Experiment No.: 04
Experiment Name: Implementation of Newton-Raphson method.

Theory:

The Newton-Raphson Method is a fast and commonly used numerical approach for finding the roots of real-valued functions. Unlike bracketing methods such as Bisection or False Position, it is an open method, meaning it only requires a single initial guess that is reasonably close to the actual root. This technique utilizes the slope of the tangent line to the curve of the function to refine the estimate of the root. Starting with an initial value x0, the next approximation is calculated using the formula:

$$X_{i-1} = X_i - \frac{f(xi)}{f'(x)}$$

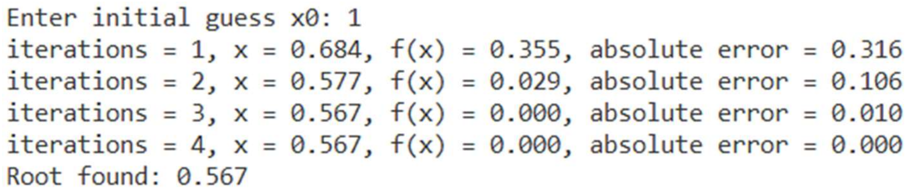
Here, f'(xi) represents the derivative of the function at xi. The process is repeated until the difference between consecutive estimates becomes smaller than the desired tolerance.

When the starting guess is close to the actual root and the function is well-behaved (continuous, differentiable, and not flat near the root), the Newton-Raphson method typically converges much faster than bracketing techniques.

Program 1: Programming Code

```python
from math import exp

def f(x):
    return x * exp(x) - 1

def df(x):
    return exp(x) * (x + 1)

def newton_raphson(f, df, x0, tol=1e-3, max_iter=100):
    if df(x0) == 0:
        print("Derivative is zero. No solution found.")
        return None

    x = x0
    for i in range(1,max_iter):
        x_new = x - f(x) / df(x)
        ae = abs(x_new - x)
        print(f"iterations = {i}, x = {x_new:.3f}, f(x) = {f(x_new):.3f}, absolute error = {ae:.3f}")
        if ae <= tol:
            return x_new
        x = x_new

    print("Maximum iterations reached. No solution found.")
    return None

# Example usage
x0 = float(input("Enter initial guess x0: "))
root = newton_raphson(f, df, x0)
if root is not None:
    print(f"Root found: {root:.3f}")
```

Output:



```
Enter initial guess x0: 1
iterations = 1, x = 0.684, f(x) = 0.355, absolute error = 0.316
iterations = 2, x = 0.577, f(x) = 0.029, absolute error = 0.106
iterations = 3, x = 0.567, f(x) = 0.000, absolute error = 0.010
iterations = 4, x = 0.567, f(x) = 0.000, absolute error = 0.000
Root found: 0.567
```

Discussion & Conclusion:

The Newton-Raphson Method was applied to the function $f(x)=xe^{-x}-1$ with an initial guess of x0=1. The method quickly converged to the root using the tangent-line approximation. Compared to methods like Bisection or False Position, it demonstrated much faster convergence when the initial guess was appropriate.

However, the method depends on the derivative of the function and may fail or diverge if the derivative becomes zero or if the initial guess is far from the actual root. In this case, the approach performed efficiently and produced an accurate result within the specified tolerance.