MATH 3351-1 **Numerical Analysis I** **Fall 2025**

Homework 2

Due Date: September 9

Grace Etzel

Complete the assignment in Google Colab and typeset your wite-up in LaTeX. Show all work and code, and provide clear justifications for your answers. Submissions that are messy or poorly organized may be returned with a grade of zero.

Link to Google Colab: Link

Problem 1. Determine the height reached by 1 cubic meter of water in a spherical tank of radius 1 meter by completing the following steps:

(a) Show that the volumn of a spherical cap of height h is given by

$$V(h) = \frac{\pi h^2}{3}(3R - h), \quad 0 \leq h \leq 2R,$$

where R is the radius of the sphere.

Proof. We are interested in finding the volume of a spherical hat. The radius of the hat will depend on the height. If we center the cap at $(0, 0)$ coordinate plane, we will travel from $R-h$ to R, where R represents the top of the hat, and $R-h$ represents the distance from the height of the hat to its radius. So, we can find the volume through an integral. Let $x^2 = r^2 - y^2$ denote the equation for a sphere. Then, we can represent the area of the cap (a section of the sphere) as

$$\begin{aligned}
V(h) &= \int_{R-h}^{R} \pi x^2 dy \\
&= \int_{R-h}^{R} \pi (r^2 - y^2) dy \\
&= \pi \left(r^2 y - \frac{h^3}{3} \right) \Bigg|_{R-h}^{R} \\
&= \pi \left(\frac{2}{3}R^3 - R^3 + R^2 h + \frac{R^3 - 3R^2 h + 3Rh^2 - h^3}{3} \right) \\
&= \pi \left(\frac{2R^3 - 3R^3 + 3R^2 h + R^3 - 3R^2 h + 3Rh^2 - h^3}{3} \right) \\
&= \pi \left(\frac{3Rh^2 - h^3}{3} \right) \\
&= \frac{\pi h^2}{3}(3R - h).
\end{aligned}$$

□

(b) In a Google Colab notebook, plot $V(h)$ vs. h for $0 \leq h \leq 2R$ to illustrate how volume depends on height. Clearly label axes (height in meters, volume in cubic meter) and mark the line $V = 1$.

(c) Solve for the height h when $V(h) = 1m^3$ with $R = 1$ using each of the following numerical methods: bisection method, fixed-point iteration, Newton's method, secant method, and a numerical solver from the SciPy library (e.g., scipy.optimize.brentq). Implement each in its own code cell and report the numerical answer and number of iterations; verify the methods agree to a reasonable tolerance (e.g.,10^{-8}).

(d) Estimate the convergence rate of each method by analyzing the sequence of errors (or successive approximations) and comparing their decay. Report the observed rate of convergence for the bisection method, fixed-point iteration, Newton's method, and the secant method.

Problem 2. Implement the following methods for calculating $2^{\frac{1}{4}}$. Rank them for speed of convergence, from fastest to slowest. Be sure to give reasons for your ranking.

(a) A. Bisection method applied to $f(x) = x^4 - 2$

(b) B. Secant method applied to $f(x) = x^4 - 2$

(c) C. Newton's method applied to $f(x) = x^4 - 2$

(d) D. Fixed point iteration applied to $g(x) = \frac{x}{2} + \frac{1}{x^3}$

(e) E. Fixed point iteration applied to $g(x) = \frac{2x}{3} + \frac{2}{3x^3}$

Solution In this problem, we rank the convergence as: C,B,E,A,D. Theoretical speaking, we know that C, Newton's method, will converge the fastest, since its order is of the second degree. Next, we can theorize that secant method is second, since it converges faster than a linear rate, which is the speed of fixed point iteration and bisection method. The ranking of E,A,D will depend on many factors such as the initial guess. In this case (look at the code in Google Colab), we find that the fixed point iteration from E converges faster than the bisection method. Note that we put D last, since it fails to converge, even for values extremely close to $2^{\frac{1}{4}}$. To look at the specific convergence rates, we direct you to the Google Colab.

Problem 3. Consider

$$f(x) = \sin(x) + x^2 \cos(x) - x^2 - x,$$

whose root at $r = 0$ has multiplicity $m = 3$ (since $f(0) = f'(0) = f''(0) = 0$ and $f^{(3)} \neq 0$). Use the initial guess $x_0 = 1$.

(a) Run Newton's method and its modified version until $\|x_{i+1} - x_i\|_\infty \leq 10^{-12}$ or a maximum of 50 iterations.

Solution. We can define the modified version of Newton's method as

$$x_{i+1} = x_i - \frac{mf(x_i)}{f'(x_i)}.$$

The code and solution can be view in the Google Colab under Problem 3.

(b) For each method, produce a table with columns

$$i, \quad x_i, \quad e_i = |x_i - r|, \quad \frac{e_i}{e_{i-1}} \quad (i \geq 1)$$

(c) For each method, track the number of correct decimal places at iteration i. Analyze the pattern in the number of correct decimal places in the approximate solutions as the number of iterations increases for both methods.

Solution. First, notice that the for each method the correct decimal places for the errors and the Newton's method estimate are the same. Additionally, both methods have the same correct decimal places for the ratio of errors. Below represents a table that has columns for the correct decimal place, times shown in the modified Newton's method and regular Newton's method:

Notice that both methods differ only at e^{-1} and e^{-13}. In fact, if you switch the first observation in "# of times in Modified Newton's Method" with its last observation, then the two methods have the exact same number of correct decimal places. This may suggest that the modified version helps decrease the errors within the first couple iterations, but the rest of the iterations are the same rate at as the regular version.

Correct Decimal Places	# of times in Modified Newton's Method	# of times in Newton's Method
e^{-1}	3	1
e^{-2}	3	3
e^{-3}	3	3
e^{-4}	3	3
e^{-5}	4	4
e^{-6}	3	3
e^{-7}	3	3
e^{-8}	4	4
e^{-9}	3	3
e^{-10}	3	3
e^{-11}	4	4
e^{-12}	3	3
e^{-13}	1	3

Table 1: Tracking Correct Decimal Places