Remember that the purpose of the exam is to show me how you think about these problems. Accordingly, please show your work, use mathematical notation correctly, and justify your answers. Each problem is worth 5 points. If any problem is unclear, please ask me about it.

1. Floating Point Representation of Real Numbers. Consider a very crude computer that stores its floating point numbers using a total of 5 bits: one for the sign $s$, two for the characteristic $c$, and two for the mantissa $f$. The floating point coding is then given by the formula $(-1)^s \cdot 2^{c-1} \cdot (1+f)$.

   a) On the number line below, sketch all the numbers that can be represented on this machine within the interval $[1, 8]$.

   ![Number Line]

   b) What is the largest number less than 1?

   For $x_1, C=1, S=0$.

   For $\text{min.}(0, 0.125)$: $C=0, F=\frac{1}{2}, \frac{1}{4} = \frac{5}{8}$.

   $\boxed{x_2 = \frac{5}{8} \cdot (1+\frac{3}{4}) = \frac{1}{2} \cdot \frac{7}{4} = \frac{7}{8}}$

   c) Let $r$ be a real number in the range $(1, 2)$. What is the largest possible round off error if you store $r$ on this machine, assuming it gets stored as the closest possible floating point number?

   Floating pt. $4$s have spacing $\frac{1}{14}$ in $(1, 2]$. If $r$ is closer to $1$, the round off error is $\frac{1}{2}$ assuming $r = x$.

   d) Now consider a 64-bit computer. What number is represented by the following?

   $010000000011 \ldots$

   $C = 2^{16} \cdot 2^{11} = 1027$

   $F = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$

   $L = 1027 - 1023 = 2$

   $\ell_3 = 2 \cdot (1 + \frac{3}{4}) = 2 \cdot \frac{7}{4} = \boxed{\frac{7}{2}}$

   e) Suppose you tried to store $\pi$ as a floating point number on a 64-bit computer. Approximately how big will the round-off error be? Justify your answer.

   Mantissa error: $\approx \frac{1}{2}$

   Thanks. For $\pi$, the exponent is $2^4$.

   So for full error: $\leq 2 + 2^4 = \boxed{2^{10}}$
2. **Numerical Differentiation.** Suppose your goal is to use a numerical method for approximating the derivative of a function \( f(x) \) at a point \( x_0 \).

a) Write down formulas for the forward, backward, and symmetric difference approximations for \( f'(x_0) \).

   - Forward diff. approx.: \[ \frac{f(x_0 + h) - f(x_0)}{h} \]
   - Backward diff. approx.: \[ \frac{f(x_0) - f(x_0 - h)}{h} \]
   - Symmetric diff. approx.: \[ \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2} \]

b) Use a first order Taylor series of \( f(x) \) around the base point \( x_0 \) to express \( f(x_0 + h) \). Make sure to include the error term.

\[ f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{f''(\xi)}{2}h^2 \]

   - Error term

\[ \frac{f''(\xi)}{2}h^2 \]

\[ f''(\xi) \text{ is } O(h) \]

c) Use the above Taylor series to show that the error of the forward difference approximation is \( O(h) \). Dividing both sides by \( h \) and decreasing \( h \) will:

\[ \frac{f(x_0 + h) - f(x_0)}{h} = f'(x_0) + \frac{f''(\xi)}{2h} \]

\[ \Rightarrow \left| \frac{f(x_0 + h) - f(x_0)}{h} - f'(x_0) \right| \leq \frac{1}{2} |f''(\xi)|h \]

So error is \( O(h) \).

d) Suppose you use a forward difference approximation with step size \( h = 0.1 \) to estimate \( f'(x_0) \), where \( f(x) = 5x^2 \) and \( x_0 = 2 \). Using your work above, derive a firm bound on the approximation error.

\[ f(x) = 5x^2 \]

\[ f''(x) = 10x \]

\[ [2, 2.1] \quad 15 \leq 10(2.1) = 63 \]

\[ \text{So error } \leq \frac{63 \cdot 0.1}{2} = \frac{6.3}{2} = 3.15 \]

\[ \boxed{3.15} \]

e) Finally, suppose that your computer has a round off-error of \( \epsilon = 10^{-6} \) and you use a forward difference scheme to approximate \( f'(x_0) \), where \( f(x) = 5x^2 \) and \( x_0 = 2 \). Calculate the optimal value of the step size, \( h \).

\[ f'(x) = \frac{f(x_0 + h) - f(x_0)}{h} \]

\[ \frac{\text{error}}{h} \leq \frac{2\epsilon}{h^2} + 3.15h \]

\[ \frac{\dot{\text{error}}}{h} = -\frac{2\epsilon}{h} + 3.15 = 0 \Rightarrow 3.15 = \frac{2\epsilon}{h^2} \]

\[ \Rightarrow h^2 = \frac{2\epsilon}{3.15} \Rightarrow h = \sqrt{\frac{2\epsilon}{3.15}} \]
3. Root finding in one variable.

a) Write down the general formula for the Newton update step in one dimension.

\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \]

b) Consider the function \( f(x) = x^2 - 1 \). (Note that this function has a root at \( x = 1 \).) Suppose your initial guess for the root is \( x_0 = 2 \). Take two Newton steps, and calculate the errors at each step. Are they decreasing?

\[ x_1 = 2 - \frac{3}{4} = \frac{5}{4} \]
\[ x_2 = \frac{1}{4} - \frac{1}{2 \left[ \frac{5}{4} \right]} = \frac{5}{4} - \frac{5 - \frac{9}{4}}{16} \]
\[ = \frac{5}{4} - \frac{5 - \frac{9}{4}}{16} = \frac{5}{4} - \frac{11}{40} = \frac{20}{40} - \frac{11}{40} = \frac{9}{40} \]

Yes, decreasing.

c) Write down what it means for a sequence \( a_n \) to converge to \( a \) quadratically. Do your results above suggest that the Newton errors are decreasing quadratically?

\[ a_n \to a \text{ quarter } \]
\[ L = \frac{|a_{n+1} - a|}{|a_n - a|^2} \]

error sequence: \( \frac{1}{4}, \frac{1}{40} \), yes, suggests at least quadratic convergence.

d) How can you cast this problem as a fixed point problem? Write down a function \( h(v) \) whose fixed point will be a root of \( f(x) \). Would you expect fixed point iterations of your \( h(v) \) to converge, assuming you started at the point \( x_0 = 2 \)? Explain why or why not.

Set \[ h(v) = v - \frac{v^2 - 1}{2v} \]

\[ h'(v) = 1 - 2v \]

\[ \frac{|h'(v)|}{2} = \frac{|1 - 2v|}{2} \leq 1 \]

So no, we will not expect fixed point iterations to converge.

e) Finally, suppose you wished to find the value of \( x \) that minimized \( f(x) \). How can you cast this optimization problem as a root finding problem? Write the root-finding problem you would need to solve, and take one Newton step, assuming \( x_0 = 2 \). How did the method perform? Are you surprised?

\[ f'(x) = 2x \to \text{ solve for root } x \]

\[ x_1 = 2 - \frac{2}{2} = 2 - 1 = 1 \]

Minimize \( f(x) \) at \( x = 1 \), so problem was good.

Expected, since \( f'(x) \) linear.
4. Root finding in two variables. Consider the problem of finding solutions to the system

\[ f(x, y) = 0 \]
\[ g(x, y) = 0. \]

a) Suppose \((x_0, y_0)\) is a starting guess. Recall that the Newton update step can be written in the form \((x_1, y_1) = A^{-1}B\). What are \(A\), \(B\), and \(M\)?

\[
A = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad B = \begin{bmatrix} f(x_0, y_0) \\ g(x_0, y_0) \end{bmatrix}, \quad M = \begin{bmatrix} \frac{2f}{x} & \frac{2f}{y} \\ \frac{2g}{x} & \frac{2g}{y} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}.
\]

b) One way systems materialize in applications is through optimization. Suppose your goal is to minimize the function \(f(x, y) = x^4 + 2x^2y^2 + y^4\). Write down the system of equations whose roots would be candidates for this minimum.

\[
\begin{align*}
\frac{df}{dx} &= 4x^3 + 4x y^2 = 0, \\
\frac{df}{dy} &= 4x^2 y + 4y^3 = 0.
\end{align*}
\]

\[\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \end{bmatrix} \quad \begin{bmatrix} 1 & 2 y \\ x & 1 \end{bmatrix} \begin{bmatrix} x-x_0 \\ y-y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\]

c) Another way these systems materialize in practice is through solving systems of ODEs with implicit schemes. Suppose you have an IVP of the form \(V' = F(V)\), with \(V_0 = (1, 2)\), where \(V\) is a vector \((x, y)\) and \(F(V) = F(x, y) = (x - xy, xy - y)\). Write down the system of equations you would need to solve to propagate \(V\) forward one time step of size \(\Delta t = 0.1\) with a standard implicit scheme.

\[
\begin{align*}
\dot{x} &= x - x y - \Delta t (x - x_0), \\
\dot{y} &= x y - y - \Delta t (y - y_0).
\end{align*}
\]

d) Finally, remember that the relationship between root finding and fixed point solutions persists in multiple dimensions. Suppose you wish to find a root of the system

\[ f(x, y) = x^2 + y^2 - 2y \]
\[ g(x, y) = x + y + 1 \]

Cast this problem as a fixed point problem, i.e. write down a vector-valued function \(H(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}^2\) such that any fixed point of \(H\) is a root of both \(f\) and \(g\).

\[
H(x, y) = \begin{bmatrix} x^2 + y^2 - 2y + x \\ x + y + 1 \end{bmatrix}.
\]