Homework 2
APPM 4600 Numerical Analysis, Fall 2025

Due date: Friday, September 6, before midnight, via Gradescope. **Instructor**: Prof. Becker
Revision date: 9/1/2025

Theme: Convergence rates, Horner's rule, and warmup for root-finding

Instructions Collaboration with your fellow students is OK and in fact recommended, although direct copying is not allowed. The internet is allowed for basic tasks (e.g., looking up definitions on wikipedia) but it is not permissible to search for proofs or to *post* requests for help on forums such as `http://math.stackexchange.com/` or to look at solution manuals. Please write down the names of the students that you worked with. Please also follow our AI policy.

An arbitrary subset of these questions will be graded.

Turn in a PDF (either scanned handwritten work, or typed, or a combination of both) to **Gradescope**, using the link to Gradescope from our Canvas page. Gradescope recommends a few apps for scanning from your phone; see the Gradescope HW submission guide.

We will primarily grade your written work, and computer source code is *not* necessary (and you can use any language you want). You may include it at the end of your homework if you wish (sometimes the graders might look at it, but not always; it will be a bit easier to give partial credit if you include your code). For nicely exporting code to a PDF, see the APPM 4600 HW submission guide FAQ.

Problem 1: Consider the polynomial

$$p(x) = (x-2)^9 = x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512.$$

Note: you can get these coefficients using Matlab's `poly` or Python's `numpy.poly` and specifying that 2 is a root with multiplicity 9; this saves you having to type them in.

a) Produce a plot that shows the evaluation of p for 20 equispaced points in the range $[1.92, 2.08]$, using the following 4 algorithms for evaluating a polynomial (overlay all 4 results on the same plot, and label the plot nicely):
   i. Make your own algorithm to naively evaluate the polynomial using its coefficients
   ii. Use the compact form $p(x) = (x-2)^9$
   iii. Implement Horner's rule
   iv. Use a software library (e.g., Matlab's `polyval` or Python's `numpy.polyval`).

b) Comment on the similarities and differences, and discuss which algorithm you think is most correct, and what possible sources of numerical error might be.

Problem 2: Convergence Rates.

a) Misleading Taylor Series. To find the convergence rate of a function f, often we use Taylor series. For example, if we say $(1+x)^n = 1 + nx + O(x^2)$ as $x \to 0$, the remainder term $O(x^2)$ was determined from the Taylor series $(1+x)^n = 1 + nx + n(n-1)x^2/2! + \ldots$. However, we need to be careful about the radius of convergence of the Taylor series and that the higher-order derivatives are bounded. Show that e^x is *not* $O(x)$ for $x \to \infty$.

b) Show that $x \sin\sqrt{x} = \theta(x^{3/2})$ as $x \to 0$.

c) Show that $e^{-t} = o(\frac{1}{t^2})$ as $t \to \infty$. (This is "little-o" notation).

d) Show that $\int_0^\varepsilon e^{-x^2} dx = O(\varepsilon)$ as $\varepsilon \to 0$. *Hint*: one nice way to do this uses the Mean Value Theorem.

e) Show that $-x/\log(x)$ is $o(x)$ but not $O(x^2)$ as $x \to 0$.

Problem 3: Consider the equation $2x - 1 = \sin x$.

a) Find a closed interval $[a, b]$ on which the equation has a root r, and use the Intermediate Value Theorem to prove that r exists.

b) Prove that r from (a) is the only root of the equation (on all of $\mathbb{R}$).