CHEN.3170 (sec 201) Applied Engineering Problem Solving
Final Exam Fall 2017

NOTE: Do all eight problems. The point distribution for each problem is given below. Show your work for maximum credit (use the back side of the individual sheets as needed). Good Luck!!!

Problem 1 2-D Function Evaluation and Plotting in Matlab (using functions) (15 points)

In Fluid Mechanics, the friction factor, f, is used to account for energy losses within long piping systems. One correlation for f in the turbulent flow regime is the Swamee-Jain empirical relation, where ε/D is the relative roughness and Re is the Reynolds number:

$$f = \frac{0.25}{\left(\log_{10}\left(\frac{\varepsilon/D}{3.7} + \frac{5.74}{Re^{0.9}}\right)\right)^2}$$

Note: The questions below ask you to write a series of Matlab routines to evaluate and plot this function. Of course, this can be done in a number of ways, but the tasks here take you down a specific path to evaluate your understanding of several features within Matlab -- **so please follow the instructions given here carefully**. Also note that, since the Reynolds number can vary over a wide range, one usually plots f vs. Re on a logarithmic scale -- so you will probably want to use Matlab's ***logspace*** command to set up the Re variable and use the ***semilogx*** command to do the actual plotting.

a. Write a function routine to compute f given the value of Re and ε/D as inputs. The function should allow a **vector input for Re**, but only **scalar values of ε/D** are treated. The output, f, should be the same size as Re. For consistency, the first line of the function should be

   function f = ffactor(Re,rr)

   where rr is the scalar code variable for the relative roughness, ε/D.

b. Write a Matlab script file that uses **ffactor.m** to evaluate and plot f vs. Re for $10^4 \le Re \le 10^7$ for three different values of ε/D (let ε/D = 0.0001, 0.001, and 0.01). Store the values of f in a 2-D array and put all three curves on the same plot axes. Properly annotate your plot!

Problem 2 Implementation of Discrete Equations within Matlab (15 points)

The evaluation of definite integrals using numerical methods is a common task that is needed in many applications. One relatively simple technique, referred to as **Simpson's 1/3 Rule**, can be summarized in discrete form as follows:

Simpson's 1/3 Rule: An estimate of the definite integral, $I_{exact} = \int_a^b f(x)dx$, is given by

$$I_{estimate} = \frac{h}{3}\left[f(x_1) + 4\left\{ \sum_{i=2,4,\cdots}^{N} f(x_i) \right\} + 2\left\{ \sum_{j=3,5,\cdots}^{N-1} f(x_j) \right\} + f(x_{N+1}) \right]$$

where N is the number of uniform intervals, with the interval width, h, given by $h = (b-a)/N$.

a. The goal is to implement this computation within a complete Matlab **function file** to compute the definite integral of a function using **Simpson's 1/3 Rule**. Assume that the variables, **N** and **h**, and the vector function **f** (containing N+1 evenly spaced values) are already available. These quantities are passed into the Matlab function file via the input argument list. Only a single scalar variable containing the estimate of the integral, **ival**, is returned to the main program.

   For consistency, assume that the first line of the function file is given by

   function ival = simp3(f, h, N)

   Your job is to write the remainder of this file to implement **Simpson's 1/3 Rule**. Be careful to use the proper Matlab syntax in your program!

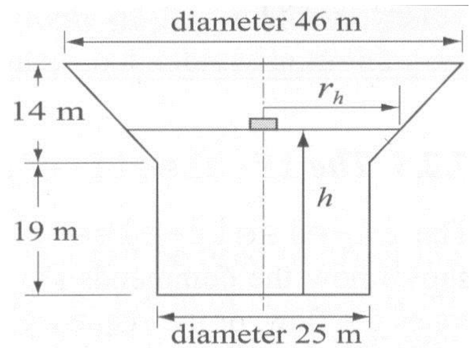
b. Now write a main program that calls your **simp3.m** function file to evaluate

$$I = \int_0^2 xe^{-x}dx$$

for N = 10. Make sure everything is properly defined and is consistent with the **simp3.m** function written in Part a.

Problem 3 Use of Conditional Statements within Matlab (10 points)

A water tower has the geometry shown in the sketch. The lower part is a cylinder and the upper part is an inverted frustum cone. Inside the tank there is a float that indicates the water level, h. The volume of water in the tower versus h is given by the following equations:



for $h \le H_B$ $\qquad V = \pi R_B^2 h$

for $H_B \le h \le H_T$ $\qquad V = \dfrac{\pi}{3m}\left(r_h^3 - R_B^3\right) + \pi R_B^2 H_B$

where $\quad m = \dfrac{R_T - R_B}{H_T - H_B} \quad$ and $\quad r_h = R_B + m\left(h - H_B\right)$

For the tower shown in the diagram, we have

$R_B = 12.5$ m $\qquad R_T = 23$ m $\qquad H_B = 19$ m $\qquad H_T = 33$ m

Write a Matlab script file to evaluate and plot the volume of water in the tower versus fluid height, h, over the range $0 \le h \le H_T$.

Problem 4 Looping Structures in Matlab (10 points)

a. Write a short segment of Matlab code to evaluate the following series for N = 25 terms:

$$S = \sum_{k=1}^{N} (-1)^{k+1} \frac{1}{2k-1}$$

b. Write a short segment of Matlab code to evaluate the following infinite series, where tol = 0.001 represents the limiting condition for stopping the infinite series to some finite number of terms -- that is, when the magnitude of the relative contribution of the next term in the series is less that tol, you should stop adding additional terms.

$$S = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{1}{2k-1}$$

Problem 5 Linear Algebra Calculations (15 points)

a. Given the matrix

$$\underline{\underline{A}} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & -2 \end{bmatrix}$$

find $\underline{\underline{\mathbf{A}}}^{\mathbf{-1}}$ using the expression $\underline{\underline{A}}^{-1} = \dfrac{\underline{\underline{C}}^{T}}{\det \underline{\underline{A}}}$, being sure to show your work for full credit!

b. Given the following matrices,

$$\underline{\underline{A}} = \begin{bmatrix} 0 & -2 & 1 \\ 4 & 1 & -1 \\ 2 & -2 & -3 \end{bmatrix} \qquad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \underline{b} = \begin{bmatrix} 3 \\ -7 \\ -11 \end{bmatrix}$$

Write $\underline{\underline{A}}$ and $\underline{b}$ as an augmented matrix, perform a series of valid row operations to put the system into upper triangular form, and then find the solution vector $\underline{x}$. Show your work!

Problem 6 Working with Taylor Series (10 points)

a. Using the definition of the Taylor series as given below, develop an expansion for f(x) = sin(ax) about the point $x_o = 0$ that includes the first five (5) terms of the expansion (i.e. up to and including the term containing the 4th derivative), where

$$f\left(x_o + (x - x_o)\right) = f(x) = \frac{f(x_o)}{0!} + \frac{f'(x_o)}{1!}\left(x - x_o\right) + \frac{f''(x_o)}{2!}\left(x - x_o\right)^2 + \quad \cdots$$

b. Now evaluate f(x) = sin(2x) at $x_1 = \pi/4$ and $x_2 = \pi/8$ and compare the result from the truncated Taylor series expansion developed in Part a with the exact result. Here you should compute the absolute error in the approximate result from the truncated Taylor series for the two different values of x. **Note**: Use $\varepsilon = |f_{approx} - f_{exact}|$ as the absolute error (i.e. do NOT compute the relative error here).

c. If you did the above computations correctly, you should have found that the absolute error for $x = x_1$ is roughly 31 times the computed error at $x = x_2$. Is this result consistent with your understanding of truncation error? In particular, clearly explain/rationalize this result based on a formal error analysis for the truncated Taylor series.

Problem 7 Solution Methods for Finding Real Roots of a Nonlinear Equation (15 points)

a. Explain the basic difference between **Bracketing** and **Open Methods** for finding real roots of nonlinear equations. Discuss the key difference in the basic solution algorithms and the relative robustness and efficiency of these two classes of methods. Also briefly discuss any advantages that a hybrid method (such as built into Matlab's **fzero** function) may offer for general applications.

b. Write a complete algorithm (including a convergence check) for the **Newton Method** for finding the real roots of a single nonlinear equation. This does not need to be a program! In fact, a general outline with a brief description of each step is preferable. Be sure to explain in detail the logic behind the choice for a new estimate of the root (recall that the desired formula was formally derived from a truncated Taylor series expansion -- derive and explain this development). Be as explicit as possible. This should be a complete algorithm!

c. What root finding method is usually faster, the **Secant Method** or the **Newton Method**? Explain your choice.

Problem 8 Solution Methods for Linear and Nonlinear Algebraic Equations (10 points)

a. Briefly explain the basic idea behind the ***LU Decomposition Method*** for solving linear equations. Be sure to identify/justify the key steps. Use a 3×3 system to illustrate the key concepts. Be as explicit as possible…

b. First precisely define the term **diagonal dominance** and then determine if the following matrix is diagonally dominant or not (be sure to justify your result):

$$A = \begin{bmatrix} -5 & 3 & 1 \\ 4 & 5 & 2 \\ -2 & 1 & 4 \end{bmatrix}$$

Finally, do you think the Gauss Seidel iterative method would work for a system of three equations with this coefficient matrix? Again be sure to explain/justify your answer…