_{i-1})}{h}$	$O(h)$
Three-point backward difference	$f'(x_i) = \dfrac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h}$	$O(h^2)$
Two-point central difference	$f'(x_i) = \dfrac{f(x_{i+1}) - f(x_{i-1})}{2h}$	$O(h^2)$
Four-point central difference	$f'(x_i) = \dfrac{f(x_{i-2}) - 8f(x_{i-1}) + 8f(x_{i+1}) - f(x_{i+2})}{12h}$	$O(h^4)$
Second Derivative		
Method	Formula	Truncation Error
Three-point forward difference	$f''(x_i) = \dfrac{f(x_i) - 2f(x_{i+1}) + f(x_{i+2})}{h^2}$	$O(h)$
Four-point forward difference	$f''(x_i) = \dfrac{2f(x_i) - 5f(x_{i+1}) + 4f(x_{i+2}) - f(x_{i+3})}{h^2}$	$O(h^2)$
Three-point backward difference	$f''(x_i) = \dfrac{f(x_{i-2}) - 2f(x_{i-1}) + f(x_i)}{h^2}$	$O(h)$
Four-point backward difference	$f''(x_i) = \dfrac{-f(x_{i-3}) + 4f(x_{i-2}) - 5f(x_{i-1}) + 2f(x_i)}{h^2}$	$O(h^2)$
Three-point central difference	$f''(x_i) = \dfrac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1})}{h^2}$	$O(h^2)$
Five-point central difference	$f''(x_i) = \dfrac{-f(x_{i-2}) + 16f(x_{i-1}) - 30f(x_i) + 16f(x_{i+1}) - f(x_{i+2})}{12h^2}$	$O(h^4)$
Third Derivative		
Method	Formula	Truncation Error
Four-point forward difference	$f'''(x_i) = \dfrac{-f(x_i) + 3f(x_{i+1}) - 3f(x_{i+2}) + f(x_{i+3})}{h^3}$	$O(h)$
Five-point forward difference	$f'''(x_i) = \dfrac{-5f(x_i) + 18f(x_{i+1}) - 24f(x_{i+2}) + 14f(x_{i+3}) - 3f(x_{i+4})}{2h^3}$	$O(h^2)$
Four-point backward difference	$f'''(x_i) = \dfrac{-f(x_{i-3}) + 3f(x_{i-2}) - 3f(x_{i-1}) + f(x_i)}{h^3}$	$O(h)$
Five-point backward difference	$f'''(x_i) = \dfrac{3f(x_{i-4}) - 14f(x_{i-3}) + 24f(x_{i-2}) - 18f(x_{i-1}) + 5f(x_i)}{2h^3}$	$O(h^2)$
Four-point central difference	$f'''(x_i) = \dfrac{-f(x_{i-2}) + 2f(x_{i-1}) - 2f(x_{i+1}) + f(x_{i+2})}{2h^3}$	$O(h^2)$
Six-point central difference	$f'''(x_i) = \dfrac{f(x_{i-3}) - 8f(x_{i-2}) + 13f(x_{i-1}) - 13f(x_{i+1}) + 8f(x_{i+2}) - f(x_{i+3})}{8h^3}$	$O(h^4)$