Project 02

APSC 607 Fall 2017

Seth Goodman

October 24, 2017

1 Introduction

This project explored methods for calculating the integral roots of functions. Each functions was examined in the range between zero and two, using the Composite Trapezoidal Rule, Composite Midpoint Rule, Composite Simpson's Rule, as well as an adaptive implementation of the Composite Simpson's Rule. The behavior and characteristics of these methods are reviewed by examining the effectiveness of the resulting value for the integral given a range of values for N.

All computations were performed using MATLAB using the code (Table 1) accompanying this report (in a zip file). The following Methods section will present the methods used in MATLAB to explore functions, as well as the outputs and results. The Results section of this report contains the outputs for each function and range along with related observations and discussion. All figures and tables found in this report are available in the output subdirectory of the accompanying zip file. Additionally, all code and figures found in the zip file can be accessed via GitHub[1].

[1] `https://github.com/sgoodm/apsc607/tree/master/project_02`

2 Methods

The functions examined in the project, Function **A** and **B**, are defined in Equations 1 and 2, respectively. For each of these functions, different integration rules will be tested to examine their effectiveness when used to perform composite numerical integration. Integration will be restricted to between zero and two for testing, but integration over an expanded range and the potnetial utility of adaptive approaches will be discussed in the /emphResults section.

$$f(x) = e^{2x} * sin(3x) \tag{1}$$

$$f(x) = \frac{1}{x+4} \tag{2}$$

To establish a baseline, the true value for the integral of each function is first calculated using built in MATLAB tools. The integral is calculated both using the symbolic toolkit function **int** as well as the numerical function **integral**. The resultign values can be seen in Table 1

Function	**Symbolic**	**Numeric**
A	2	2
B	2	2

Table 1: True values of integrals between zero and two

Overview of methods

2.1 Composite Trapezoidal Rule

TEXT

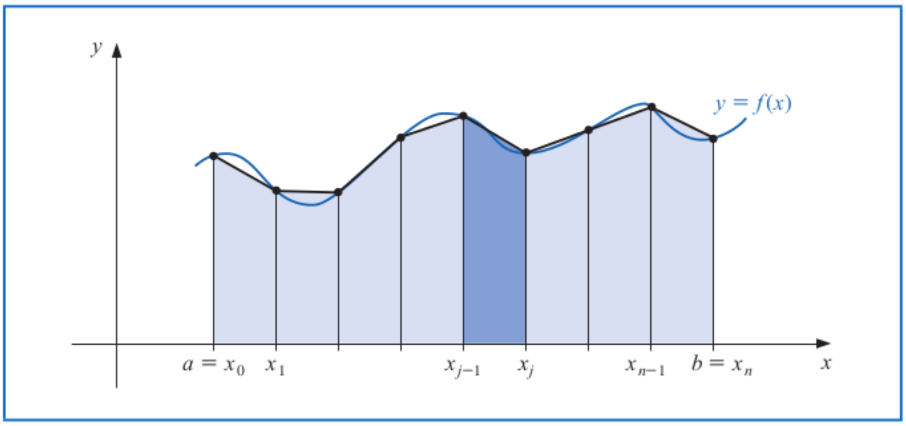

Figure 1: Trapezoidal Figure[1]

The Composite Trapezoidal Rule for n intervals, as seen in Figure 1, can be defined by Equation 3, given $h = (b-a)/n$ and $x_j = a + jh$, for each $j = 0, 1, \ldots, n$[1].

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right] \quad (3)$$

Implementing this in a MATLAB function is extremely straightforward. The function accepts a function handle **f** defining $f(x)$, (e.g., $cos(x)$), the desired number of internval **n**, along with our bounds **rmin** and **rmax**. The value for **h** given **n** is calculated, as are the vectors **j** and x_j. Using these components and the summation function, the final integral for the input conditions can be calculated. The function then returns the integral value, along with the value of **h** used.

This function implementing the Trapezoidal Rule for integration (as well as subsequent Midpoint and Simpson's Rules) is called over a range of values for **n** using the **arrayfun** function in MATLAB. This function accepts another function, defining our integration rule, and a vector, and simply repeatedly calls the specified value while iterating over the values in the vector. The call to **arrayfun** then return a vector (or set of vectors in this case) containg the results from each call to the integration function.

The resulting vector of integral values across varying **n** can compared with the true integral value generated earlier, to produce an error vector. The error vector is used to identify the value of **n** (and thus **h**) at which the integration rule produced results that were accurate within a desired tolerance.

2.2 Composite Midpoint Rule

TEXT

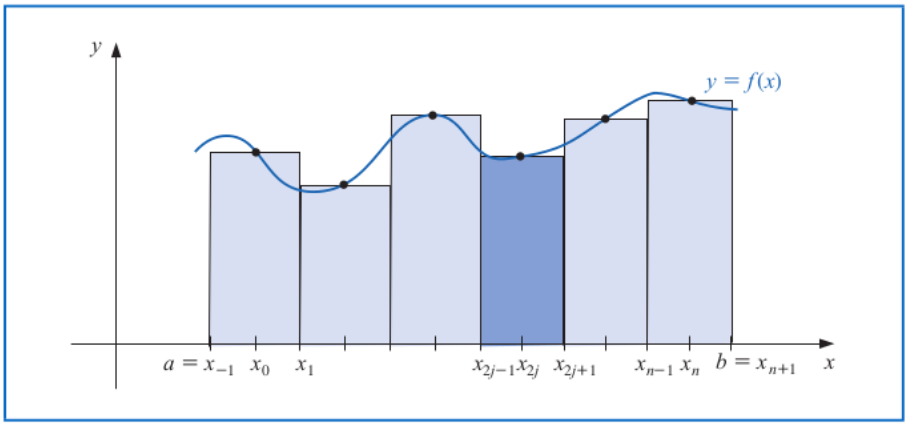

Figure 2: Midpoint Figure[1]

The Composite Midpoint Rule for $n+2$ intervals, as seen in Figure 2, can be defined by Equation 4, given $h = (b-a)/(n+2)$ and $x_j = a + (j+1)h$, for each $j = -1, 0, \ldots, n+1$[1].

$$\int_a^b f(x)dx = 2h\sum_{j=0}^{n/2} f(x_2 j) \quad (4)$$

2.3 Composite Simpson's Rule

TEXT

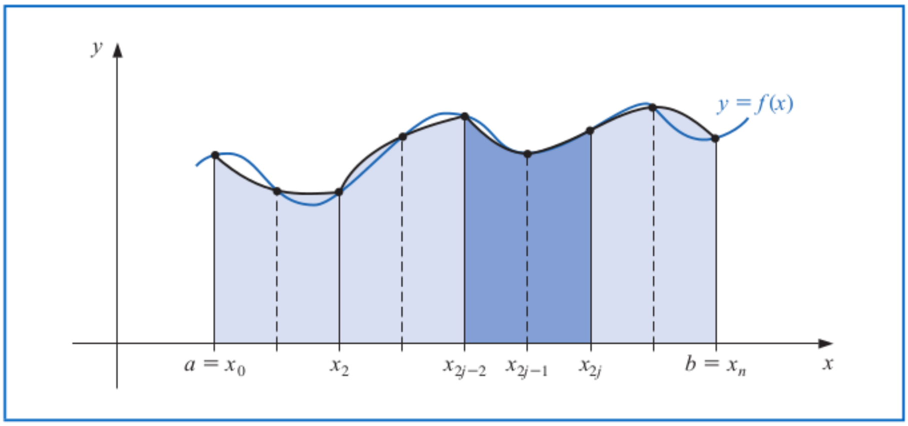

Figure 3: Simpson's Figure[1]

The Composite Simpson's Rule for n intervals, as seen in Figure 3, can be defined by Equation 5, given $h = (b-a)/n$ and $x0_j = a+jh$, $x1_j = a+jh+h/2$, $x2_j = a + jh + h$, for each $j = 0, 1, \ldots, n-1$.

$$\int_a^b f(x)dx = \sum_{j=0}^{n-1} \left[\frac{h}{6} \left[f(x0_j) + 4f(x1_j) + f(x2_j) \right] \right] \tag{5}$$

2.4 Adaptive Simpson's Rule

TEXT

3 Results

TEXT
Error comparison

$$\frac{b-a}{6}h^2 f''(u) \tag{6}$$

$$\frac{b-a}{12}h^2 f''(u) \tag{7}$$

$$\frac{h^5}{90}f^{(4)}(\xi_j) \tag{8}$$

1. Like this,
2. and like this.

- Like this,
- and like this.

Word Definition

Concept Explanation

Idea Text

. . . using dots . . .
before figure 1a

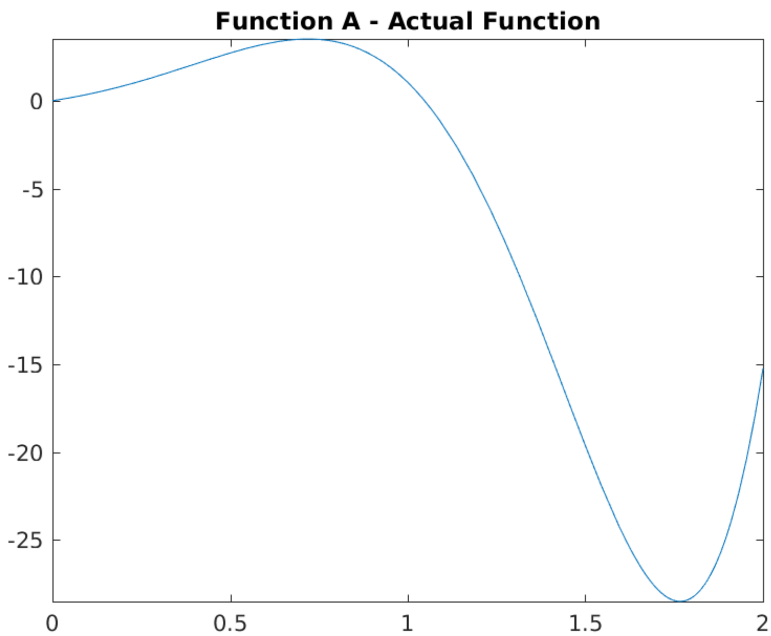

Figure 4: caption text a

after figure 1a

Item	**Quantity**
Widgets	42
Gadgets	13

Table 2: An example table.

Something about minimum error in footnote[2].

[2]See following link on MATLAB precision limitions (general limitations of floating point representations apply) https://www.mathworks.com/help/fixedpoint/ug/limitations-on-precision.html

References

[1] Burden, R., Faires, J., Numerical Analysis 9th Edition. 2010