1. Explain why the equation $e^{-x} = x$ has a solution on the interval $[0, 1]$. Use bisection to find the root to 4 decimal places. Can you prove that there are no other roots?

**Solution:** If $f(x) = e^{-x} - x$, then $f(0) = 1$, $f(1) = 1/e - 1 < 0$, and hence a root is guaranteed by the Intermediate Value Theorem. Using Bisection, the value of the root is $x^* = .5671$. Since $f'(x) = -e^{-x} - 1 < 0$ for all $x$, the function is strictly decreasing, and so its graph can only cross the $x$ axis at a single point, which is the root.

2. Find $\sqrt[3]{3}$ to 5 decimal places by setting up an appropriate equation and solving using bisection.

**Solution:** Using the equation $u^6 - 3 = 0$ and the initial interval $a = 1$, $b = 2$, bisection produces $\sqrt[3]{3} \approx 1.200937$ in 21 iterations.

3. Find all real roots of the polynomial $x^5 - 3x^2 + 1$ to 4 decimal places using bisection.

**Solution:** $x_1 = -56107$, $x_2 = 599241$, $x_3 = 1.34805$.

4. Let $g(u)$ have a fixed point $u^*$ in the interval $[0, 1]$, with $g'(u^*) \neq 1$. Define

$$G(u) = \frac{ug'(u) - g(u)}{g'(u) - 1}.$$ 

(a) Prove that, for an initial guess $u^{(0)}$ near $u^*$, the fixed point iteration scheme $u^{(n+1)} = G(u^{(n)})$ converges to the fixed point $u$. (b) What is the order of convergence of this method? (c) Test this method on the non-convergent cubic scheme in Example 2.16.
Solution:

(a) First, if \( g(u^*) = u^* \), then

\[
G(u^*) = \frac{u^* g'(u^*) - g(u^*)}{g'(u^*) - 1} = u^* \frac{u^* g'(u^*) - g(u^*)}{g'(u^*) - 1} = u^*,
\]

and hence \( u^* \) is also a fixed point of \( G \). Moreover,

\[
G'(u) = \frac{g''(u)(g(u) - u)}{(g'(u) - 1)^2}, \quad \text{and so} \quad G'(u^*) = 0.
\]

Thus, the convergence is quadratic.

(b) In this case \( G(u) = \frac{2u^3 + 1}{3u^2 - 1} \). Starting with an initial guess of \( u^{(0)} = 1 \), the iterates converge quadratically fast to the fixed point, producing \( u^* = 1.324717957245 \) to 12 decimal places after only 5 iterations.

5. Let \( g(u) = 1 + u - \frac{1}{8} u^3 \). (a) Find all fixed points of \( g(u) \). (b) Does fixed point iteration converge? If so, to which fixed point(s)? What is the rate of convergence? (c) Predict how many iterates will be needed to get the fixed point accurate to 4 decimal places starting with the initial guess \( u^{(0)} = 1 \). (d) Check your prediction by performing the iteration.

Solution:

(a) The only fixed point is \( u^* = 2 \). Since \( g'(2) = -\frac{1}{2} \), the fixed point is stable, and the convergence is linear at a rate of \( \frac{1}{2} \).

(b) If the convergence rate were exactly \( \frac{1}{2} \), since our initial error is 1, to obtain an error of \( .5 \times 10^{-4} \), we would expect to need \( -\log_2(.5 \times 10^{-4}) = -14.2877 \) iterations.

(c) In fact, \( u^{(13)} = 1.99997393 \), which is already correct to 4 decimal places.


Solution: Same solutions; faster convergence.

7. (a) Let \( u^* \) be a simple root of \( f(u) = 0 \). Discuss the rate of convergence of the iterative method (sometimes known as Olver’s Method, in honor of the author’s father) based on \( g(u) = u - \frac{f(u)^2 f''(u) + 2 f(u) f'(u)^2}{2 f'(u)^3} \) to \( u^* \). (b) Try this method on the equation in Exercise 3, and compare the speed of convergence with that of Newton’s Method.
Solution:

(a) Clearly, if \( f(u^*) = 0 \), then \( g(u^*) = u^* \). Moreover,

\[
g'(u) = f(u)^2 \left( \frac{3 f''(u)^2 - f'(u) f^{(3)}(u)}{2 f'(u)^4} \right),
\]

\[
g''(u) = f(u) \left( \frac{6 f'(u)^2 f''(u)^2 - 12 f'(u) f''(u)^3 - 2 f'(u)^3 f^{(3)}(u)}{2 f'(u)^5} + 9 f(u) f'(u)^2 f''(u) f^{(3)}(u) - f(u) f'(u)^2 f^{(4)}(u)} \right),
\]

and hence \( g'(u^*) = g''(u^*) = 0 \). Thus, Olver’s method has (at least) third order convergence: \( |e^{(k+1)}| \leq \tau |e^{(k)}|^3 \). In fact, the order is equal to three provided \( g'''(u^*) = \frac{3 f''(u^*)^2 - f'(u^*) f^{(3)}(u^*)}{f'(u^*)^2} \neq 0 \).

(b) In all cases, to obtain 15 decimal place accuracy of the roots, starting with \(-1, .5, 2\), Newton requires, respectively, 6, 4, 7 iterates, while Olver requires 4, 3, 5 iterates to converge to the closest root.