Draft version March 20, 2023
Typeset using LaTeX **twocolumn** style in AASTeX631

A Program to Interpolate a Runge Function using Lagrangian and Cubic Spline Interpolation and Integrate a Gaussian Function using Trapezoidal and Simpson's Composite Integration

Akshat S. Chaturvedi[1]

[1] *Department of Astronomy & Astrophysics, The Pennsylvania State University, University Park, PA 16802, USA*

ABSTRACT

I created a `python` program that interpolates a Runge function using discrete data points using Lagrangian interpolation, cubic spline interpolation and compares the two. The program also integrates a given Gaussian function using Trapezoidal composite and Simpson's composite integration, and compares the two.

1. INTRODUCTION

Lagrange interpolation was developed by Joseph-Louis Lagrange in 1795, and is used for interpolating data when data is presented in a discrete form. Interpolating in this context refers to approximating how data is expected to behave between the given data points. Lagrange interpolation used Lagrange polynomials to interpolate data, which are found as described below.

The program also uses cubic splines for interpolation. Cubic splines work by creating piecewise fits between the discrete data points, which all together forms a proper interpolated fit for the data.

The program also makes use of Trapezoidal and Simpson's composite integration to find the value of the integral of a Gaussian function at non-infinite bounds.

The Trapezoidal composite calculates the integral by forming a large number of trapezoids and calculating the area of all the trapezoids.

The Simpson's composite works by fitting multiple parabolas to the given function using "triplets" of data and calculating the integral of each of those parabolas[1].

2. METHODS

2.1. *Lagrange Interpolation*

The program uses Lagrange interpolation to interpolate the Runge function shown in equation 1 over the range $[-1, 1]$.

$$f(x) = \frac{1}{25x^2 + 1} \tag{1}$$

The program works by first calculating the y values at n nodes in the given range. These are simply the Runge function in equation 1 being evaluated at those points.

[1] http://encyclopediaofmath.org/index.php?title=Simpson_formula&oldid=48712

It then calculates the barycentric weights of the nodes using equation 2 (Berrut & Trefethen (2004))

$$w_j = \prod_{m \neq j} \frac{1}{x_j - x_m} \tag{2}$$

These weights are then multiplied by their respective Lagrange basis polynomial, which is found using equation 3.

$$l(x) = \prod_m (x - x_m) \tag{3}$$

The overall Lagrange interpolation polynomial is then found using equation 4

$$L(x) = \sum_{j=0}^{n} y_j \cdot l_j(x) \cdot w_j \tag{4}$$

These polynomials can then be used to interpolate the given data for a given number of nodes.

2.2. *Cubic Spline Interpolation*

This program also makes use of cubic spline interpolation to interpolate the same Runge function as shown in equation 1.

The cubic spline works by forming third degree polynomial splines which act as piecewise fits between two consecutive nodes (Young & Mohlenkamp (2017)).

These splines are of the form shown in equation 5

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \tag{5}$$

These splines must follow the following conditions:

- $S_i(x_i) = y_i = S_{i-1}(x_i)$
- $S_0(x) = y_0$

- $S_{n-1}(x_n) = y_n$
- $S'_i(x_i) = S'_{i-1}(x_i)$
- $S''_i(x_i) = S''_{i-1}(x_i)$
- $S''_0(x_0) = S''_{i-1}(x_i) = 0$

These splines, when plotted together as a set of piecewise functions form the interpolated data set.

These splines can be found using an algorithmic approach as shown in the `python` program.

2.3. *Composite Trapezoidal Integration*

The program also makes use of composite trapezoidal integration to approximate the integral of the Gaussian function shown in equation 6 from -100 to 100

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}} \tag{6}$$

The trapezoidal integration works by forming n trapezoids between $n+1$ points, where the lengths of the trapezoids are the Gaussian function evaluated at those points, and the heights of the trapezoids are a fixed length, found using equation 7 where a and b are the starting and ending points of the integration (-100 and 100 in this case).

$$h = \frac{|a| + |b|}{n} \tag{7}$$

The areas of these trapezoids are calculated and summed, giving an approximation of the total area under the curve, which equals the value of the integral. This summed area is found using equation 8

$$\int_a^b f(x)dx = \frac{h}{2}(f(a) + f(b)) + h\sum_{i=1}^{n-1} f(x_i) \tag{8}$$

2.4. *Simpson's Composite Integration*

The Simpson's composite integration works by fitting multiple parabolas function by dividing the entire range into triplets. The integral is calculated for these parabolas which tends to be considerably less complex than the actual function itself.

The overall integration is performed using equation

$$\begin{aligned}\int_a^b f(x)dx = \frac{h}{3}(f(a) + f(b)) + \frac{4h}{3}\sum_{i=1}^{\frac{n}{2}} f(x_{2i-1}) \\ + \frac{2h}{3}\sum_{i=1}^{\frac{n}{2}-1} f(x_{2i})\end{aligned} \tag{9}$$

3. RESULTS AND DISCUSSION

3.1. *Interpolation*

The resultant polynomials from the Lagrange interpolation is as shown in figure 1. As expected, as n gets larger, the interpolated data fits the Runge function better.

The cubic spline interpolation on the other hand, is shown in figure 1. As is clearly visible, for 10 "iterations", the cubic spline interpolates the data to a significantly better fit than the Laggrange interpolation. I believe that this is due to the piecewise nature of the cubic splines, which makes it easier to fit for smaller intervals, as compared to the Lagrange interpolation, which works on a larger scale, and hence is more prone to flaws, such as the Runge phenomenon, which is the oscillatory spikes towards the edges of the data as seen in the Lagrange interpolations (Pierce (2017)).

3.2. *Integration*

The composite trapezoidal integration and Simpson's integration both converged to the analytical value of the Gaussian integral when given a large enough number of iterations. A plot of the value of the integral as a function the number of steps is shown in figure 2, as well as a plot of the error as a function of the number of steps.

As is clearly visible from the plot, the trapezoidal composite converges to the analytical value quicker, as well as having smaller errors for smaller step sizes. This is expected as the composite trapezoidal integration is numerically simpler, and as such requires fewer iterations to converge. Simpson's composite rule also works with parabolas made using triplets of points, and as such for a shape such as a Gaussian, which has parabolic nature itself, it requires more iterations to converge.

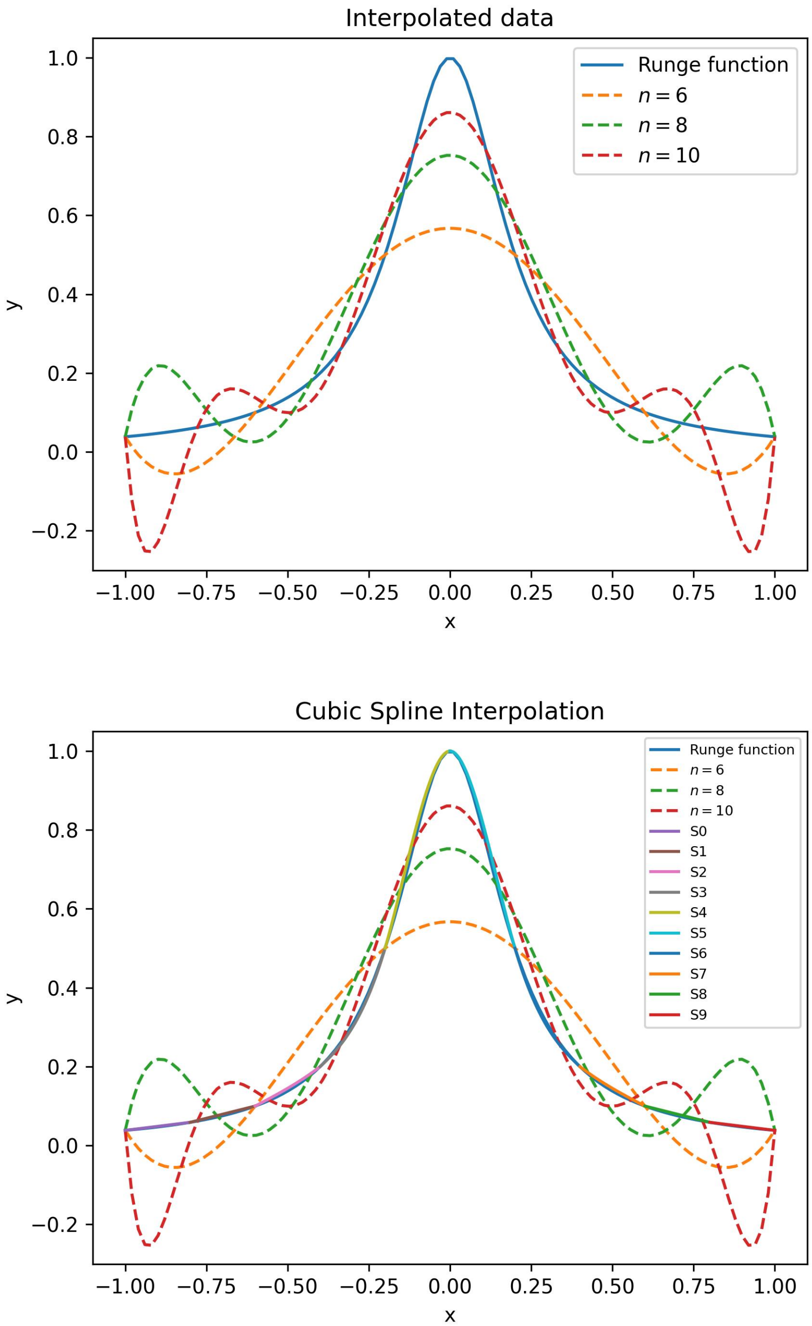

Figure 1: **a.** Lagrange Interpolation for $n = 6, 8, 10$, and **b.** The cubic spline interpolation over plotted with the Lagrange interpolation

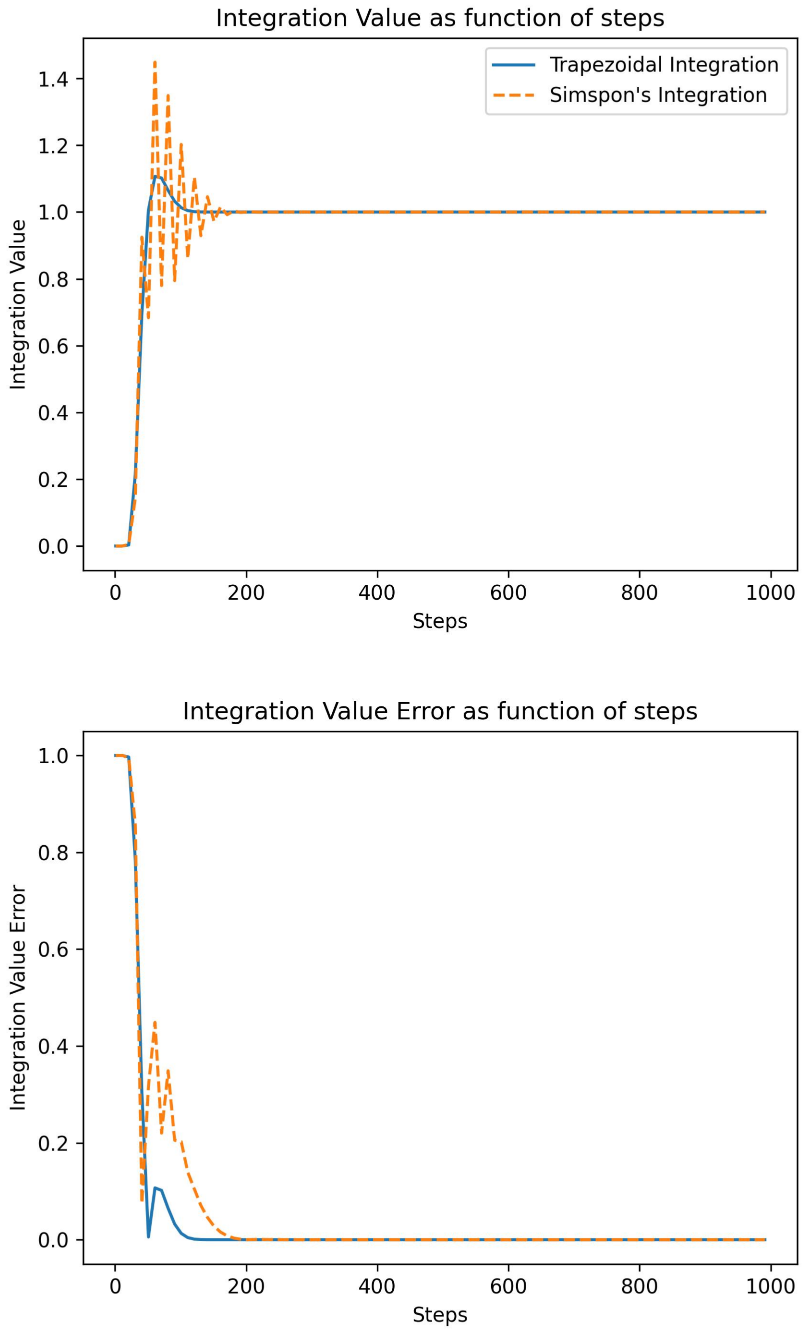

Figure 2: **a.** The integral value as a function of the number of steps, and **b.** the error in the integral value as a function of the number of steps

REFERENCES

Berrut, J.-P., & Trefethen, L. N. 2004, SIAM review, 46, 501

Pierce, A. 2017, The Runge Phenomenon and Piecewise Polynomial Interpolation, University of British Columbia

Young, T., & Mohlenkamp, M. J. 2017, Department of Mathematics Ohio University, 93