# Numerical Integration

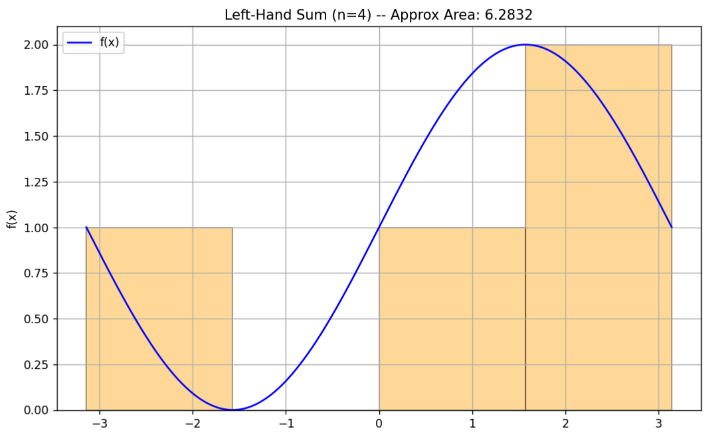
Kaceson Tisdel

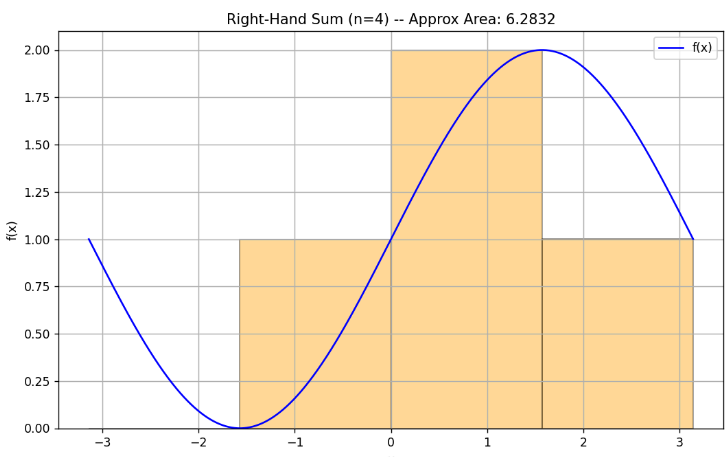
**Responsibilities and Tasks**

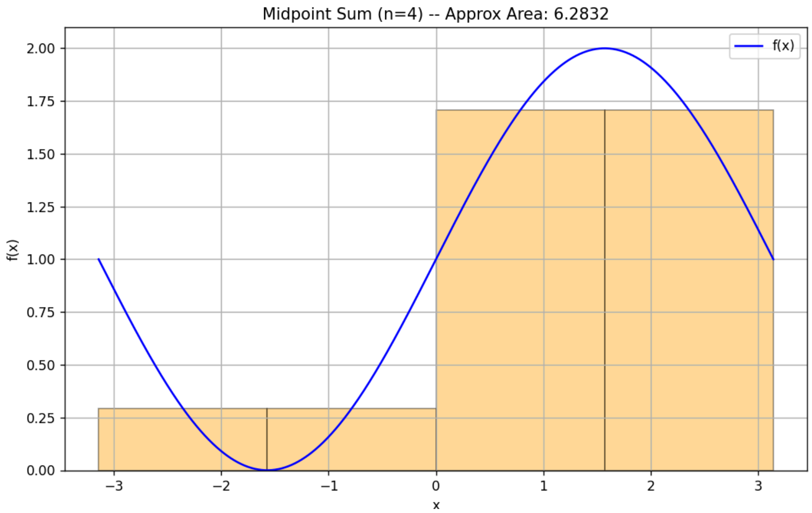
Not in a group so this was all me!

**Part 1:**

This project implements an algorithm for numerical integration to model the effects of a computing task. Part 1 focuses on developing a tool to evaluate a Riemann integral by partitioning a function over an interval into subintervals. This method involves using left, right, and midpoint sums to compare the Riemann sums to the definite integral.



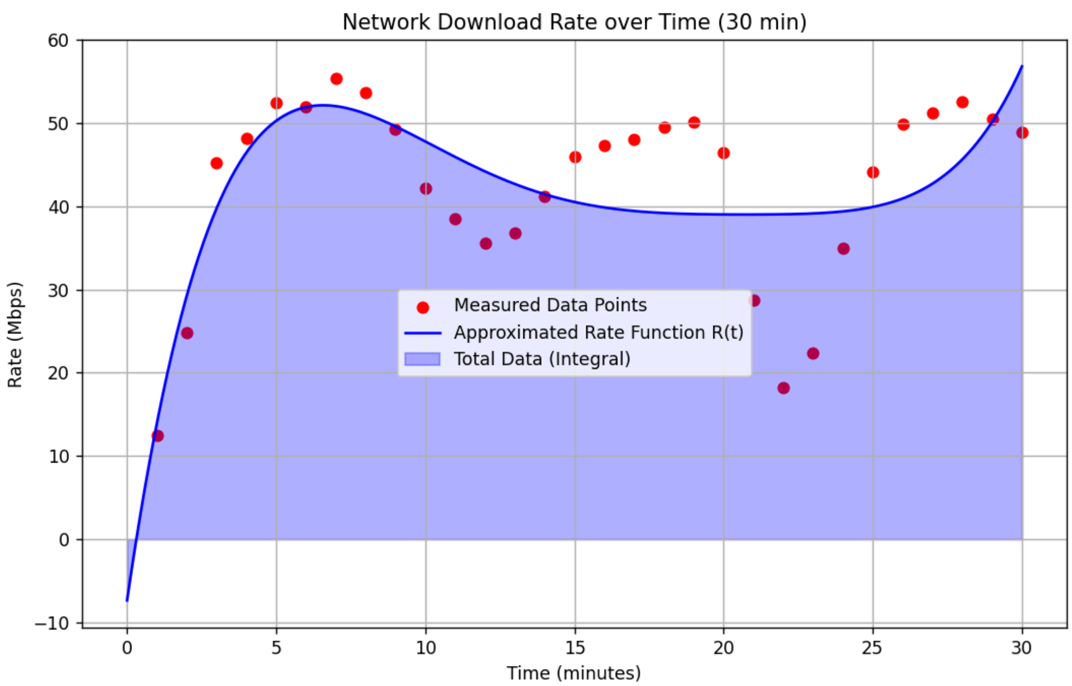


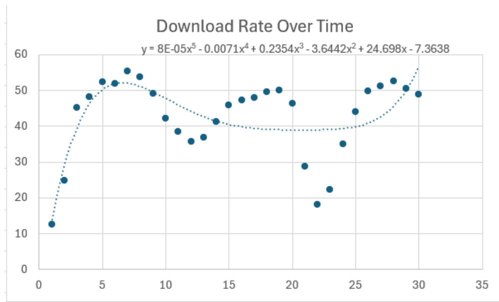

**Part 2:**

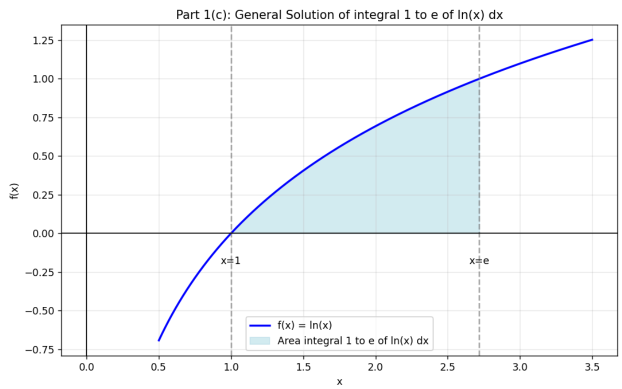
Part 2 uses the tool developed in Part 1 to calculate the amount of data transferred over a network. After recording the download rate of a media server over 1-minute intervals, the tool integrates the rate function in order to determine the total download volume over the session.

| Time (min) | Download Rate (Mbps) |
|---|---|
| 1 | 12.5 |
| 2 | 24.8 |
| 3 | 45.2 |
| 4 | 48.1 |
| 5 | 52.4 |
| 6 | 51.9 |
| 7 | 55.3 |
| 8 | 53.7 |
| 9 | 49.2 |
| 10 | 42.1 |
| 11 | 38.5 |
| 12 | 35.6 |
| 13 | 36.8 |
| 14 | 41.2 |
| 15 | 45.9 |
| 16 | 47.3 |
| 17 | 48 |

| | |
|---|---|
| 18 | 49.5 |
| 19 | 50.1 |
| 20 | 46.4 |
| 21 | 28.7 |
| 22 | 18.2 |
| 23 | 22.4 |
| 24 | 34.9 |
| 25 | 44.1 |
| 26 | 49.8 |
| 27 | 51.2 |
| 28 | 52.6 |
| 29 | 50.5 |
| 30 | 48.9 |






### System Performance and Context Description

This models a server on the internet where large files are being downloaded. The download rate may fluctuate depending on the internet conditions. This data shows how variable rates accumulate over time to produce the total transfer volume.

### Specific Problem Solved

The specific problem solved includes creating a numerical integration tool, and using it to visualize Riemann sums for f(x+ = sin(x)+1. This also involves finding the exact Riemann sum formula and limits for the functions (3x+2x^2) and (x^2-x^3) to find the exact area under the curve. Lastly, it solves for the total data transferred, as it models the discrete download rate as a continuous function and integrates it.

### Mathematical Approach

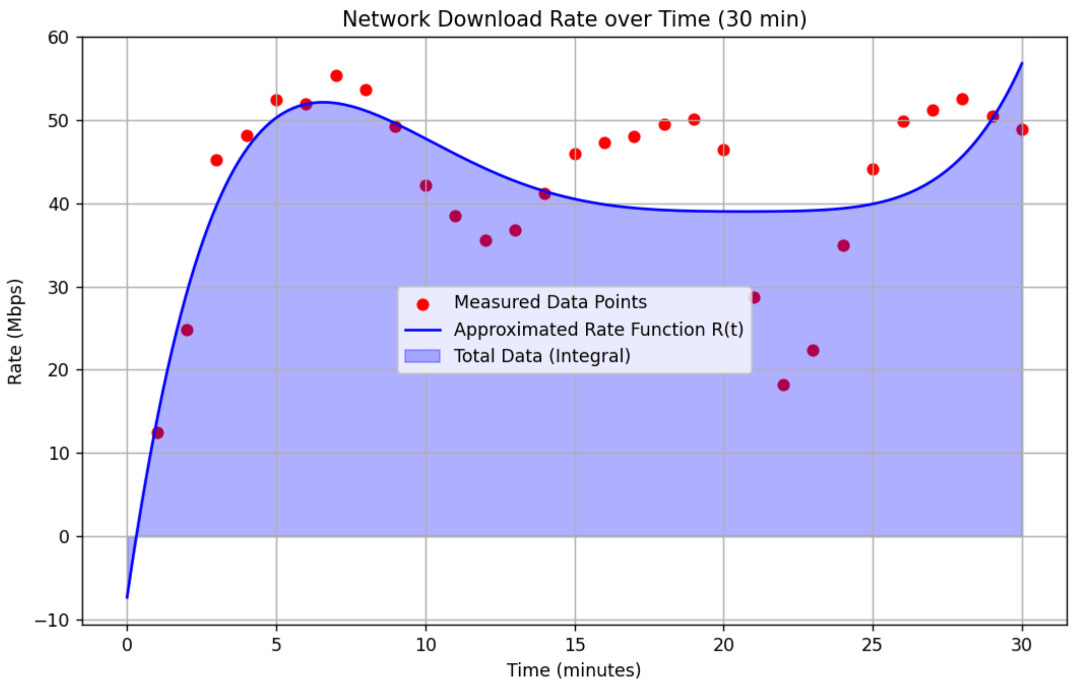
The hand-calculated math is all in the other attached PDF document

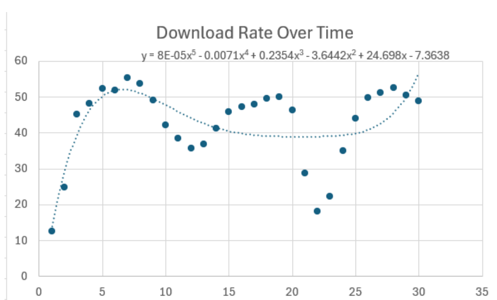
### Implementation Approach

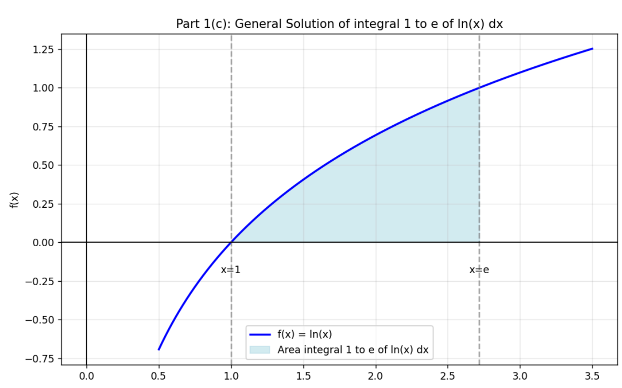
- Define a Python function named riemann_sum that takes the function, start and end points (a, b), number of subintervals (n), and the method (Left, Right, or Midpoint) as inputs.
- Inside the function, calculate the width of each subinterval (delta x) and generate the specific sample points based on the chosen method to sum the area of the rectangles.
- Generate graphs for f(x) = sin(x) + 1 over the interval [-pi, pi] using the matplotlib library, drawing rectangles to visually represent the Left, Right, and Midpoint Riemann sums.
- Create a specific plot with high detail (granularity) to show the general solution for the integral of ln(x) dx from 1 to e .
- Create an array containing the 30 data points representing the download rate in Mbps for each minute of the 30-minute session.
- Use the numpy.polyfit function to create a continuous polynomial function R(t) that fits the discrete download rate data points.
- Apply the Riemann sum tool developed in Part 1 to integrate this new rate function R(t) over the interval [0, 30] to calculate the total area under the curve.

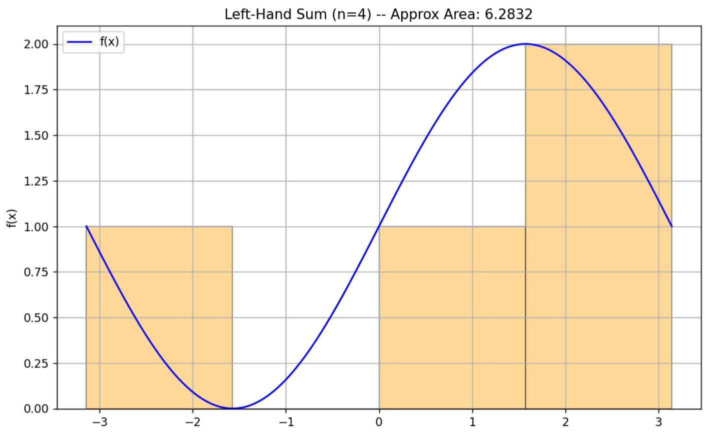
- Multiply the result by 60 to convert the value from "Megabits per minute" to "Megabits" and then divide by 8 to get "Megabytes" for the final answer.

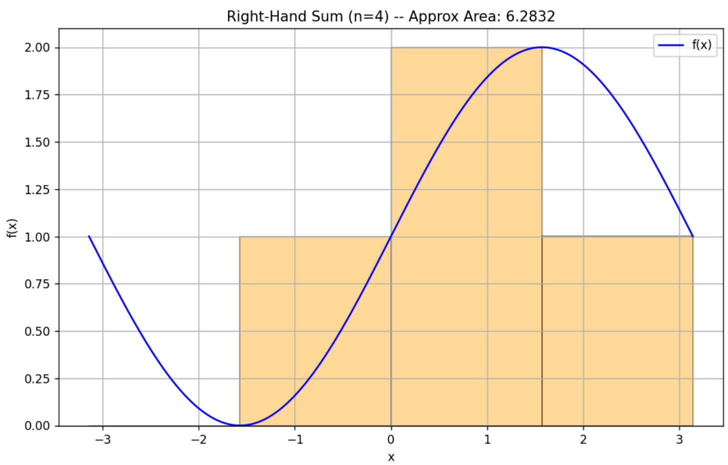
**Key Phases**

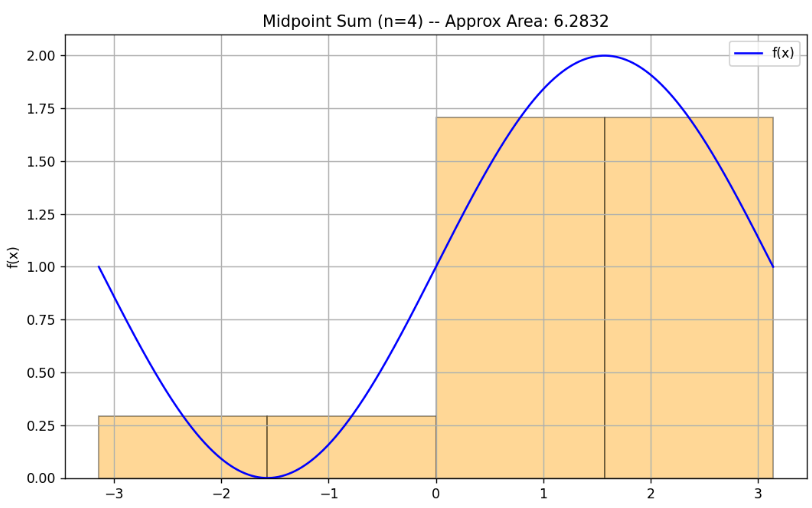





## Refrences

Your slides!