Lab Manual for Numerical Analysis

Lab No. 9

INTEGRATION - SIMPSON'S THREE-EIGHTH RULE, INTEGRATION WITH UNEQUAL SEGMENTS

Software Engineering Department
Bahria University (Karachi Campus)

BAHRIA UNIVERSITY KARACHI CAMPUS
Department of Software Engineering

NUMERICAL ANALYSIS

LAB EXPERIMENT # 9

Integration - Simpson's three-eighth rule, Integration with unequal segments

OBJECTIVE:

The objective of this lab is to familiarize students with advanced numerical integration techniques, specifically **Simpson's three-eighth** rule and integrating functions with **unequal segment**.

Introduction

As previously discussed, **integration** is a fundamental mathematical operation used to ascertain the accumulation or total value of a quantity within a specified range. It involves calculating the area under a curve or summing smaller parts to derive a complete whole. This mathematical technique holds immense value across various fields including physics, engineering, economics, and others, enabling the analysis of rates of change, area computations, and predictions based on accumulated data.

In our last chapter, we delved into numerical integration methods like the Trapezoidal rule and Simpson's 1/3 rule, which approximate definite integrals by partitioning the area under a curve into simpler shapes like trapezoids or parabolic segments. Building upon this foundation, we will explore **Simpson's 3/8 rule**, another numerical integration technique that offers increased accuracy by employing cubic approximations. Additionally, we'll investigate **integration** with **unequal parts**, extending our understanding of numerical integration to handle scenarios where the segments being integrated are not of uniform size.

Implementation of Simpson's three-eighth rule and Integration with unequal segments

This section aims to extensively cover the principles and Python implementation of the **Simpson's three-eighth rule** and integration with **non-uniform segments** in numerical integration.

a. Simpson's three-eighth rule

Simpson's 3/8 rule is a numerical integration technique used to approximate definite integrals by dividing the interval into multiple segments and employing **cubic polynomial**

approximations. It extends the idea of Simpson's 1/3 rule by using three equally spaced points to fit a cubic polynomial instead of parabolic segments.

The main difference between Simpson's 1/3 rule and Simpson's 3/8 rule lies in the number of points used for approximation. Simpson's 1/3 rule uses **two intervals** and fits **quadratic curves** to these pairs of intervals, while Simpson's 3/8 rule uses **three intervals**, employing **cubic approximations**. Due to this Simpson's 1/3 rule is preferred when dealing with an **even number of intervals** due to its utilization of pairs of intervals, allowing for the application of quadratic curves efficiently. Conversely, Simpson's 3/8 rule becomes advantageous when the number of intervals is a **multiple of three**, permitting the use of cubic approximations across these sets of intervals for improved accuracy in approximation.

The formula for Simpson's 3/8 rule for numerical integration is:

$$\int_a^b f(x)dx \approx \frac{3h}{8}\left[f(a) + 3\left\{\sum_{i=2,5,8,\ldots}^{n-1} f(xi)\right\} + 2\left\{\sum_{j=4,7,10,\ldots}^{n-2} f(xj)\right\} + f(b)\right]$$

Where;

- a and b are the lower and upper limits of integration
- *h* is the width of each interval, calculated as $\boldsymbol{h} = \frac{\boldsymbol{b-a}}{\boldsymbol{n}}$
- f(x) is the function to be integrated
- The summations are used to calculate the function values at specific points within each interval.

Implementation in Python

- The following code provides a function **simpson_three_eight_rule** to approximate the integral using Simpson's 3/8 rule for a given function, represented by $\frac{1}{1+x^2}$

```
# Define the function to be integrated
def function(x):
    return 1 / (1 + x * x)

# Function to calculate the integral using Simpson's 3/8 rule
def simpson_three_eight_rule(a, b, n):
    h = (b - a) / n
    integral = function(a) + function(b)

    # Calculate the sum for the integral within the given limit
    for i in range(1, interval_limit):
        if i % 3 == 0:
            integral += 2 * function(a + i * h)
        else:
            integral += 3 * function(a + i * h)

    # Compute the integral value using Simpson's 3/8 rule formula
    return (3 * h / 8) * integral
```

```
# Define the limits and interval for the integral calculation
interval_limit = 9
lower_limit = 1
upper_limit = 10


# Calculate the integral using Simpson's 3/8 rule
integral_result = simpson_three_eight_rule (lower_limit,
upper_limit, interval_limit)


# Print the rounded integral result
print(round(integral_result, 6))
```

- Meanwhile the following code performs Simpson's 3/8 rule directly on given data points using function **simpson_three_eight_rule_from_values**

```
# Function to calculate the integral using Simpson's 3/8 rule with
data points
def simpson_three_eight_rule_from_values(data_points):
    n = len(data_points) - 1  # Number of intervals

    h = (data_points[1][0] - data_points[0][0])
    integral = data_points[0][1] + data_points[-1][1]


    # Calculate the sum for the integral within the given data points
    for i in range(1, n):
        if i % 3 == 0:
            integral += 2 * data_points[i][1]

        else:
            integral += 3 * data_points[i][1]

    # Compute the integral value using Simpson's 3/8 rule formula
    return (3 * h / 8) * integral


# Define the data points (x, y) for the function
data_points = [(1, 1),(4, 0.2), (7, 0.1), (10, 0.038)]
```

```
# Calculate the integral using Simpson's 3/8 rule with data points
integral_result = simpson_three_eight_rule_from_values (data_points)


# Print the rounded integral result
print(round(integral_result, 6))
```

b. Integration with unequal segments

Integration with **unequal segments** involves approximating the integral of a function when the integration interval is divided into **segments** of **varying sizes** due to changing behavior or varying conditions within the integration range.

While most numerical integration formulas assume equally spaced data points, real-world scenarios often present unequal-sized segments. Experimental data, for instance, frequently exhibits such characteristics. In these cases, employing the trapezoidal rule for individual segments and summing the obtained results offers an approach to approximate the integral, accounting for the diverse segment sizes within the integration range.

The formula of trapezoidal rule with uneven segments/sub-intervals is:

$$\int_a^b f(x)dx \approx h_1 \frac{f(x_0)+f(x_1)}{2} + h_2 \frac{f(x_1)+f(x_2)}{2} + \cdots + h_n \frac{f(x_{n-1})+f(x_n)}{2}$$

Where;

- $h_n = x_n - x_{n-1}$ represents the width of **n^{th} interval**.

Implementation in Python

- The following code provides a function **trapezoidal_rule_for_uneven_subintervals** to approximate the integral using Trapeziodal rule with uneven segments.

```
def trapezoidal_rule_for_uneven_subintervals(data_points):
    n = len(data_points)

    integral = 0.0
    for i in range(len(data_points) - 1):
        # Calculating h from x values
        h = data_points[i + 1][0] - data_points[i][0]

        integral+= 0.5*h*(data_points[i][1]+data_points[i+1][1])

    return integral
```

```
# Define data points (x, y)
data = [(4, 1.386), (4.2, 1.435), (4.4, 1.481),
        (4.6, 1.526), (4.8, 1.568), (5, 1.609),
        (5.2, 1.648)] # Example data points

# Calculate the integral using the Trapezoidal rule with provided
data points
result = trapezoidal_rule_for_uneven_subintervals data)
print("Approximate integral using Trapezoidal rule with provided
data points:", result)
```

Lab Tasks

1. Write a Python program utilizing Simpson's 3/8 Rule to compute the definite integral $\int_0^{0.8} \left(\frac{1}{2} + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5\right) dx$ where n = 3 represents the number of intervals used in the approximation.

   (Output: 1.75917)

2. Write a python program to determine the approximation of the area beneath the curve represented by y = f(x) using Simpson's 3/8. The values of the function f(x) are provided within following table.

x	1.4	1.6	1.8	2	2.2	2.4	2.6
f(x)	4.0552	4.953	6.0436	7.3891	9.025	10.092	11.099

   (Output: 9.02049)

3. Write a python code in table below to determine the integral for this data:

x	**f(x)**
0.0	0.200000
0.12	1.309729
0.22	1.305241
0.32	1.743393
0.36	2.074903
0.40	2.456000
0.44	2.842985
0.54	3.507297
0.64	3.181929
0.70	2.363000
0.80	0.232000

   (Output: 1.5948)