Assignment Problems

Cosmic Charade

Part 1: Solving Equations - Root Finding Methods

Problem 1: Bisection Method

Use the **Bisection Method** to find the root of the function f(x) = x^3 - 4x^2 + 3x - 2f within the interval $[1, 2]$ to an accuracy (tolerance) of 10^{-5}.

- **Hint**: Implement the Bisection method as a Python function and use it to compute the root.
- **Deliverable**: Code implementation and the computed root.

Problem 2: Newton-Raphson Method

Solve the equation $f(x) = \cos(x) - x$ using the **Newton-Raphson method**. Start with an initial guess of $x_0 = 0.5$, and iterate until the solution converges within a tolerance of 10^{-5}.

- **Hint**: Implement the Newton-Raphson method. Use the derivative $f'(x) = -\sin(x) - 1$.
- **Deliverable**: Code implementation and the computed root.

Part 2: Numerical Integration

Problem 3: Trapezoidal Rule

Use the **Trapezoidal Rule** to numerically integrate the function $f(x) = e^{-x^2}$ over the interval $[0, 1]$ using 100 subintervals.

- **Hint**: Implement the Trapezoidal Rule in Python.
- **Deliverable**: Code implementation, the computed integral value, and a brief explanation of the accuracy of the method.

Problem 4: Simpson's Rule

Apply **Simpson's Rule** to integrate the function $f(x) = \ln(x + 1)$ over the interval $[0, 2]$. Use 100 subintervals (ensure n is even).

- **Hint**: Implement Simpson's Rule as a Python function.
- **Deliverable**: Code implementation, the computed integral, and a comparison with the exact integral value (if known).

Part 3: Differentiation - Finite Differences

Problem 5: Forward Difference Approximation

Use the **Forward Difference Approximation** to estimate the derivative of the function $f(x) = \sin(x)$ at $x = \pi/4$, using $h = 0.01$.

- **Hint**: Use the formula $f'(x) \approx \frac{f(x+h)-f(x)}{h}$.
- **Deliverable**: Code implementation and the computed derivative value. Compare the result with the exact derivative.

Problem 6: Central Difference Approximation

Use the **Central Difference Approximation** to compute the derivative of $f(x) = x^3 - x^2 + 2x$ at $x = 1$, using $h = 0.01$.

- **Hint**: Use the formula $f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$.
- **Deliverable**: Code implementation, computed derivative value, and comparison with the exact derivative.

Part 4: Monte Carlo Integration

Problem 7: Monte Carlo Integration

Use **Monte Carlo Integration** to approximate the value of the integral $\int_0^1 e^{-x^2} dx$ by generating \$10,000 random samples from a uniform distribution in the interval $[0, 1]$.

- **Hint**: Implement Monte Carlo Integration by averaging the values of e^{-x^2} at randomly chosen points.
- **Deliverable**: Code implementation, the estimated integral value, and comparison with the exact integral.

Problem 8: Monte Carlo Area Estimation

Estimate the area of the unit circle $x^2 + y^2 \leq 1$ using **Monte Carlo Integration**. Generate $10,000$ random points in the square $[-1, 1] \times [-1, 1]$ and determine the proportion of points inside the circle.

- **Hint**: Use random sampling to estimate the area.
- **Deliverable**: Code implementation, estimated area, and comparison with the actual value of π.

Submission Checklist:

1. Python code for each problem.
2. Brief explanations of the results, including comparisons with known values (if applicable).
3. Visualization or plots (if relevant).

All the Best