Midterm 1
APPM/MATH 4650 Fall '20 Numerical Analysis

Due date: Saturday, September 26, before 3 PM, via Gradescope and Canvas. **Instructor**: Prof. Becker

Instructions There are **two components** to this midterm, with separate rules:

50 points The online Canvas "quiz" which is true/false or multiple choice. You have up to **45 minutes** to complete this. Please do this *before* doing the second component of the test. This component of the test is **closed note, closed book, closed computer/calculator/phone**, meaning that you should not use any resource other than your mind and scratch paper.

100 points The written part (with questions listed below on this document). You have up to **2 hours** to complete this. This component of the test is **open note**, and you **can use the Burden and Faires textbook** (9th or 10th edition), and you can **use Matlab/python or a calculator for simple things** (i.e., for programming things from scratch, and using low-level functionality like core Numpy routines). You may *not* use Matlab/python's builtin root-finders, for example. You **cannot use the internet** other than for uploading to Gradescope, or checking Canvas/Piazza, or connecting to `colab` or something similar. In particular, you may **not use wikipedia or stackexchange websites** with the exceptions of looking up trigonometric identities.

This exam only works if you follow the CU Honor Code. Violating the rules of the exam are simply not fair to your fellow students. Do not discuss any aspect of this exam with other students until after 3 PM Saturday.

Have questions? Please ask on Piazza (and use your judgment about whether to make it a private or public post). Prof. Becker will be actively answering questions from 3–5 PM Friday. After that, he will check for questions infrequently, though TAs may occasionally check too.

On neither portion of the exam are you allowed to use a symbolic math program (graphing calculator, Mathematica, Maple, Desmos, Sage, Wolfram Alpha, Matlab/Python with symbolic packages, etc.). You *can* use a calculator if you want. You *can* write your answers on a tablet if you like (alternatively, write on paper and take a picture or scan it).

Problem 1: **[25 points] Fixed-point equations** The logistic differential equation $f'(t) = r \cdot f(t) \cdot (1 - f(t))$ is sometimes used to model population growth as a function of time, where the population grows exponentially at first but eventually reaches a carrying capacity; for example, the number of people in the country who are infected with a disease. We'll consider a finite-time approximation

$$(\forall n = 1, 2, 3, \ldots) \quad x_{n+1} = r \cdot x_n \cdot (1 - x_n) \tag{1}$$

(for some constant r). Suppose $0 \leq r < 1$. *Prove* that for any starting point $x_0 \in [0, 1]$ that the sequence defined by Eq. (1) converges to a unique limit.

Problem 2: **[25 points] Approximation, stability and conditioning.** Consider trying to evaluate $f(x) = \cos(x) - 1$ when $x \approx 0$.

a) I type `cos(2e-8)-1` into Matlab/Python and it says the answer is `-2.22e-16`. Use a theorem you learned in calculus to prove that this is incorrect.

b) What is the relative condition number of f as $x \to 0$?

c) Interpret your result about the condition number: is mathematical problem sensitive to small changes in the input? Is there any hope to compute the answer accurately in floating point?

d) Describe a better way to calculate $f(x)$.

Problem 3: **[25 points] Root-finding** Consider approximating $\sqrt{2}$ by solving the root-finding problem for $f(x) = x^2 - 2$.

a) Run *one* iteration of Newton's method on f starting at $x = 1$. Calculate this by hand and show your work.

b) Run *three* iterations of the bisection method starting with the interval $[1, 2]$. Calculate this by hand (show the sequence of estimates and intervals) and show your work.

c) How many iterations of the bisection method do we need in order to guarantee an accuracy of $2^{-20} \approx 1^{-6}$?

d) Observe that $h(x) = x^4 - 4x^2 + 4$ also has $\sqrt{2}$ as a root. Would running Newton's method on h work better, worse, or the same as running Newton's method on f? Explain why.

e) What is the *absolute* condition number of this root-finding problem (with $f(x) = x^2 - 2$)?

Problem 4: **[25 points] Interpolation.** Consider the set of nodes $\{0, 1, 2, 3\}$.

a) Draw a rough sketch of the 4th Lagrange polynomial generated by these nodes. Your drawing doesn't have to be perfect, but please make it clear what the degree of the polynomial is, what the values of it are on the nodes, and what will be the limiting behavior as $x \to \pm\infty$.

b) Let p be the minimum degree interpolating polynomial on these nodes if the y-values of the data are $\{1, 2, 3, -4\}$. Evaluate $p(1/2)$ and show your work. You *may* use a calculator or Matlab/Python, but please include the code with your solution then. You may *not* use an existing interpolation package, only use basic functionality. [If you do use Matlab/python, don't worry about including your code in a nice way, a simple screen shot would be fine]

c) One way to do polynomial interpolation on $n + 1$ nodes $\{x_0, x_1, \ldots, x_n\}$ with data $\mathbf{y} = (y_0, y_1, \ldots, y_n)$ is to form the Vandermonde matrix

$$A = \begin{bmatrix} 1 & x_0 & x_0^1 & \ldots & x_0^n \\ 1 & x_1 & x_1^1 & \ldots & x_1^n \\ \vdots & & \ddots & \vdots & \\ 1 & x_n & x_n^1 & \ldots & x_n^n \end{bmatrix}$$

and solve the equation $A\mathbf{c} = \mathbf{y}$ for the coefficients $\mathbf{c} = (c_0, c_1, \ldots, c_n)$ and set $p(x) = c_0 + c_1 x + \ldots x_n x^n$.

Give at least one good reason why this is *not* a good approach to do interpolation.