Project Report

Numerical Methods

CSE-4746

Submitted By:

Team Convergence Crew

Al-Fahad:C213001

Mohammad Abrar Yasir:C213002

Md Mokaddess Hossain Adnan:C213007

Mostafa Zamal Zahid: C213035

Section:7AM

Submitted To:

Prof.Mohammad Shamsul Alam

Professor

Department of CSE

IIUC

Date of submission

January 14 ,2025

Project Name: Numerical Integration Simulation using Python

Introduction

This project develops a user-friendly software tool for numerical integration, implementing methods like Simpson's 1/3 Rule, Simpson's 3/8 Rule, Trapezoidal Rule, and Weddle's Rule. Equipped with a graphical user interface (GUI), it enables users to input functions, set integration limits, select methods, and visualize results with intuitive plots. The tool is designed to enhance learning, promote accuracy, and support computational needs in science, engineering, and mathematics.

Background

Numerical integration, also known as numerical quadrature, is indispensable in scenarios where integrals cannot be evaluated symbolically. Traditional methods like antiderivatives often fail when dealing with functions that lack closed-form solutions. This project focuses on four key numerical methods:

- **Simpson’s 1/3 Rule**: A method based on quadratic polynomial interpolation.
- **Simpson’s 3/8 Rule**: A variation of Simpson's rule using cubic polynomial interpolation.
- **Trapezoidal Rule**: A straightforward approach that approximates the area under the curve as a series of trapezoids.
- **Weddle’s Rule**: A lesser-known method designed for higher accuracy by using fixed coefficients.

These methods ensure flexibility, precision, and reliability for a variety of functions and applications. The inclusion of error analysis further supports validation of results, making the tool academically and practically valuable.

Project Requirements

1. Software Requirements:

- **Programming Language:** Python 3.13 or later
- **Integrated Development Environment (IDE):** Any Python-compatible IDE (e.g., PyCharm, Visual Studio Code, or Jupyter Notebook)
- **Libraries:**
    - tkinter (for GUI development)
    - numpy (for numerical computations)
    - matplotlib (for plotting graphs)
    - math (for mathematical operations)

2. Hardware Requirements:

- **Processor:** Intel Core i5 or higher
- **RAM:** Minimum 4 GB (8 GB or more recommended for smooth performance)
- **Storage:** At least 100 MB of free disk space
- **Display:** Screen resolution of 1366x768 or higher (recommended for optimal GUI layout)

Features

1. Graphical User Interface (GUI)

- A clean, interactive interface created with Tkinter.
- Input fields for function expressions, integration limits, subintervals, and significant digits.
- Dropdown menu for selecting the desired integration method.

2. Multiple Integration Methods

- Implements four numerical integration methods to cater to different use cases.
- Error handling for method-specific constraints (e.g., even subintervals for Simpson’s 1/3 Rule).

3. Visualization

- Dynamic plotting of the input function with the shaded integration area.
- Comparison of integration results from different methods through graphical representation.

4. Accuracy and Error Analysis

- Calculates absolute error when the exact integral is known.
- Supports high precision by allowing users to specify significant digits.

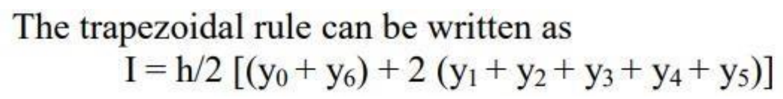
5. Code Modularity

- Modular design with separate files for individual methods, plotting, and utility functions.
- Ensures easy maintenance and future extensibility.

Theoretical Background

4.1. Trapezoidal Rule

The Trapezoidal Rule approximates the area under a curve by dividing it into trapezoids and summing their areas. It is mathematically represented as:



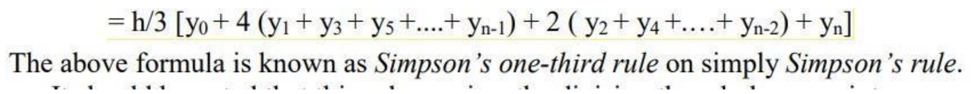
The trapezoidal rule can be written as

$$I = h/2\,[(y_0 + y_6) + 2\,(y_1 + y_2 + y_3 + y_4 + y_5)]$$

4.2. Simpson’s 1/3 Rule

This method uses quadratic polynomials to approximate the function. It is given by:

It is accurate for polynomials of degree up to 3.

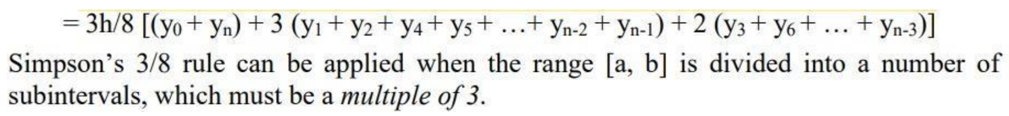


$$= h/3\,[y_0 + 4\,(y_1 + y_3 + y_5 + \ldots + y_{n-1}) + 2\,(y_2 + y_4 + \ldots + y_{n-2}) + y_n]$$

The above formula is known as *Simpson’s one-third rule* on simply *Simpson’s rule*.

4.3. Simpson’s 3/8 Rule

This rule utilizes cubic polynomials and requires the number of subintervals to be a multiple of 3:



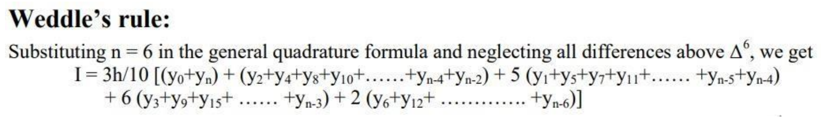
$$= 3h/8\,[(y_0 + y_n) + 3\,(y_1 + y_2 + y_4 + y_5 + \ldots + y_{n-2} + y_{n-1}) + 2\,(y_3 + y_6 + \ldots + y_{n-3})]$$

Simpson’s 3/8 rule can be applied when the range [a, b] is divided into a number of subintervals, which must be a *multiple of 3*.

4.4. Weddle’s Rule

This is a fixed-coefficient method based on higher-order polynomial interpolation. It uses the following formula:

It requires a fixed number of subintervals (6).



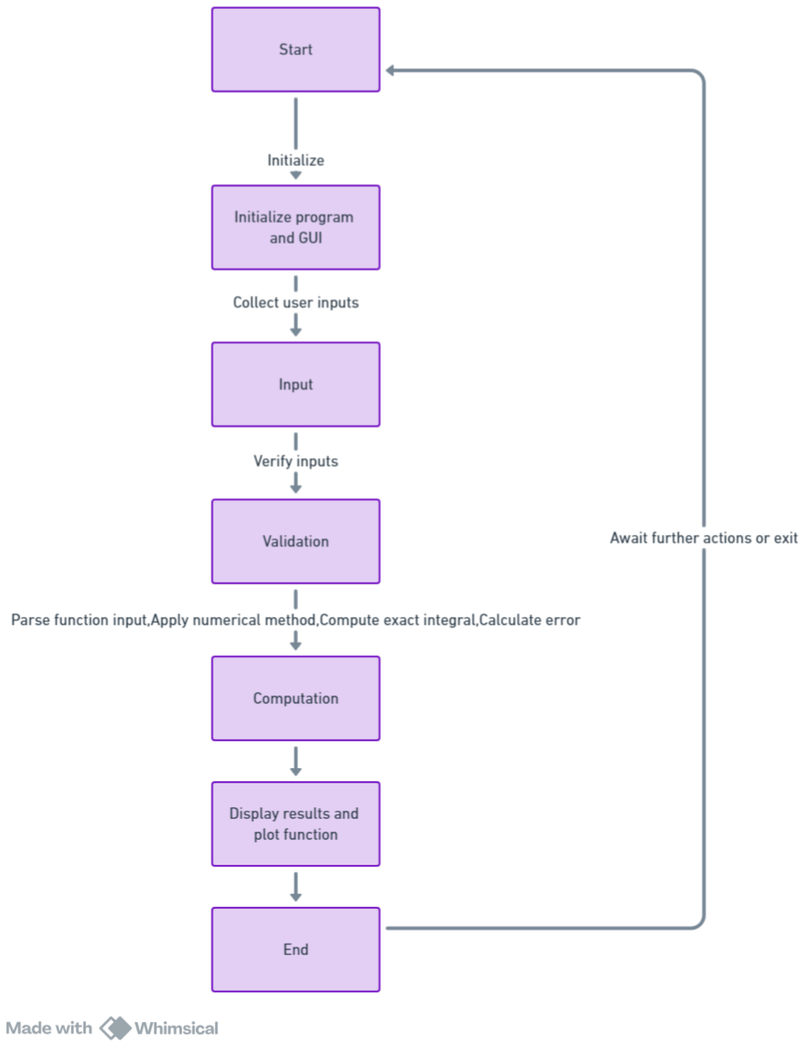
Weddle’s rule:

Substituting n = 6 in the general quadrature formula and neglecting all differences above Δ^6, we get

$$I = 3h/10\,[(y_0+y_n) + (y_2+y_4+y_8+y_{10}+\ldots\ldots+y_{n-4}+y_{n-2}) + 5\,(y_1+y_5+y_7+y_{11}+\ldots\ldots+y_{n-5}+y_{n-4}) + 6\,(y_3+y_9+y_{15}+\ldots\ldots+y_{n-3}) + 2\,(y_6+y_{12}+\ldots\ldots\ldots\ldots+y_{n-6})]$$

Implementation:

Flowchart


Start
Initialize
Initialize program and GUI
Collect user inputs
Input
Verify inputs
Validation
Parse function input,Apply numerical method,Compute exact integral,Calculate error
Computation
Display results and plot function
End
Await further actions or exit
Made with Whimsical

Functions Used

Numerical Methods:

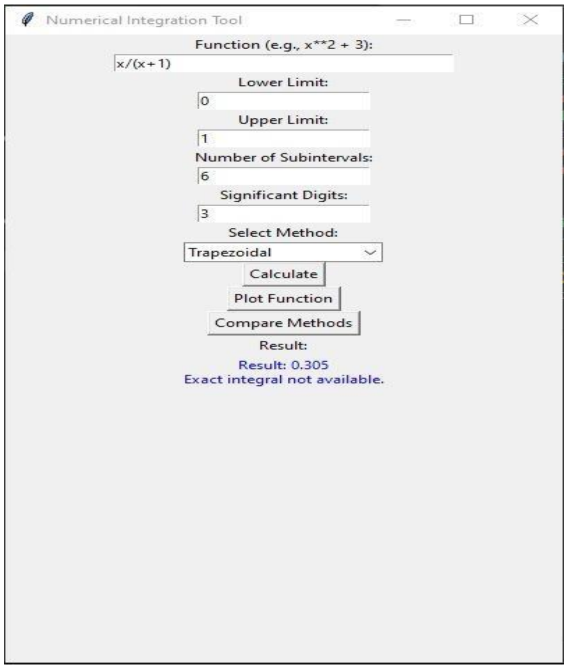
- **simpsons_one_third(f, a, b, n):** Implements Simpson's 1/3 Rule.
- **simpsons_three_eighth(f, a, b, n):** Implements Simpson's 3/8 Rule.
- **trapezoidal_rule(f, a, b, n):** Implements the Trapezoidal Rule.
- **weddle_rule(f, a, b):** Implements Weddle's Rule.

Utility Functions:

- **exact_integral(f, a, b):** Computes the exact integral using symbolic evaluation.
- **absolute_error(approx, exact):** Calculates the absolute error.
- **is_number(value):** Validates numerical input.
- **plot_function(f, a, b):** Generates a graphical plot of the function within the specified range.

Simulation

Trapezoidal



Numerical Integration Tool

Function (e.g., x**2 + 3): x/(x+1)

Lower Limit: 0

Upper Limit: 1

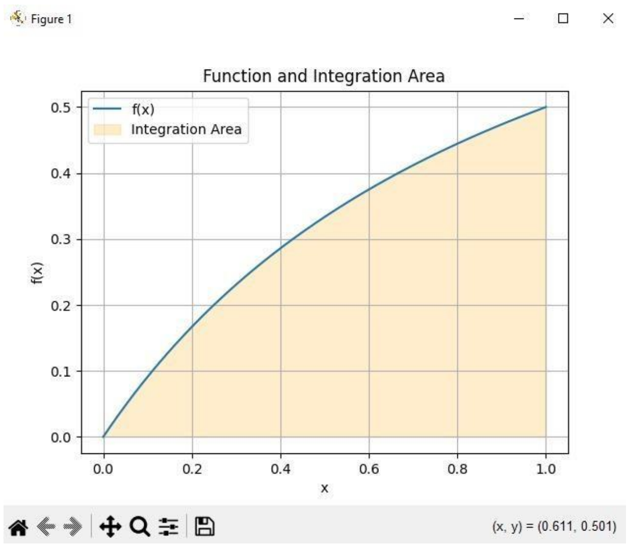
Number of Subintervals: 6

Significant Digits: 3

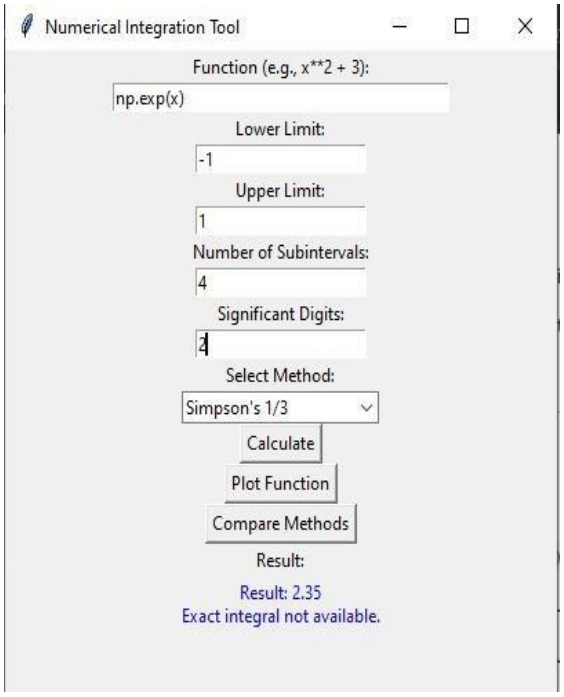
Select Method: Trapezoidal

Result:

Result: 0.305
Exact integral not available.



Simpson's 1/3



Numerical Integration Tool

Function (e.g., x**2 + 3): np.exp(x)

Lower Limit: -1

Upper Limit: 1

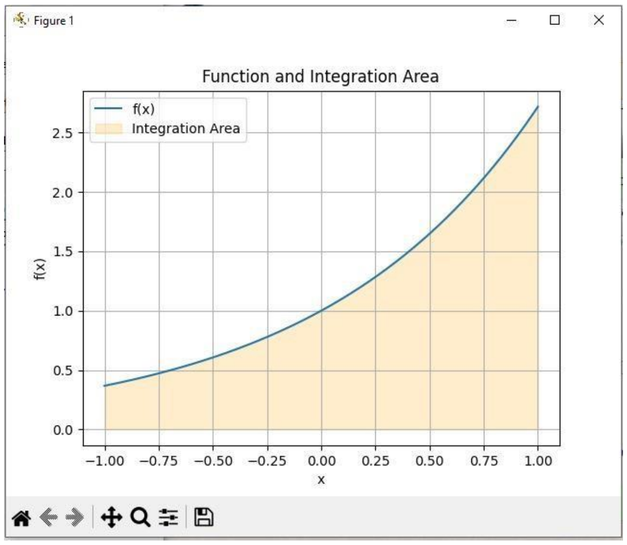
Number of Subintervals: 4

Significant Digits: 2

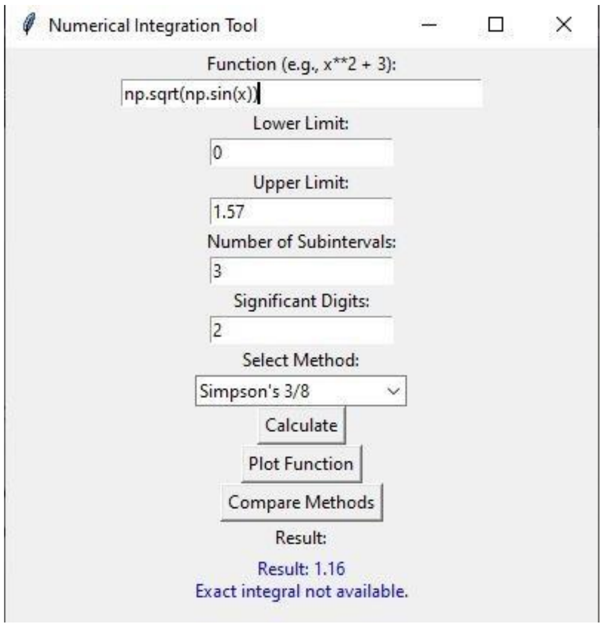
Select Method: Simpson's 1/3

Result:

Result: 2.35
Exact integral not available.



Simpson's 3/8


Numerical Integration Tool

Function (e.g., x**2 + 3): np.sqrt(np.sin(x))

Lower Limit: 0

Upper Limit: 1.57

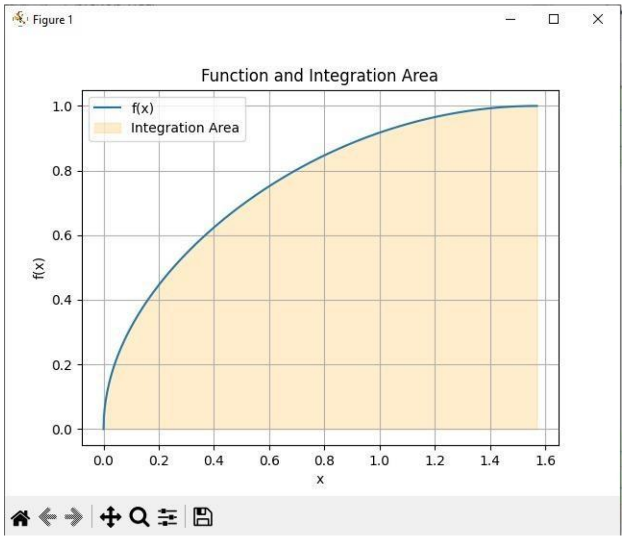
Number of Subintervals: 3

Significant Digits: 2

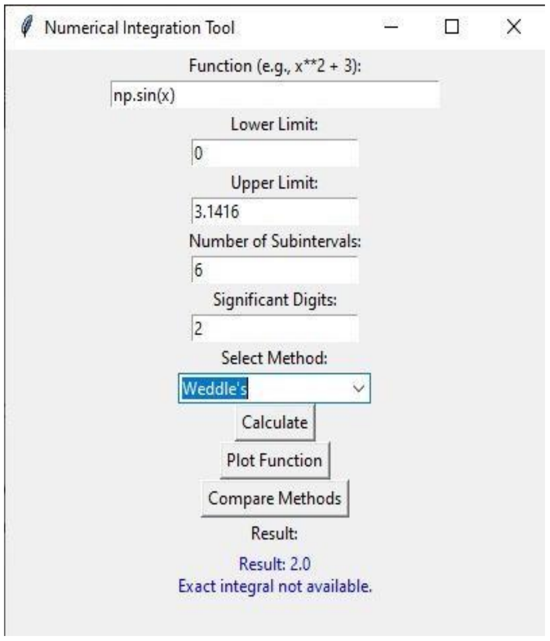
Select Method: Simpson's 3/8

Result:

Result: 1.16
Exact integral not available.



Weddle's


Numerical Integration Tool

Function (e.g., x**2 + 3): np.sin(x)

Lower Limit: 0

Upper Limit: 3.1416

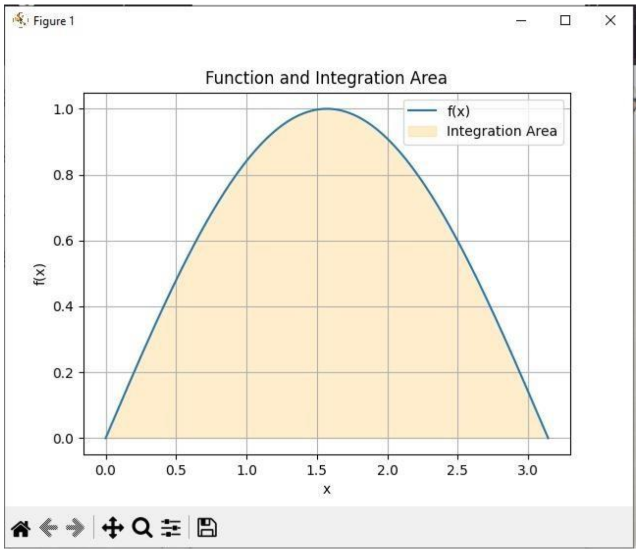
Number of Subintervals: 6

Significant Digits: 2

Select Method: Weddle's

Result:

Result: 2.0
Exact integral not available.



Future Scope

- **Advanced Methods**: Incorporate more advanced techniques like Gaussian Quadrature or Monte Carlo integration.
- **Symbolic Integration**: Integrate libraries like SymPy to provide symbolic solutions where possible.
- **Adaptive Quadrature**: Implement adaptive methods that dynamically adjust subintervals for improved accuracy.
- **Performance Optimization**: Utilize parallel computing to handle large-scale integration problems efficiently.
- **Cloud Integration**: Deploy the tool as a web-based application for broader accessibility.
- **Enhanced Error Analysis**: Provide detailed error estimation and confidence intervals.

Conclusion

This project demonstrates the implementation of a powerful and versatile numerical integration tool. By combining mathematical rigor with practical usability, it provides an excellent learning resource for numerical methods. The modular design, intuitive interface, and visualization capabilities make it a robust application for exploring and applying integration techniques. Future enhancements will further broaden its scope and utility, establishing it as a comprehensive tool for academic and professional use.