MTH 343 - Numerical Analysis 1

Numerical Quadrature - Simpson's Rules

Abdul Aziz Mohammed @00094186

Hamad Alowais @00079428

Khalifa Almatrooshi @00090847

21 November 2023

Abstract

In this research, our team will be exploring the Simpson's Rule, a numerical integration technique. We delve into the history of the Simpson's rule both the 1/3 and 3/8 rule. The use of the technique is to approximate integrals of functions that are analytically insolvable. There are various integrals that are widely used in different fields of engineering and science. For instance, Elliptic integrals help to calculate the circumference of an ellipse to help determine the paths of celestial bodies. In our paper, the accuracy of the 1/3 and 3/8 will be assessed through first a simple example, then a real world example. Alongside that, a MATLAB code is produced to compare with the work done by hand. Using the results from our MATLAB code we intend to demonstrate the accessibility of Simpson's rule and its importance in various scientific fields.

Introduction

Simpson's rule is a key technique in numerical quadrature that has a rich history that predates its namesake, Thomas Simpson (1710-1761), involving several key figures. Today, we have two versions: the 1/3 Simpson's rule, and the 3/8 Simpson's rule.

The conceptual roots of Simpson's Rule can be traced back to the $17th$ century with Johannes Kepler (1571-1630). Kepler, dealing with the practical problem of calculating the volume of wine barrels, developed a method known as "Kepler's Barrel Rule" (*1*). The volume was found to be close to $h(A+4C+B)/6$, where A and B are the cross sectional areas at both ends, and C at the center (*2*). This result is close to the 1/3 Simpson's rule. Kepler's method laid the groundwork for future developments in approximating areas and volumes of irregular figures using geometric shapes.

In the mid-$17th$ century, Bonaventura Cavalieri (1598-1647), a student of Galileo, used a version of Simpson's rule in his work. After that, James Gregory (1638-1675) made significant strides in the field of numerical quadrature (*3*). Particularly his development of a quadrature formula that is an early form of the Newton-Cotes formulas, with the idea of increasing equidistant nodes (*4*).

Clearly, Simpson did not actually invent the rule named after him. His part was the popularization and further development of the method in the mid-$18th$ century. In 1743, Simpson published a paper in which he described the relevant rule, along with other methods for numerical quadrature. This was a significant contribution that eventually lead to it being named after him (*3*).

Today, both versions of Simpson's Rule are essential tools in numerical analysis, widely used in engineering, physics, and mathematics. The rules' simplicity and accuracy make them effective for a range of applications, particularly where exact solutions to integrals are not feasible.

Simpson's rules are closed Newton-Cotes formulae of $n = 2$ and $n = 3$ that approximate the integral of a given function. This is done by integrating an interpolating polynomial at a certain degree of that given function. In general, the Lagrangian interpolation polynomial is used; including its error term.

$$f(x) = \sum_{i=0}^{n} f(x_i)L_i(x) + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{j=0}^{n}(x - x_j)}_{E_n(x)}$$

Integrating this function over a finite interval $[a, b]$.

$$\int_a^b f(x)\ dx = \sum_{i=0}^{n} a_i f(x_i) + E_n \approx \sum_{i=0}^{n} a_i f(x_i)$$

For the Newton-Cotes formulae, the points x_i are uniformly spaced by $x_i = x_0 + ih$ $(i = 0, 1, \ldots, n)$, where h is a constant called the step size. In the case of the closed Newton-Cotes formulae, they include the endpoints of the closed interval $[a, b]$, where $x_0 = a$, $x_n = b$, and $h = \dfrac{b-a}{n}$. The coefficients a_i, called the Cotes numbers, are the weights assigned to each $f(x_i)$ that indicate how much each point contributes to the overall approximation of the integral (5).

With that, the $(n + 1)-$point closed Newton-Cotes formulae can be derived. This project focuses on Simpson's rules which are the $n = 2$ and $n = 3$ cases, quadratic interpolation and cubic interpolation, respectively (6).

$$\int_{x_0}^{x_2} f(x)\ dx = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right] - \frac{h^5}{90}f^{(4)}(\xi), \qquad where \quad x_0 < \xi < x_2$$

$$\int_{x_0}^{x_3} f(x)\ dx = \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right] - \frac{3h^5}{80}f^{(4)}(\xi), \qquad where \quad x_0 < \xi < x_3$$

$$\text{Absolute Error} = |\text{Exact Value} - \text{Approximate Value}|$$

The goal of this project is to test and compare the effectiveness of the 1/3 Simpson's Rule and the 3/8 Simpson's Rule. First with a short example done by hand that demonstrates the accessibility of the technique. Then run the same example through a MATLAB code with a greater accuracy. An application of Simpson's Rule will be explored in Elliptic Integrals.

Example

The following function comes from the Mathematical Methods in Physics and Engineering book, by K. F. Riley, M. P. Hobson, and S. J. Bence.; specifically problem 27.10 on page 1034 (*7*). We want to find the integral of the function bounded by -1 and 1.

$$f(x) = \frac{\mathrm{e}^{-2x^2}}{4x^2 + 3x + 1}$$

$$F(x) = \int_{-1}^{1} \frac{\mathrm{e}^{-2x^2}}{4x^2 + 3x + 1}\, dx$$

At a glance, it is difficult to find it analytically. Even an online integral calculator (`https://www.integral-calculator.com`) resorts to numerical integration. The approximate value of the integral is 1.258 275 887 872 749 (15 d.p.), by courtesy of said integral calculator. It also mentions that the estimated absolute error is $2.895\,914\,430\,441\,944 \times 10^{-14}$, therefore it is appropriate to take the approximation as the exact value. The following graph was plotted on `https://www.desmos.com/calculator/cvpnb3lsls`

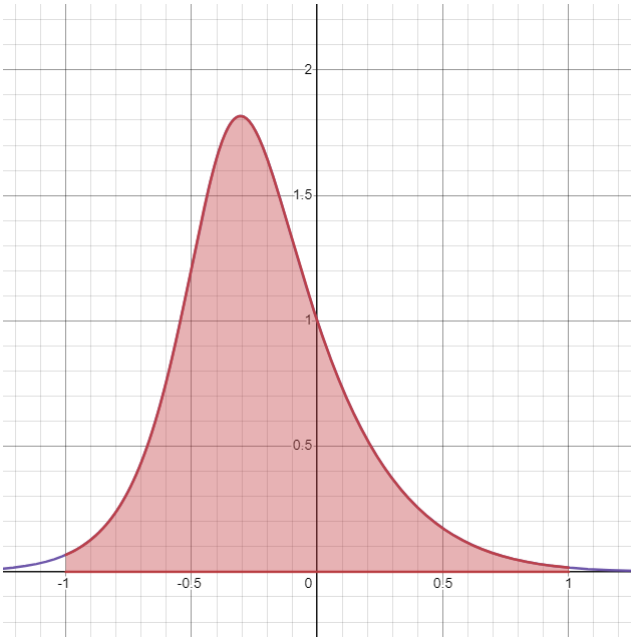


Figure 1: Graph of function with shaded area

Now to use Simpson's rules to approximate the integral, starting with 1/3 Simpson's rule.

$$h = \frac{1-(-1)}{2} = 1 \quad , \quad x_i = -1 + i(1) \text{ for } i = 0, 1, 2$$

$$\begin{aligned}\int_{-1}^{1} f(x)\ dx &\approx \frac{1}{3}\left[f(-1) + 4f(0) + f(1)\right] \\ &\approx \frac{1}{3}\left[0.067\,67 + 4(1) + 0.016\,92\right] \approx 1.361\,53\end{aligned}$$

$$\text{Absolute Error} = |1.258\,28 - 1.361\,53| = 0.103\,25$$

$$E_{\text{t}} = \frac{h^5}{90} f^{(4)}(\xi) \quad , \quad -1 < \xi < 1$$

It is tedious to find the derivative for this function, therefore we resorted to finding the maximum $f^{(4)}(\xi)$ value graphically, $f^{(4)}(-0.348) = 4222.031\,25$ (`https://www.desmos.com/calculator/xzz9dn5oco`).

$$E_{\text{t}} = \frac{(1)^5}{90}(4222.031\,25) = 46.911\,46 >> 0.103\,25$$

Now for 3/8 Simpson's rule.

$$h = \frac{1-(-1)}{3} = \frac{2}{3} \quad , \quad x_i = -1 + i\left(\frac{2}{3}\right) \text{ for } i = 0, 1, 2, 3$$

$$\begin{aligned}\int_{-1}^{1} f(x)\ dx &\approx \frac{3(\frac{2}{3})}{8}\left[f(-1) + 3f(-\frac{1}{3}) + 3f(\frac{1}{3}) + f(1)\right] \\ &\approx \frac{1}{4}\left[0.067\,67 + 3(1.801\,66) + 3(0.327\,57) + 0.016\,92\right] \approx 1.618\,07\end{aligned}$$

$$\text{Absolute Error} = |1.258\,28 - 1.618\,07| = 0.359\,79$$

$$E_{\text{t}} = \frac{3\left(\frac{2}{3}\right)^5}{80}(4222.031\,25) = 20.849\,54 >> 0.359\,79$$

For both rules, the error term is much larger compared to its absolute error. This is clearly because of the inherent step size in the rules, and the quadratic and cubic interpolating polynomials. From the graphing tool, the function exhibits an oscillatory behavior as more derivatives are taken. Therefore the error term may be larger to account for the oscillatory behaviors in the function that the methods cannot capture accurately. In the next section, we will create a MATLAB code for both Simpson's rules.

Figure 2: Example done by hand

Q7 $\int_{-1}^{1} \frac{e^{-2x^2}}{4x^2+3x+1}\,dx$ Actual value = 1.2582759

$n = 2$

1/3rd $h = \frac{1+1}{2} = 1$

$$\int_a^b f(x)\,dx = \frac{h}{3}\left[f(a) + 4f(x_1) + f(b)\right]$$

$$\therefore \int_{-1}^{1} f(x)\,dx = \frac{1}{3}\left[f(-1) + 4f(0) + f(1)\right]$$

$$= \frac{1}{3}\left[0.067668 + 4 + 0.0169169\right]$$

$$= 1.36153$$

Absolute Error $= |1.36153 - 1.25828|$

$= 0.10325$

Error Bound $= \frac{h^5}{90} f^{(4)}(\xi)$ $\quad -1 < \xi < 1$

$f^{(4)}(\xi)$ is max at -0.348

error bound $= \frac{1}{90}(4222.031) = 46.911$

3/8th $h = \frac{1+1}{3} = \frac{2}{3} = 0.667$

$$\int_a^b f(x)\,dx = \frac{h}{8}\left[f(a) + 3f(x_1) + 3f(x_2) + f(b)\right]$$

$$\therefore \int_{-1}^{1} f(x)\,dx = \frac{0.667}{8}\left[f(-1) + 3f(-\tfrac{1}{3}) + 3f(\tfrac{1}{3}) + f(1)\right]$$

$$= 1.61807$$

Abs. Error $= |1.25828 - 1.61807| = 0.35979$

Error bound $= \frac{3(0.667)^5}{80}(4222.03125) = 20.84954$

MATLAB Code

```
%% Simpson's 1/3 Rule for Numerical Integration

% This function applies Simpson's 1/3 Rule for numerical integration to
    estimate the integral of a given function.

function simpson1by3(f, a, b)
    % Calculate the exact integral
    exact_integral = integral(f, a, b);

    %% Apply Simpson's 1/3 Rule for n = 2
    % Divide the interval [a, b] into three subintervals and
       approximate the integral using Simpson's 1/3 Rule.
    h = (b - a) / 2;
    x_interpolation = [a, a + h, b];
    interpolated_values = (h / 3) * (f(x_interpolation(1)) + 4 * f(
       x_interpolation(2)) + f(x_interpolation(3)));

    %% Approximate the fourth derivative of the function using finite
       differences to find the error term.
    num_points = 1000;
    x_points = linspace(a, b, num_points);
    delta_x = x_points(2) - x_points(1);
    fourth_derivative_values = zeros(size(x_points));

    for i = 3:(num_points - 2)
        fourth_derivative_values(i) = (f(x_points(i - 2)) - 4 * f(
           x_points(i - 1)) + 6 * f(x_points(i)) - 4 * f(x_points(i +
           1)) + f(x_points(i + 2))) / (delta_x^4);
    end

    % Determine the maximum absolute value of the fourth derivative
       within the interval.
    max_fourth_derivative = max(abs(fourth_derivative_values));
    %disp(fourth_derivative_values);
    %disp(max_fourth_derivative);

    % Compute the error term for Simpson's 1/3 Rule using the maximum
       fourth derivative and the step size.
    error_term = (h^5 / 90) * max_fourth_derivative;

    %% Determine the absolute and relative error by comparing the
       estimated integral with the exact integral.
    absolute_error = abs(exact_integral - interpolated_values);
    relative_error_percentage = (absolute_error / abs(exact_integral))
       * 100;

    %% Construct a quadratic interpolating polynomial based on the
       interpolating points.
    p = polyfit(x_interpolation, [f(x_interpolation(1)), f(
       x_interpolation(2)), f(x_interpolation(3))], 2);
    interpolated_polynomial = polyval(p, x_interpolation);

    %% Initialize and create a plot
    % Generate evenly spaced points for plotting the actual function
       and interpolation.
```

```
    x_values = linspace(a, b, 1000);
    f_values = f(x_values);

    % Plot the actual function, interpolating points, and interpolating
        polynomial to visualize the results.
    figure;
    plot(x_values, f_values, 'b', 'LineWidth', 2); % Actual function in
        blue
    hold on;
    plot(x_interpolation, f(x_interpolation), 'ro', 'MarkerSize', 10);
       % Interpolating points in red
    plot(x_interpolation, interpolated_polynomial, 'g--', 'LineWidth',
       2); % Interpolating polynomial in green dashed line

    %$ Label the axes, add a title, and add a legend.
    xlabel('x');
    ylabel('f(x)');
    title('Actual Function and Interpolating Polynomial (Simpson''s 1/3
        Rule)');
    legend('f(x)', 'Interpolating Points', 'Interpolating Polynomial',
       'Location', 'NorthWest');

    %% Display the results
    fprintf('Estimated Integral: %f\n', interpolated_values);
    fprintf('Exact Integral: %f\n', exact_integral);
    fprintf('Absolute Error: %f\n', absolute_error);
    fprintf('Relative Error (Percentage): %.2f%%\n',
       relative_error_percentage);
    fprintf('Error Term: %f\n', error_term);
end
```

```
%% Simpson's 3/8 Rule for Numerical Integration

% This function applies Simpson's 3/8 Rule for numerical integration to estimate the integral of a given function.

function simpson3by8(f, a, b)
    % Calculate the exact integral
    exact_integral = integral(f, a, b);

    %% Apply Simpson's 3/8 Rule for n = 3
    % Divide the interval [a, b] into four subintervals and approximate the integral using Simpson's 3/8 Rule.
    h = (b - a) / 3;
    x_interpolation = [a, a + h, a + 2 * h, b];
    interpolated_values = (3 * h / 8) * (f(x_interpolation(1)) + 3 * f(x_interpolation(2)) + 3 * f(x_interpolation(3)) + f(x_interpolation(4)));

    %% Approximate the fourth derivative of the function using finite differences for error estimation.
    num_points = 1000;
    x_points = linspace(a, b, num_points);
    delta_x = x_points(2) - x_points(1);
    fourth_derivative_values = zeros(size(x_points));

    for i = 3:(num_points - 2)
        fourth_derivative_values(i) = (f(x_points(i - 2)) - 4 * f(x_points(i - 1)) + 6 * f(x_points(i)) - 4 * f(x_points(i + 1)) + f(x_points(i + 2))) / (delta_x^4);
    end

    % Determine the maximum absolute value of the fourth derivative within the interval.
    max_fourth_derivative = max(abs(fourth_derivative_values));
    %disp(fourth_derivative_values);
    %disp(max_fourth_derivative);

    % Compute the error term for Simpson's 3/8 Rule using the maximum fourth derivative and the step size.
    error_term = (3 * h^5 / 80) * max_fourth_derivative;

    %% Determine the absolute and relative error by comparing the estimated integral with the exact integral.
    absolute_error = abs(exact_integral - interpolated_values);
    relative_error_percentage = (absolute_error / abs(exact_integral)) * 100;

    %% Construct a cubic interpolating polynomial based on the interpolating points.
    p = polyfit(x_interpolation, [f(x_interpolation(1)), f(x_interpolation(2)), f(x_interpolation(3)), f(x_interpolation(4))], 3);
    interpolated_polynomial = polyval(p, x_interpolation);

    %% Initialize and create a plot
    % Generate evenly spaced points for plotting the actual function and interpolation.
```

```
    x_values = linspace(a, b, 1000);
    f_values = f(x_values);

    % Plot the actual function, interpolating points, and interpolating
        polynomial to visualize the results.
    figure;
    plot(x_values, f_values, 'b', 'LineWidth', 2); % Actual function in
        blue
    hold on;
    plot(x_interpolation, f(x_interpolation), 'ro', 'MarkerSize', 10);
       % Interpolating points in red
    plot(x_interpolation, interpolated_polynomial, 'g--', 'LineWidth',
       2); % Interpolating polynomial in green dashed line

    %% Label the axes, add a title, and add a legend.
    xlabel('x');
    ylabel('f(x)');
    title('Actual Function and Interpolating Polynomial (Simpson''s 3/8
        Rule)');
    legend('f(x)', 'Interpolating Points', 'Interpolating Polynomial',
       'Location', 'NorthWest');

    %% Display the results
    fprintf('Estimated Integral: %f\n', interpolated_values);
    fprintf('Exact Integral: %f\n', exact_integral);
    fprintf('Absolute Error: %f\n', absolute_error);
    fprintf('Relative Error (Percentage): %.2f%%\n',
       relative_error_percentage);
    fprintf('Error Term: %f\n', error_term);
end
```

Testing the Code

To test the code we will redo the first example. The code outputs the estimated integral, exact integral (adaptive numerical quadrature), absolute error, relative error, and the error term. Alongside that we were able to produce figures that show the actual function and the interpolating polynomial for each rule.

$$f(x) = \frac{\mathrm{e}^{-2x^2}}{4x^2 + 3x + 1}$$

$$F(x) = \int_{-1}^{1} \frac{\mathrm{e}^{-2x^2}}{4x^2 + 3x + 1}\, dx$$

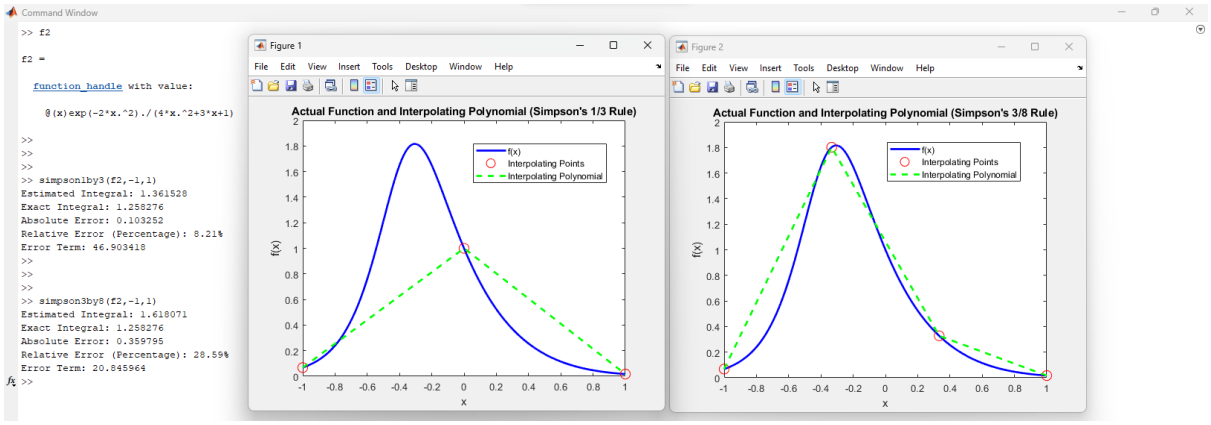


Figure 3: Example 1 in MATLAB

We get similar results as to when done by hand. The large error term could be because of the apparent discrepancy between the areas under the actual function and the interpolating polynomials. Overall, this shows the accessibility of the Simpson's rules.

Application

Numerical quadrature is a fundamental technique used in many real-world applications in all of STEM. The main idea is to compute the integral of a function that is either difficult to solve analytically, or even there is no analytical solution described by elementary functions. Simpson's Rule is one such method to do that.

There are many special integral equations like Fredholm integrals, Volterra integrals, Elliptic integrals, etc. Each type of integral plays a distinct role in addressing a variety of mathematical and scientific problems. We will focus on Elliptic integrals.

Elliptic integrals are a family of special functions used to compute various geometric and mathematical properties of elliptic curves and related mathematical structures, particularly in areas such as classical mechanics and complex analysis. There are several types of elliptic integrals, including incomplete and complete elliptic integrals, and they are categorized into three main types: elliptic integrals of the first kind, second kind, and third kind. In our case, we will explore how the complete elliptic integral of the second kind is applied to find the circumference of an ellipse ($\mathcal{8}$).

$$E(\frac{\pi}{2}, k) = E(k) = \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 \theta}\ d\theta$$

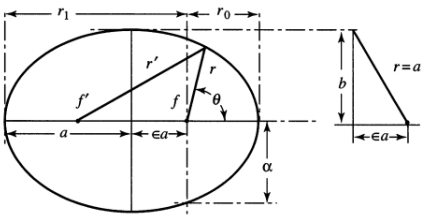


Figure 6.5.1 The ellipse

f, f'	The two foci of the ellipse
a	Semimajor axis
b	Semiminor axis: $b = (1 - \epsilon^2)^{1/2} a$
ϵ	Eccentricity: each focus displaced from center by ϵa
α	Latus rectum: Distance of focus from point on the ellipse perpendicular to major axis: $\alpha = (1 - \epsilon^2)\, a$
r_0	Distance from the focus to the pericenter: $r_0 = (1 - \epsilon) a$
r_1	Distance from the focus to the apocenter: $r_1 = (1 + \epsilon) a$

Figure 4: Ellipse parameters (9)

For an ellipse of the form

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

The circumference of the ellipse is found by the arc length formula for parametric equations. Let $x = a\cos\theta$ and $y = b\sin\theta$. $a > b > 0$.

$$L = 4\int_0^{\pi/2} \sqrt{dx^2 + dy^2}$$

$$L = 4\int_0^{\pi/2} \sqrt{a^2\sin^2\theta + b^2\cos^2\theta}\; d\theta = 4\int_0^{\pi/2} \sqrt{a^2 - a^2\cos^2\theta + b^2\cos^2\theta}\; d\theta$$

$$= 4a\int_0^{\pi/2} \sqrt{1 - \left(1 - \frac{b^2}{a^2}\right)\cos^2\theta}\; d\theta \qquad \text{Let } k^2 = 1 - \frac{b^2}{a^2} = \frac{a^2 - b^2}{a^2} = \frac{c^2}{a^2} = \epsilon^2$$

$$L = 4a\int_0^{\pi/2} \sqrt{1 - e^2\cos^2\theta}\; d\theta$$

In the case of integrating between 0 and $\frac{\pi}{2}$, $\sin\theta$ and $\cos\theta$ are interchangeable, can be justified graphically.

$$L = 4a\int_0^{\pi/2} \sqrt{1 - e^2\sin^2\theta}\; d\theta$$

$$L = 4aE(e)$$

Therefore, for one to find the circumference of an ellipse, only the semi-major axis a and the eccentricity e are needed. The expression can be verified with the circumference of a circle, $L = 2\pi r$. For a circle, $a = r$ and $e = 0$.

$$L = 4aE(0)$$

$$= 4r\int_0^{\pi/2} \sqrt{1 - 0^2\sin^2\theta}\; d\theta$$

$$= 4r\int_0^{\pi/2} d\theta = \frac{4r\pi}{2} = 2\pi r$$

For example, $a = \frac{7\sqrt{7}}{4}$ and $e = \frac{\sqrt{6}}{3}$.

$$L = 4\left(\frac{7\sqrt{7}}{4}\right)\int_0^{\pi/2}\sqrt{1-\left(\frac{\sqrt{6}}{3}\right)^2\sin^2\theta}\;d\theta$$

Approximating the integral using the MATLAB code.

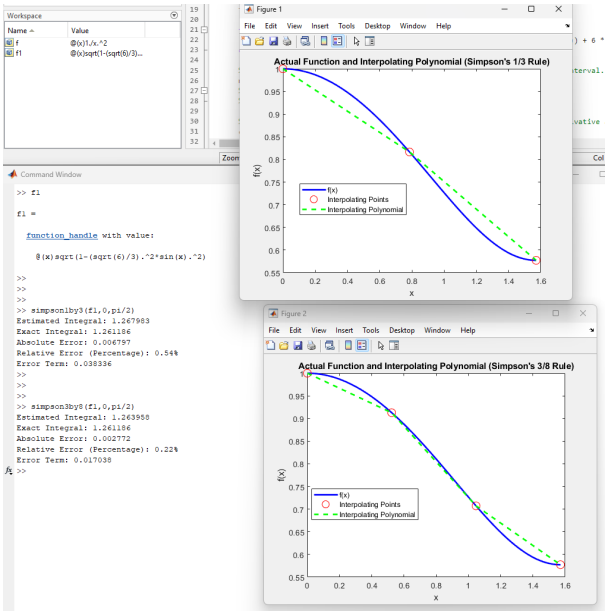


Figure 5: Approximating the integral using Simpson's Rules

Now to list out the values for L, the exact and the two approximations.

$$\text{Exact}: L = 4\left(\frac{7\sqrt{7}}{4}\right)(1.261186) = 23.35749$$

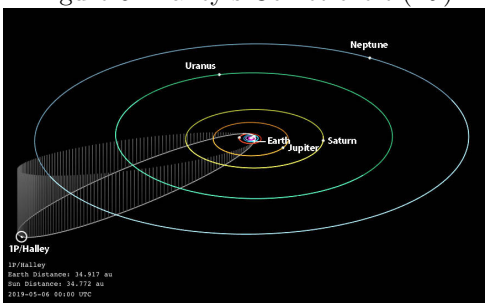
$$\frac{1}{3}\ \text{Simpson's Rule}: L = 4\left(\frac{7\sqrt{7}}{4}\right)(1.267983) = 23.48337$$

$$\frac{3}{8}\ \text{Simpson's Rule}: L = 4\left(\frac{7\sqrt{7}}{4}\right)(1.263958) = 23.40883$$

Both approximations are within one decimal place of the exact value. This shows how accurate Elliptic integrals are in finding the circumference of an ellipse even when using an approximation.

Another example is the orbital circumference of a celestial body. Halley's Comet, named after the English astronomer Edmond Halley, is a famous periodic comet with an elliptical orbit that brings it into the inner solar system approximately every 76 years. It has been observed for over two millennia, with notable sightings in 1066 and 1910.

Figure 6: Halley's Comet orbit (*10*)



Halley's Comet is characterized by its bright coma and tail, which vary in appearance during each apparition. It has been a subject of extensive scientific study, including missions like the European Space Agency's Giotto, which provided valuable data. The comet holds cultural significance, with associations ranging from omens to artistic inspiration, and its predicted return in 2061 and in 2134 is highly anticipated as one of its closest approaches to Earth (*10*). With the formula, we can find the distance the comet would travel in one full orbit. $a = 2.689\,73 \times 10^{12}\,\text{m}$ and $e = 0.967$ (*11*).

$$L = 4\left(2.689\,73 \times 10^{12}\,\text{m}\right)\int_0^{\pi/2}\sqrt{1-(0.967)^2\sin^2\theta}\;d\theta$$

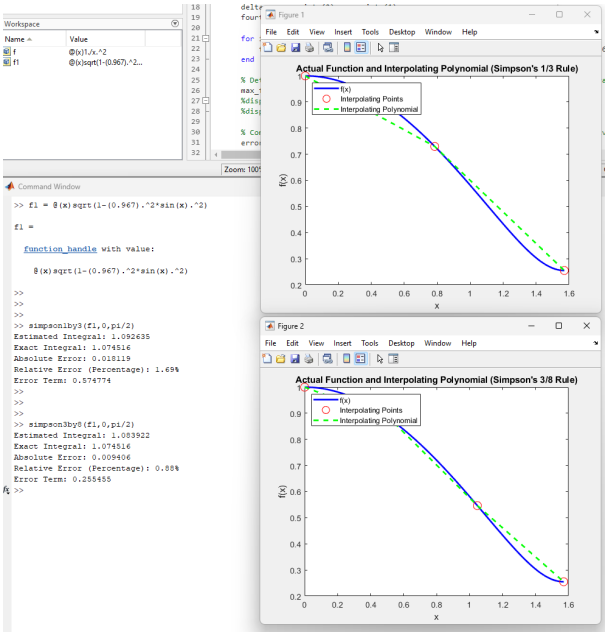


Figure 7: Approximating the integral using Simpson's Rules

$$\text{Exact}: \; L = 4\left(2.689\,73 \times 10^{12}\,\text{m}\right)(1.074516) = 1.155\,81 \times 10^{13}\,\text{m}$$

$$\frac{1}{3}\;\text{Simpson's Rule}: \; L = 4\left(2.689\,73 \times 10^{12}\,\text{m}\right)(1.092635) = 1.175\,30 \times 10^{13}\,\text{m}$$

$$\frac{3}{8}\;\text{Simpson's Rule}: \; L = 4\left(2.689\,73 \times 10^{12}\,\text{m}\right)(1.083922) = 1.165\,92 \times 10^{13}\,\text{m}$$

Similar to the first example, both approximations are within one or two decimal places of the exact value. At these scales, astrophysicists tend to evaluate their work with the orders of magnitude, and in our case they are all the same. This shows how accurate Elliptic integrals are in finding the path travelled of a celestial body even when the semi-major axis a and the eccentricity e are at large values.

Conclusion

With the help of this small project, we were able to show how Simpson's Rule can be used practically to solve difficult, computationally impossible problems that arise frequently in science and engineering. With the help of our programming skills, we were able to create a MATLAB code that allowed us to connect the dots between theoretical mathematics and practical applications. This investigation not only demonstrated the importance of Simpson's Rule in numerical integration, but it also brought to light the critical role that programming abilities play in translating abstract ideas into practical solutions for scientific and engineering problems.

References

1. R. Cardil, *Kepler: The Volume of a Wine Barrel* (2023; https://maa.org/press/periodicals/convergence/kepler-the-volume-of-a-wine-barrel-derivatives-tangents-and-slopes-conclusion).
2. Oliver Knill, *Introduction to Calculus : Numerical integration* (https://people.math.harvard.edu/~knill/teaching/math1a2021/handouts/lecture27.pdf).
3. D. J. Velleman, *The American Mathematical Monthly* **112**, Publisher: Mathematical Association of America, 342–350, ISSN: 00029890, 19300972, (2023; http://www.jstor.org/stable/30037470) (2005).
4. *Gregory formula - Encyclopedia of Mathematics* (2023; https://encyclopediaofmath.org/wiki/Gregory_formula).
5. H. M. Antia, *Numerical Methods for Scientists and Engineers* (Hindustan Book Agency, Third, 2012), (https://doi.org/10.1007/978-93-86279-52-1).
6. Richard L. Burden, J. Douglas Faires, *Numerical Analysis* (CENGAGE Learning, Ninth, 2011).
7. Riley, K., Hobson, M., Bence, S., *Mathematical Methods for Physics and Engineering: A Comprehensive Guide* (Cambridge: Cambridge University Press, Third, 2006), (doi:10.1017/CBO9780511810763).
8. M. R. Spiegel, S. Lipschutz, J. Liu, *Mathematical handbook of formulas and tables*, Section: x, 293 pages : illustrations ; 28 cm. (McGraw-Hill, New York, Fourth edition. 2013), ISBN: 978-0-07-179537-1 0-07-179537-5.
9. G. Fowles, G. Cassiday, *Analytical Mechanics* (Thomson Brooks/Cole, 2005), ISBN: 978-0-534-49492-6.
10. *955 Years Ago: Halley's Comet and the Battle of Hastings - NASA*, Section: NASA History (2023; https://www.nasa.gov/history/955-years-ago-halleys-comet-and-the-battle-of-hastings/).
11. David R. Williams, *Comet Fact Sheet* (2023; https://nssdc.gsfc.nasa.gov/planetary/factsheet/cometfact.html).