Accuracy and Error in Polynomial Representations

Alina Enikeeva: 251160024
Erina Mitha: 251279433

January 2024

1 Abstract

Accuracy and Error in Polynomial Representations serves as a gateway to a series of four interconnected labs that delve into the world of numerical analysis. This lab explores the computation and applications of the Taylor series, which essentially allow one to represent complex mathematical functions as a summation of polynomial terms. Taylor series converge for an region around the origin, but past this radius of convergence, they are no longer an accurate representation of the function they are trying to emulate. While the addition of each subsequent polynomial term improves the accuracy of the series, unless an infinite number of terms is calculated, one will still be left with an error bound. This lab explores the absolute and relative error produced by a Taylor series and goes on to also calculate the error bound.

2 Introduction

Archimedes is credited for the "method of exhaustion"—a trick theorizing that one can find a finite solution to an infinite problem by breaking the problem down into smaller pieces (Merriam). Brook Taylor employed a similar idea when he discovered the Taylor expansion, which allows one to approximate almost any function in its entirety using an infinite number of polynomial terms. Even though calculating infinite terms for a series is essentially impossible, the Taylor series still proves useful with a finite number of terms. These polynomial representations prevail by allowing one to still find very accurate approximations of an original function, simply with the restriction that the series will only converge on a certain interval. Variants of this theorem were also discovered by James Gregory, Johann Bernoulli, Issac Newton, Gottfried Leibniz, and Abraham de Moivre (O'Connor).

This lab belongs to a four part series that experiments with various introductory numerical analysis concepts. Accuracy and Error in Polynomials explores the behaviour, error, and applications of Taylor expansions in particular.

These polynomial representations can be incredibly useful. Mathematically speaking, they tend to be much easier to work with as they are all smooth, continuous, and hence differentiable. Additionally, finding the Taylor expansion of a function can make subsequent transformations much easier. For example, if one wanted to integrate $x * e^x$, this would require a technique called integration by parts. If the Taylor representation for e^x (a polynomial expression) was used instead of the original function, this general term could be multiplied by x, resulting in a new general polynomial term which would be much easier to integrate. As functions become increasingly complex, having these polynomial representations for integration, differentiation, and general transformations becomes even more important.

Calculating the Taylor series of a function, while very useful, tend to leave us with a degree of error since it is an attempt at the ”method of exhaustion”, trying to solve an infinite problem with a finite solution. This lab explores the calculations of the error, investigating how well a Taylor expansion can truly approximate a function. One measure for this accuracy is the Lagrange remainder. Instead of giving the error itself, the Lagrange remainder provides a bound (“Lagrange Remainder”), that reflects the accuracy of the present solution and whether additional iterations are required to attain the desired level of precision. The rate of convergence using the error bound allows to efficiently compare various algorithms.

3 Problem 0

This laboratory exercise, delves into the practical applications of truncated (finite) Taylor series for approximating functions. It also explores the utility, limitations, and interplay between truncation error and roundoff error. By analyzing the behavior of the truncated series and comparing it with that of the true function, the aim is to gain a deeper understanding of approximation techniques and their implications in computational mathematics.

The truncated Taylor series of a function $f(x)$ about a point x_0 serves as an approximation to $f(x)$ using only a finite number of terms. The formula for the truncated Taylor series $T_n(x)$ is given by:

$$T_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

where $f^{(k)}(x_0)$ denotes the k-th derivative of $f(x)$ evaluated at x_0. This series provides an approximation to $f(x)$ within a neighborhood of x_0, with the accuracy improving as the number of terms n increases (Sauer).

Accompanying this approximation is the remainder term R_n, representing the discrepancy between the truncated series and the true function. The remainder term is given by:

$$R_n = \frac{f^{(n+1)}(z)}{(n+1)!}(x - x_0)^{n+1}$$

where z lies between x and x_0, and attempts to give a "worst-case scenario" for the error (Sauer). This term provides a bound for error introduced when truncating a series after n terms. The remainder also provides insights into the convergence behavior of the approximation, as it is a quantitative representation of how accurate a Taylor series is in approximation the original function.

If one were to take the function e^{-x} about $x_0 = 0$, the general term for the Taylor series could be evaluated. Since $\frac{df}{dx}e^{-x} = -e^{-x}$ and $\frac{df}{dx} - e^{-x} = e^{-x}$, this would mean the $f^{(k)}(x_0)$ in the general form of the Taylor series (shown above) would evaluate to $e^0 = 1$ for even values of k and $e^0 = -1$ for odd values of k. Hence for the function e^{-x}, $f^{(k)}(x_0) \Rightarrow (-1)^k$. Since the series will be centered at $x = 0$, $(x - x_0)^k \Rightarrow x^k$ for the general term. Lastly, the $k!$ term remains the same as this term is not dependent on the e^{-x} function. This then yields the general term for the Taylor series of $e{-}x$ to be:

$$T_n(x) = \sum_{k=0}^{n} \frac{(-1)^k}{k!}(x)^k$$

.

The form of the remainder term for the Taylor series of e^{-x} about $x_0 = 0$ is calculated with a similar logic. The $f^{(n+1)}(z)$ in the remainder formula (shown above) would yield $(-1)^{(n+1)}e^{-x}$. This is because, as calculated above, the evaluation of the general term of the Taylor series produces an alternator $((-1)^{n+1})$ and this will be multiplied by the original function (e^{-x}) to produce the "worst-case scenario" for the error bound. Similar to the general term, since the series will be centered at $x = 0$, $(x - x_0)^k \Rightarrow x^k$ for the remainder term. The $(n + 1)!$ term remains the same as it is not dependent on the function e^{-x}. This produces the remainder term of the form:

$$R_n = \frac{(-1)^{(n+1)}e^{-z}}{(n+1)!}x^{n+1}$$

4 Problem 1

In this section we produce the results and truncation error for Taylor series evaluated above value 0 as point where the derivatives are considered. We have the function:

$$F(x) = \frac{e^{-x} - 1 + x}{x^2}$$

To find the Taylor series with $n+1$ terms and the associated truncation error for $F(x) = \frac{e^{-x}-1+x}{x^2}$, we will substitute the series $e^{-x} = \sum_{k=0}^{n} \frac{(-1)^k x^k}{k!}$ into $F(x)$ and then simplify. First, $F(x)$ in terms of e^{-x}:

$$F(x) = \frac{e^{-x} - 1 + x}{x^2}$$

Now, substitute the series expansion of e^{-x}:

$$F(x) = \frac{\left(\sum_{k=0}^{n} \frac{(-1)^k x^k}{k!}\right) - 1 + x}{x^2}$$

This simplifies to:

$$F(x) = \frac{\sum_{k=0}^{n} \frac{(-1)^k x^k}{k!} - 1 + x}{x^2}$$

Now, we needed to simplify this expression. We can do this by separating the terms in the numerator and then combining like terms:

$$F(x) = \frac{\sum_{k=0}^{n} \frac{(-1)^k x^k}{k!}}{x^2} - \frac{1}{x^2} + \frac{x}{x^2}$$

$$F(x) = \sum_{k=0}^{n} \frac{(-1)^k x^k}{k! x^2} - \frac{1}{x^2} + \frac{1}{x}$$

$$F(x) = \sum_{k=0}^{n} \frac{(-1)^k x^{k-2}}{k!} - \frac{1}{x^2} + \frac{1}{x}$$

Further, series with $n+1$ terms:

$$F(x) = \frac{(-1)^0 x^{-2}}{0!} + \frac{(-1)^1 x^{-1}}{1!} + \frac{(-1)^2 x^0}{2!} + \ldots + \frac{(-1)^n x^{n-2}}{n!} - \frac{1}{x^2} + \frac{1}{x}$$

This can be simplified further, but this expression already gives the Taylor series with $n+1$ terms for $F(x)$ without using the general formula for Taylor series. Now, for the remainder (truncation error) term, it is usually given by the $n+1$ term in the Taylor series, which is the term with $k = n$:

$$R_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!}(x - x_0)^{n+1}$$

Where $f^{(n+1)}(c)$ is the $(n+1)$-th derivative of $f(x)$ evaluated at some point c between x and x_0. Since we are dealing with $f(x) = e^{-x}$, its $(n+1)$-th derivative is also e^{-x} with a sign change. So, the remainder term would be:

$$R_n(x) = \frac{(-1)^{n+1} e^{-c}}{(n+1)!} x^{n+1}$$

This expression gives the truncation error associated with the Taylor series approximation of $F(x)$.

5 Problem 2

This section explores the implementation of the Taylor series for:

$$F(x) = \frac{e^{-x} - 1 + x}{x^2}$$

First it was important to find the general term for F(x) which can be written as:

$$F(x) = \sum_{k=0}^{n} \frac{(-1)^k}{(2+k)!} x^k$$

The general form is derived as follows:Given:

$$F(x) = \frac{e^{-x} - 1 + x}{x^2}$$

First, we write the series expansion for e^{-x}:

$$e^{-x} = \sum_{k=0}^{\infty} \frac{(-1)^k x^k}{k!}$$

Substituting this into $F(x)$, we get:

$$F(x) = \frac{\sum_{k=0}^{\infty} \frac{(-1)^k x^k}{k!} - 1 + x}{x^2}$$

Now, we rewrite the series for $F(x)$ with the first n terms:

$$F(x) = \frac{\sum_{k=0}^{n} \frac{(-1)^k x^k}{k!} - 1 + x}{x^2}$$

$$F(x) = \frac{\sum_{k=0}^{n} \frac{(-1)^k x^k}{k!}}{x^2} - \frac{1}{x^2} + \frac{x}{x^2}$$

$$F(x) = \sum_{k=0}^{n} \frac{(-1)^k x^k}{k! x^2} - \frac{1}{x^2} + \frac{1}{x}$$

$$F(x) = \sum_{k=0}^{n} \frac{(-1)^k x^{k-2}}{k!} - \frac{1}{x^2} + \frac{1}{x}$$

We simplify this further:

$$F(x) = \sum_{k=0}^{n} \frac{(-1)^k x^{k-2}}{k!} - \frac{1}{x^2} + \frac{1}{x}$$

We notice that x^{k-2} starts from x^{-2} when $k = 0$. So we can rewrite the series starting from $k = 2$:

$$F(x) = \sum_{k=2}^{n} \frac{(-1)^k x^{k-2}}{k!} - \frac{1}{x^2} + \frac{1}{x}$$

So if we consider the series starting from $k = 0$, then the series' general form of $F(x)$ is:

$$F(x) = \sum_{k=0}^{n} \frac{(-1)^k}{(2+k)!} x^k$$

The graph for F(x) and its Taylor representation are shown in Plot 2.1 and the corresponding code is given in Figure 2.1. Over the interval $x \in (-0.01, 0.01)$, the two curves look essentially identical, illustrating how well a Taylor expansion can approximate a function over a given interval.

6 Problem 3

In the first part of this section, one can examine what happens when the interval of x is significantly reduced for the plot produced in the previous section. When the range is changed from $x \in (-0.01, 0.01)$ to $x \in (-10-6, 10-6)$, the curves that once appeared indistinguishable are now noticeably different (see Plot 3.1 created by the code seen in Figure 3.1) This observation tells us that although Taylor expansions can be very good approximations for a function, they still are not exact and present a degree of error.

Plots of the absolute and relative error can be seen in Plot 3.2 (created by the code seen in Figure 3.2). This plot also presents the comparison between F(x) and its Taylor representation. It appears that the Taylor expansion of F(x) oscillates around the original function, producing error graphs that also oscillate. The error appears to be greatest near, but not at the center of the series $x = 0$. For values of x further away from the center in both the positive and negative directions, the error appears to decrease exponentially. When observing the error bounds for the Taylor expansion of F(x) as seen in Plot 3.3 (corresponding to the code presented in Figure 3.3), noticeably there is no error at $x = 0$. The error bound logarithmically increases for x values moving away from the center in both the positive and negative directions. The error bound observably begins to level out. Plot 3.3 provides a good visual representation of how the first handful of terms generated for a Taylor series are the most crucial in ensuring the representation is accurate enough to use as an approximation. The addition of every subsequent term will then make the Taylor series more accurate to the original function, but terms progressively become less important in ensuring accuracy as the error bound levels off. Taken together, the results found in Plot 3.3 correspond directly to those seen in Plot 3.2.

7 Problem 4

This section creates a function called "myfofx" that plots the Taylor representation of F(x) for $x \in (-10^{-6}, 10^{-6})$ and plots the original F(x) for all other values of x. The Taylor representation is used over a small interval containing the center because this is where the approximation of the original function is most accurate. Plot 4.1 and Plot 4.2 (corresponding to the code in Figure 4.1 and Figure 4.2, respectively) illustrate the plots of myfofx over both a large and a small interval.

Main Function (myfofx):

- `myfofx(x)`: Computes the value of the function for a given input x. It determines the appropriate method (Taylor series or explicit expression) based on the magnitude of x.
- For small values of x (absolute value less than 10^{-6}), it utilizes a Taylor series approximation with the helper function `fTaylorPoly(x, n)`.
- For large values of x (absolute value greater than or equal to 10^{-6}), it directly evaluates the function using the explicit expression via the helper function `myfx(x)`.

Helper Functions:

- `evaluate_f(x)`: Determines the method of computation based on the value of x and calls either the Taylor series approximation or explicit expression evaluation.
- `myfx(x)`: Computes the function using the explicit expression $\frac{e^{-x}+x-1}{x^2}$.
- `fTaylorPoly(x, n)`: Calculates the Taylor series approximation with n terms for the given x.

8 Summary and Conclusions

The biggest takeaway from this laboratory exercise is that Taylor representations can be incredibly useful over a small interval, however they are not an exact replacement for the original function. Even if over a greater range a Taylor series appears identical to its original function, if one were to zoom in, the error in the approximations becomes abundantly clear. While this error is inevitable, depending how much precision is required for a particular problem, Taylor representations can still be much easier to work with than some complex functions. When examining the error bound, it increases logarithmically away from the center point in both the positive and negative directions and eventually levels off. This is a good representation of how the first couple of terms in a Taylor series prove to be the most important in ensuring an accurate approximation of the original function. One application of the Taylor series is to be part of a piece-wise function, whereby the series is used over its interval of convergence and the original function is used for the mapping the rest of the domain. This could potentially make calculations easier over the interval of convergence of the Taylor representation, since one would only have to work with polynomial terms, without losing the integrity of the original function outside of this interval. Taylor approximations continue to be a very useful tool, but should be used with caution as the effects of truncation error should also be taken into consideration when producing results. It would be interesting to see how much more extensive these applications could be when extending these findings to Maclaurin series (Taylor series centered at $x \in R$, where $x \neq 0$) or to other areas of numerical analysis.

9 References:

"Lagrange Remainder." Lagrange Remainder, Math Open Reference, 2011, https://www.mathopenref.com/calclagrange.html.

Merriam, Areeba. "History of the Taylor Series." Medium, Cantor's Paradise, 1 Oct. 2023, www.cantorsparadise.com/history-of-the-taylor-series-dc173b5836fe.

O'Connor, J J, and E F Robertson. "Brook Taylor - Biography." MacTutor Index, University of St. Andrews, May 2000, mathshistory.st-andrews.ac.uk/Biographies/Taylor/.

Sauer, Tim. Numerical Analysis. Pearson, 2018.

10 Teamwork Statement

Work for this laboratory was done split evenly between both partners. Both members worked on the coding, analysis, and formatting of this report with each person working on the areas that best played to their strengths. One problem encountered when putting together this report was not being able to simultaneously work on the same Matlab file. This was resolved by having each person work on the problems individually and then copying and pasting code into a shared Google Doc. This way, both partners could compare code and help each other when stuck on a particular section. Together, both members were able to effectively collaborate to work through all the provided laboratory exercises.

11 Code Appendix

```
% PROBLEM 2

% PLOTTING THE FUNCTIONS
%Define the domain to be over the interval (-0.01, 0.01) using 50 points
x = linspace(-0.01, 0.01, 50);
y = fofx(x); % Calculate F(x)
% Calculate the Taylor Series for the first 4 terms of F(x)
Taylor = fTaylorPoly(x, 4);

figure(1); % Create a new figure
plot(x,y,'b--o',x,y,'r'); % Plot F(x)
plot(x,Taylor,'b--o',x,y,'r'); % Plot the Taylor Polynomial for F(x)
legend({'T = T(x)','y = F(x)'},'Location','southwest'); % Create a legend
% Add a title
title('Problem 2: Plot of F(x) and the Taylor Series of F(x)');
xlabel('x'); % Add axis label for x
ylabel('y'); % Add axis label for y

% DEFINING THE FUNCTIONS
% Modified Taylor Series Polynomial Calculator from Lab1A
function T = fTaylorPoly(x, n)
    % Initialize sum as 1/2, since the first term of the series is 1/2
    T = 1/2;

    for k = 1:n % loop over terms in series
        % Add the general term for each n to the total sum
        T = T + (((-1).^k) * (x.^k)) ./ (factorial(2 + k));
    end % end for loop
end % end function

% Define F(X) function
function y = fofx(x)
    y = (exp(-x) - 1 + x) ./ (x.^2);
end
```

Figure 2.1: The code for Problem 2, including the functions that define F(x) and the Taylor Series for F(x)

```
% PROBLEM 3

% PART A - change the range of the plot from Problem 2

% PLOTTING THE FUNCTIONS
%Define the domain to be over the interval (-0.01, 0.01) using 50 points
x = linspace(-10^-7, 10^7, 50);
y = fofx(x); % Calculate F(x)
% Calculate the Taylor Series for the first 4 terms of F(x)
Taylor = fTaylorPoly(x, 4);

figure(1); % Create a new figure
plot(x,y,'b--o',x,y,'r'); % Plot F(x)
plot(x,Taylor,'b--o',x,y,'r'); % Plot the Taylor Polynomial for F(x)
legend({'T = T(x)','y = F(x)'},'Location','southwest'); % Create a legend
% Add a title
title('Problem 3: Plot of F(x) and the Taylor Series of F(x) [smaller domain]');
xlabel('x'); % Add axis label for x
ylabel('y'); % Add axis label for y

% PART B - PLOT THE ABSOLUTE AND RELATIVE ERRORS
xInterval = [-10^(-6),10^(-6)]; % Define the interval
figure; % Create a new figure
```

Figure 3.1: The code for Problem 3 Part A which is similar to Figure 2.1, but shows the functions over a smaller range

```
% PART B - PLOT THE ABSOLUTE AND RELATIVE ERRORS
xInterval = [-10^(-6),10^(-6)]; % Define the interval
% Find F(X)
mainF = @(x) fofx(x);
figure; % create a new figure
subplot(3,1,1); % create three plots
fplot(mainF, xInterval, 'b'); % Plot F(x) with blue color
hold on;
% Find the Taylor Series for F(X)
yTayF = @(x) fTaylorPoly(x, 3);
fplot(yTayF, xInterval, 'r--'); % Plot Taylor series with red dashed line
legend("F(x)", "Taylor Series");
xlabel('x');  % Add axis label for x
ylabel('y'); % Add axis label for y
title('Comparison of F(x) and its Taylor Series');

% Plot Absolute Error (on a log scale)
subplot(3,1,2); % create plot for absolute error
absErr = @(x) abs(yTayF(x) - mainF(x)); % define absolute error
fplot(absErr, xInterval, 'g'); % Plot absolute error with green color
xlabel('x');  % Add axis label for x
ylabel('Absolute Error'); % Add axis label for y
title('Absolute Error between F(x) and its Taylor Series'); % add title
set(gca, 'YScale', 'log'); % Set y-axis to logarithmic scale
xlim(xInterval); % Adjust plot limits for better visibility
 % Plot Relative Error
 subplot(3,1,3); % create plot for relative error
 relErr = @(x) absErr(x) ./ mainF(x); % define relative error
 fplot(relErr , xInterval, 'r'); % Plot absolute error with green color
 xlabel('x'); % Add axis label for x
 ylabel('Relative Error'); % Add axis label for y
 title('Relative Error between F(x) and its Taylor Series'); % add title
```

Figure 3.2: The code for Problem 3 Part B which plots the relative and absolute errors of F(x) and its Taylor Representation and shows a comparison between the two functions

```
% PART C - BOUNDS FOR THE ERROR
xInterval = [-10^(-6),10^(-6)]; % Define the interval
% Find the Taylor Series for F(X)
yTayF = @(x) fTaylorPoly(x, 3);
% Plot Absolute Error (on a log scale)
figure; % create a new figure
hold on; % continue to plot on the same figure
% Bound for x > 0 (using alternating series error bound)
bound_x_pos = @(x) abs((-1)^4 * (x^4) / factorial(6)); % Bound for x > 0, truncate after 3 terms
% Bound for x < 0 (assuming derivative behavior)
bound_x_neg = @(x) abs(x.^4 / factorial(6)); % A simple bound, may not be tight
% Plot bounds
fplot(bound_x_pos, [0, 1e-6], 'r--'); % Plot bound for x > 0
fplot(bound_x_neg, [-1e-6, 0], 'b--'); % Plot bound for x < 0
xlabel('x');
ylabel('Absolute Error');
title('Absolute Error and Bounds for F(x) Taylor Series');
set(gca, 'YScale', 'log'); % Set y-axis to logarithmic scale
% Adjust plot limits for better visibility
xlim(xInterval);
```

Figure 3.3: The code for Problem 3 Part C which plots the bounds for the error between F(x) and its Taylor Representation

```
% DEFINING THE FUNCTIONS
function y = fofx(x) % define given F(x) function
    y = (exp(-x) - 1 + x) ./ (x.^2);
end

% Modified Taylor Series Polynomial Calculator from Lab1A
function T = fTaylorPoly(x, n)
    T = 1/2; % Initialize sum as 1/2 since 1/2 is the first term
    for k = 0:n % loop over terms in series
         % Add the general term for each n to the total sum
        T = T + ( ( (-1).^k) * (x.^k) )./ (factorial(2 + k) );
    end % end for loop
end % end function
```

Figure 3.4: The code for the functions used in Problem 3 Parts A, B, and C

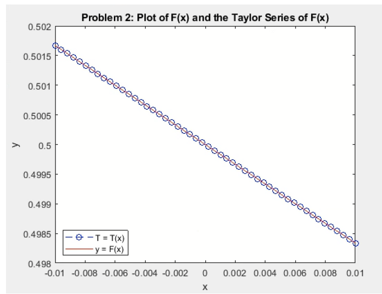
```
% PROBLEM 4

% MYFOFX Computes the value of the function using Taylor series for small
% x, and explicit expression for large x.
%   Inputs:
%       x: The input value(s) at which to evaluate the function.
%   Output:
%       fx: The value(s) of the function at the given x.

function fx = myfofx(x)
% Define the range of x values
x_range_small = linspace(-0.01, 0.01, 50); % For the interval (-0.01, 0.01)
% Call myfofx to compute the corresponding y values for the small range
y_values_small = zeros(size(x_range_small));
for i = 1:length(x_range_small)
   y_values_small(i) = evaluate_f(x_range_small(i));
end
% Plot the result for the small range in red
figure;
plot(x_range_small, y_values_small, "o");
xlabel('x'); % create x label
ylabel('F(x)'); % create y label
title('Plot of myfofx for -0.01 < x < 0.01'); % create title
% Define the range for the interval (-10^(-6), 10^(-6))
x_range_large = linspace(-10^(-6), 10^(-6), 50);
% Call myfofx to compute the corresponding y values for the large range
y_values_large = evaluate_f(x_range_large);

figure; % Plot the result for the large range
plot(x_range_large, y_values_large);
xlabel('x'); % create x label
ylabel('F(x)'); % create y label
title('Plot of myfofx for -10^{-6} < x < 10^{-6}'); % create title
fx = [y_values_small, y_values_large]; % Combine the results
end
```

Figure 4.1: The code for the fofx function used in Problem 4

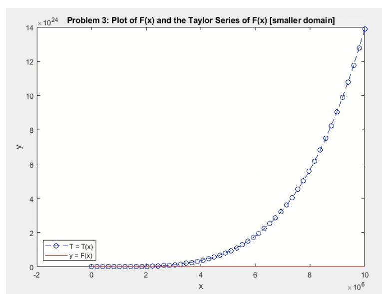
```
function y = evaluate_f(x)
    % Evaluate f(x) using the appropriate method based on x
    if abs(x) < 10^(-6)
        % Use Taylor series approximation for small x
        y = fTaylorPoly(x, 8);
    else
        % Use explicit expression for large x
        y = myfx(x);
    end
end

function y = myfx(x) % define the given F(x) function
   y = (exp(-x) + x - 1) ./ (x.^2);
end % end F(x) function

function T = fTaylorPoly(x, n)
    T = 0; % Initialize sum as 0
    % Loop over terms in series

    for k = 2:n % Add the general term for each n to the total sum
        T = T + (-1)^(k)*(x).^(k-2) ./ factorial(k);
    end % end loop
end % end fTaylorPoly
```

Figure 4.2: The code for the supporting functions for fofx function used in Problem 4

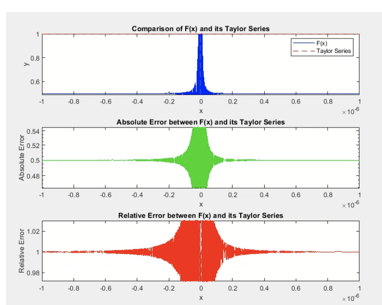
12 Plot Appendix



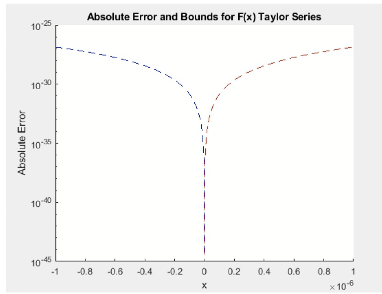
Plot 2.1: The plot of F(x) and the Taylor Representation of F(x) for Problem 2



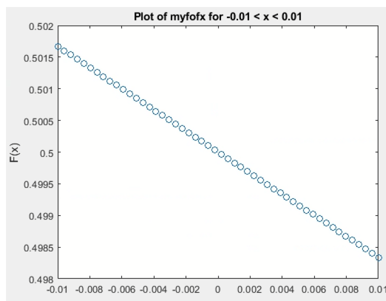
Plot 3.1: The plot of F(x) and the Taylor Representation of F(x) over a smaller range for Problem 3



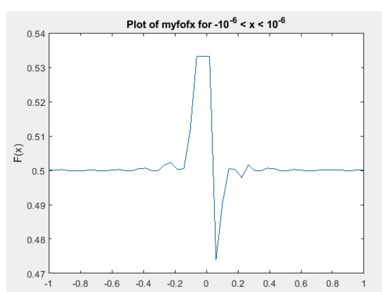
Plot 3.2: The plot of the absolute and relative error between F(x) and its Taylor Representation, as well as a comparison of the two functions for Problem 3



Plot 3.3: The plot of the error bound for $x < 0$ and $x > 0$ for F(x) and its Taylor Representation for Problem 3



Plot 4.1: The plot of the myfofx function on the smaller range for Problem 4



Plot 4.2: The plot of the myfofx function on the bigger range for Problem 4