DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Experiment 1

Student Name: Shruti Raj
UID: 23BCS10685
Branch: CSE
Section/Group: 610-A
Semester: 6th
Date of Performance: 7/1/26
Subject Name: Numerical Methods
Subject Code: 23SMH-341

1. Aim:

To study the nature of numerical errors arising due to truncation and round-off, and to analyze their behaviour using Python.

2. Objective:

To understand truncation and round-off errors and analyze error propagation in floating-point numerical computations.

3. S/W Requirements:

Python 3.11 (Install the latest according to your system).

4. Implementation/Code:

```
import math

print("----- NUMERICAL ERROR ANALYSIS -
----")
```

```
# True value x = 1 true_value =
math.exp(x) print(f"\nTrue value of
e^{x} = {true_value}")

# --- Truncation Error using Maclaurin Series ---
# Maclaurin series of e^x: 1 + x + x^2/2! +
x^3/3! + ... def maclaurin_exp(x, n):
   s = 0.0     for k
in range(n):
       s += x**k / math.factorial(k)
return s

print("\n--- Truncation Error Analysis (Maclaurin Series) --
-") for n in [1, 2, 3, 4, 5, 10]:
   approx = maclaurin_exp(x, n)
trunc_error = abs(true_value - approx)
   print(         f"Terms = {n}, Approx =
{approx:.8f}, "         f"Truncation Error
= {trunc_error:.8f}"
   )

# --- Round-off Error Demonstration ---
print("\n--- Round-off Error Analysis ---")

pi_true = math.pi

pi_3 = round(pi_true, 3)
pi_5 = round(pi_true, 5)
pi_10  =  round(pi_true,
10)

round_values = [pi_3, pi_5, pi_10] labels =
["3 decimal", "5 decimal", "10 decimal"]
```

```
for lbl, val in zip(labels, round_values):
    abs_err = abs(pi_true - val)
rel_err = abs_err / pi_true
print(          f"{lbl}: {val} | "
f"Abs Error = {abs_err:.12f}, "
f"Rel Error = {rel_err:.12e}")

# --- Error Propagation Example --- print("\n---
Error Propagation Example ---") a = 1e16 b = 1.0
c = a - (a - b) print(f"Expected result = 1.0,
Computed result = {c}") print(f"Round-off Error
= {abs(1.0 - c)}")
```

5. Output:

```
----- NUMERICAL ERROR ANALYSIS -----

True value of e^1 = 2.718281828459045

--- Truncation Error Analysis (Maclaurin Series) ---
Terms = 1, Approx = 1.00000000, Truncation Error = 1.71828183
Terms = 2, Approx = 2.00000000, Truncation Error = 0.71828183
Terms = 3, Approx = 2.50000000, Truncation Error = 0.21828183
Terms = 4, Approx = 2.66666667, Truncation Error = 0.05161516
Terms = 5, Approx = 2.70833333, Truncation Error = 0.00994850
Terms = 10, Approx = 2.71828153, Truncation Error = 0.00000030

--- Round-off Error Analysis ---
3 decimal: 3.142 | Abs Error = 0.000407346410, Rel Error = 1.296623894703e-04
5 decimal: 3.14159 | Abs Error = 0.000002653590, Rel Error = 8.446638650626e-07
10 decimal: 3.1415926536 | Abs Error = 0.000000000010, Rel Error = 3.248971946611e-12

--- Error Propagation Example ---
Expected result = 1.0, Computed result = 0.0
Round-off Error = 1.0
```

Figure 1: Error Analysis

6. Learning Outcomes:

- Understand what numerical errors are and why they occur in computer calculations.
- Identify truncation errors caused by using a truncated Maclaurin series.

- Analyze round-off errors due to finite precision in floating-point representation.
- Observe that increasing the number of series terms reduces truncation error.
- Gain practical experience using Python for numerical error analysis.