MAE 107
Assignment 5
Due 10:00pm, Saturday, 27 May

Note: You must show all your work (including your codes) in order to get credit!

Problems to hand in (Not all problems may be graded.)

1. Using the method indicated in class for tridiagonal systems (converting the augmented system to an augmented upper triangular form, and then obtaining x through back-substitution), **by hand** solve $Ax = b$ where
$$A = \begin{pmatrix} 2 & 1 & 0 & 0 \\ -2 & 2 & -2 & 0 \\ 0 & -6 & 5 & 3 \\ 0 & 0 & 1 & 7 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 5 \\ 4 \\ -8 \\ 36 \end{pmatrix}.$$
Further, how many multiplications/divisions and additions/subtractions were required in obtaining the upper triangular form in the augmented form? How many multiplications/divisions and additions/subtractions were required in going from the upper triangular form to the solution of the system? Show all you steps!

2. Based on your answers from the previous problem, estimate the number of multiplications/divisions and additions/subtractions required for each of the two substeps (i.e., obtaining the augmented upper triangular form, and then obtaining x through back-substitution) in solution of $Ax = b$ given the following problem data. (Do not actually solve $Ax = b$ for this data.)
$$A = \begin{bmatrix} 4 & -2 & 0 & 0 & 0 & 0 \\ 2 & 7 & -1 & 0 & 0 & 0 \\ 0 & -2 & -5 & -1 & 0 & 0 \\ 0 & 0 & 2 & 9 & -3 & 0 \\ 0 & 0 & 0 & -1 & -8 & -2 \\ 0 & 0 & 0 & 0 & 1 & 7 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} -1 \\ 1 \\ 9 \\ 11 \\ -9 \\ -11 \end{bmatrix}.$$

3. Write matlab code for the bisection algorithm, and apply it to find a

root of

$$g(x) = \min\left\{\, 0.5 + \sin(\pi x) + 3x,\ 5 - 2x\exp[1/(1+x^2)] \,\right\}.$$

Your code should take the form of a function with a_0, b_0 and ε as inputs. You are welcome to hardwire the function, g, if you like. The code will need to determine the number of steps based on the inputs. Your code does not need to include error checks for potentially bad input data. Apply the code with the inputs $a_0 = 0$, $b_0 = 4$ and $\varepsilon = 10^{-5}$.

4. By hand, apply four steps of Newton's method to the problem of finding the root of $f = xe^x - (x^2 + 1)$, starting from $x_0 = 0$.

5. Consider attempting to solve

$$\arctan(x + 2x^3/3) = 0.7$$

via Newton's method. Convert this problem into TWO DIFFERENT root finding problems. For the first, say $f(x) = 0$, indicate a starting condition such that Newton's method converges to the correct solution. For the second, say $\hat{f}(x) = 0$, indicate a starting condition such that Newton's method does not converge. Demonstrate your solutions with at least five steps of the method in each case; you can do this either by hand or with matlab code, but you must show your work. In the case where the algorithm converges, continue the method until $|f(x_n)| < 10^{-12}$. Also in that case, discuss the errors in the x_n iterates in the context of the digit-doubling concept.

6. Write matlab code for the secant method. The code should be structured as a function that takes as inputs two initial guesses, x_0 and x_1, and a fixed number, n, of steps for which the method will run. You do not need to input a convergence criterion in your code; the code should simply run for the indicated number of steps. That is, if $n = 8$, the code will simply stop after computing x_8. The output should be consist of a vector containing x_0 through x_8 and another vector containing the values of the function at those points. Employ your code to find a root of $f(x) = \sin(x) + (5/4)x - 2$, starting from $x_0 = 3$ and $x_1 = 2$, continuing on to obtain x_8 and $f(x_8)$. Draw a graph illustrating how x_2 was obtained from x_0 and x_1 and, similarly, x_3 from x_1 and x_2. Also discuss the rate of convergence in the light of the digit-doubling aspect of Newton's method.

7. Consider attempting to solve $\tan(4x) = 2x + 1/2$ via the fixed point method. Convert this into a form for which the method is guaranteed to converge, and prove that it will converge. Also find a form for which the method is not guaranteed to converge, and indicate how the guaranteed convergence criterion is not met.

8. Apply the fixed-point method to the two forms you obtained in the previous problem, starting from $x_0 = 0$, $x_0 = 2$ and $x_0 = 1000$. Run the method until either it's obviously diverging, or until succeeding iterates do not change by more than 10^{-5}. You may do this using either code or a calculator, and in the former case, you do not need to comment your code. However, you must show all your steps.

Problems 3, 5, 6 and 7 are worth 10 points each. Problems 1, 2, 4 and 8 are worth 5 points each.

Study Problems (Will not be graded.)

- Suppose you are using the bisection method to find a root of $f(x)$. At the n^{th} step, you know that the root is between a_n and b_n. Given this information, one estimate of the root is $\bar{x}_{n+1} = c_{n+1} = (a_n + b_n)/2$. Assume that there is only one root between a_n and b_n. Show that $\bar{x}_{n+1}$ is the estimate that minimizes the worst-case (maximum) error in the location of the root.

- Suppose you will use the bisection method to find a root, $\bar{x}$, of $f(x)$ where you know $f(1) = 3.3$ and $f(6) = -11.1$. Estimate the minimal values of n which guarantee that $|c_n - \bar{x}| \leq 10^{-3}$ and that $|c_n - \bar{x}| \leq 10^{-6}$.

- Consider the root-finding task of Problem 6. The computational cost of $\sin(x), \cos(x), \tan(x), \exp(x)$ etc. is typically much greater than that of a single FLOP. Assuming that the computational cost of computing either $f(x)$ or $f'(x)$ is roughly that of computing a sine or cosine, and that both are roughly equal, compare the efficiency of Newton's method to that of secant method on that problem. More specifically, suppose that you apply the secant method starting from the x_0 and x_1 given in Problem 6, and apply Newton's method starting from that same x_0. Suppose you apply each one until you obtain an x_n such that $|f(x_n)| \leq 10^{-8}$. Compare the computational cost of the two approaches in terms of the number of evaluations of sines and/or cosines.

- Suppose you are using the fixed-point method to find a fixed point of some function, $f(x)$, and that you know $|f'(x)| < 0.5$ for all x. Let $\bar{x}$ denote the (unknown) fixed point. Suppose, that starting from $x_0 = 40$, you obtain $x_1 = 44.44$. What is the smallest n such that you can guarantee $e_n = |x_n - \bar{x}| \leq 5 \times 10^{-}4$?