Taylor and Maclaurin Polynomials

Shubha Swarnim Singh

September 6th, 2024

Dr. Brandy Wiegers

CSC – 455 Numerical Computation

1. Introduction

This report aims to analyze and approximate the function $f(x) = e^x \cos(x)$ using a 3rd-degree Maclaurin polynomial centered at $x_0 = 0$ and a 3rd-degree Taylor polynomial centered at $x_0 = 2$. The approximation techniques are crucial for understanding how well these polynomials can estimate the function's behavior over a given interval. This process will allow us to compare the accuracy of both approximations at various points and assess the errors associated with the approximations in the interval [0, 2].

This analysis includes constructing the Maclaurin polynomial $M_3(x)$, and the Taylor polynomial $T_3(x)$, and computing the actual function values at specific points to determine the degree of error.

2. Maclaurin Polynomial Approximation

We first construct the 3rd-degree Maclaurin polynomial $M_3(x)$ for the function $f(x) = e^x \cos(x)$, centered at $x_0 = 0$. The general form of the Maclaurin series is:

$$M(x) = f(x_0) + f^1(x_0)(x - x_0) + \frac{f^2(x_0)(x - x_0)^2}{2!} + \frac{f^3(x_0)(x - x_0)^3}{3!}$$

After calculating the necessary derivatives of the function f(x), we obtained:

- $f(0) = e^0 \cos(0) = 1$
- $f'(0) = e^x (\cos(x) - \sin(x)) = 1$
- $f''(0) = e^x (-2 \sin(x)) = 0$
- $f'''(0) = e^x (-3 \sin(x) - \cos(x)) = -2$

Thus, the 3rd-degree Maclaurin polynomial becomes:

$$M_3(x) = 1 + x - \frac{x^3}{6}$$

This polynomial will be tested over a range of values for x and compared to the actual values of the function f(x).

Using $M_3(x)$ at x = 0.5:

$$M_3(0.5) = 1 + 0.5 - \frac{(0.5)^3}{6} = 1.5 - \frac{0.125}{6} = 1.47917$$

The true value of f (0.5) is approximately f (0.5) = $e^{0.5}$ cos (0.5) ≈ 1.357.

The absolute error is:

$$|f(0.5) - M_3(0.5)| = |1.357 - 1.47917| = 0.12217$$

Error Calculation:

To find an error bound for the 3rd-degree Maclaurin polynomial $M_3(x)$ in its approximation of f(x) = e^x cos(x) over the interval [0, 2], we will use the remainder term in the Taylor series approximation, also known as the Lagrange remainder. For a 3rd-degree Maclaurin polynomial $M_3(x)$, the error $R_3(x)$ is given by:

$$R_3(x) = \frac{f^4(\xi)}{4!} x^4$$

where:

- $f^4(\xi)$ is the 4th derivative of f(x) = e^x cos(x),

- ξ is some point between 0 and x (this value is typically unknown but can be bounded).

For the 3rd-degree polynomial, the 4th derivative of f (x):

$$f^4\,(x) = e^x\,(-4cos\,(x))$$

For the error bound, we need to estimate f^4 (x) for x in [0, 2]. Since e^x cos(x) and e^x sin(x) are bounded on this interval, we can use the maximum possible value of | f^4 (x) | in the interval to get an error bound. To estimate the maximum of | f^4 (x) |, we evaluate f^4 (x) at key points x = 0, x = 2, and possibly use numerical methods or graphing tools like Desmos or a Python script to find the maximum value of | f^4 (x) |in the interval.

At x = 0:

$$f^{4}(0) = -4e^{0}\, cos\,(0) + e^{0}\, sin\,(0)$$

At x = 2:

$$f^{4}\,(2) = -4e^{\,2} cos\,(2) + e^{2}\, sin\,(2)$$

Given that e^x grows rapidly, and cos(x) and sin(x) oscillate, f^4 (x) will reach larger values as x increases. We can approximate the maximum value of | f^4 (x) | on [0, 2] using a calculator or Python to find that it's around 60.

Using m = 60 as the maximum value of | f^4 (x) |, we can now compute the error bound for x in [0, 2]:

$$R_3(x) \leq \frac{60}{4!}\, x^4 = \frac{60}{24}\, x^4 = 2.5x^4$$

Let's test x = 1 and compare the actual error with the bound.

- Error bound at x = 1:

$$R_3(1) \leq 2.5 * 1^4 = 2.5$$

- Actual error at x = 1:

We can compute the actual error by finding the difference between the true value of f(1) and the approximation $M_3(1)$:

$$f(1) = e^1 \, cos(1) \approx 1.4687$$

$$M_3(1) = 1 + 1 + \frac{1^2}{2} + \frac{1^3}{6} = 1 + 1 + 0.5 + 0.1667 = 2.6667$$

- Actual error:

$| f(1) – M_3(1) | = | 1.4687 - 2.6667 | \approx 1.198$

The actual error of 1.198 is smaller than the error bound of 2.5, as expected. This shows that the Lagrange remainder provides a useful upper bound for the error in the approximation.

3. Taylor Polynomial Approximation

Next, we construct the 3rd-degree Taylor polynomial centered at $x_0 = 2$. The general form of the Taylor polynomial is:

$$T_3(x) = f(2) + f'(2)(x-2) + \frac{f''(2)}{2!}(x-2)^2 + \frac{f'''(2)}{3!}(x-2)^3$$

To obtain the Taylor polynomial, we calculated the function and its derivatives at $x_0 = 2$:

- f (2) = e^2 cos (2) ≈ −3.0751
- f' (2) = e^2 (cos (2) – sin (2)) ≈ − 9.791
- f'' (2) = e^2 (-2 sin (2)) ≈ −16.507
- f''' (2) = e^2 (-3 sin(2) – cos(2)) ≈ −17.085

By substituting these values into the Taylor series, we constructed $T_3(x)$, which will also be evaluated at the same points as $M_3(x)$.

The 3rd-degree Taylor polynomial is:

$$T_3(x) = -3.0751 - 9.791(x-2) - 8.2535(x-2)^2 - 2.8475(x-2)^3$$

Using $T_3(x)$ at x = 1:

$$T_3(1) = 1.214$$

The true value of f (1) is approximately f (1) = e^1 cos (1) ≈ 1.460.

The absolute error is:

$$|f(1) - T_3(1)| = |1.460 - 1.214| = 0.246$$

Error Calculation:

To find an error bound for the 3rd-degree Taylor polynomial $T_3(x)$ in its approximation of f(x) = e^x cos(x) over the interval [0,2], we will use the remainder term in the Taylor series approximation, also known as the Lagrange remainder. For a 3rd-degree Taylor polynomial $T_3(x)$, the error $R_3(x)$ is given by:

$$R_3(x) = \frac{f^4(\xi)}{4!} \cdot (x - 0)^4$$

where:

- $f^4(\xi)$ is the 4th derivative of f(x) = e^x cos (x),
- ξ is some point between 0 and x (this value is typically unknown but can be bounded).

We already have:

$$f^4\ (x)\ =\ e^x\ (-\ 4\ cos(x))$$

At x = 0:

$$f^4\ (0)\ =\ e^0\ (-4\ cos\ (0))\ =\ -4$$

At x = 2:

$$f^4\ (2) =\ e^2\ \left(-4\ cos\ (2)\right) =\ -4e^2 \cos(2) =\ -12.4$$

Given that e^x grows rapidly, and cos(x) oscillate, f^4 (x) will reach larger values as x increases. We can approximate the maximum value of | f^4 (x) | on [0, 2] using a calculator or Python to find that it's around 30.

Using m = 30 as the maximum value of | f^4 (x) |, we can now compute the error bound for x in [0, 2]:

$$R_3(x) \leq \frac{30}{4!}\, x^4 = \frac{30}{24}\, x^4 \;=\; 1.25x^4$$

Let's test x = 1 and compare the actual error with the bound.

- Error bound at x = 1:

$$R_3(1) \leq\; 1.25 * \;1^4 \;=\; 1.25$$

- Actual error at x = 1:

We can compute the actual error by finding the difference between the true value of f (1) and the approximation $T_3(1)$:

$$f(1) \;=\; e^1\, cos(1) \approx 1.4687$$

$$T_3(1) \;= 1 - 1 + \frac{1^2}{2} - \frac{1^3}{6} = \frac{1}{3} \approx 0.333$$

- Actual error:

| f (1) – $T_3(1)$ | = | 1.4687 – 0.33 | ≈ 1.135

The actual error of 1.135 is smaller than the error bound of 1.25, as expected. This shows that the Lagrange remainder provides a useful upper bound for the error in the approximation.

4. Comparison of Function Values and Polynomial Approximations for $f(x)=e^x \cos(x)$ at Specified Points

Here is the table summarizing, the values for f (x), $M_3(x)$, and $T_3(x)$ at the provided x-values

x	$f(x)=e^x \cos(x)$	$M_3(x)$	$T_3(x)$
-1	0.198	0.333	-1.3679
0	1	1	-0.6456
0.5	1.447	1.485	0.5977
0.75	1.549	1.609	1.0414
1	1.469	1.667	1.2146
1.5	0.317	1.375	0.2940
2	-3.075	0.333	-3.0749
15	-2483432.984	-1109	-3934.4418

5. Predictions about the M_3 and T_3 approximations

Based on the analysis, both the 3rd-degree Maclaurin polynomial ($M_3(x)$) and the 3rd-degree Taylor polynomial ($T_3(x)$) provide accurate approximations of $f(x) = e^x \cos(x)$ near $x = 0$. However, their accuracy decreases as x moves away from 0. At larger values like $x = 2$ and $x = 15$, the deviation between the actual function values and the approximations becomes significant. This indicates that while $M_3(x)$ and $T_3(x)$ are useful locally, their effectiveness diminishes for larger x.

6. Computer Analysis

The python code that I used to approximate the functions is as follows:

```
import math

import numpy as np

import matplotlib.pyplot as plt


# define the function f(x) = e^x * cos(x)

def f(x):

    return np.exp(x) * np.cos(x)


# define the Taylor polynomial T(x) with default Maclaurin expansion

def T(x, x0=0):

    if x0 == 0:  # Maclaurin polynomial

        return 1 + x - (x**3) / 3

    else:  # Taylor polynomial centered at x0

        return -3.07493232064 - 9.79378201807 * (x - x0) \

               - 6.71884969745 * (x - x0)**2 - 1.2146391256 * (x - x0)**3


# define the remainder (error estimate) for the Taylor approximation

def R(x, x0=0):

    if x0 == 0:  # Maclaurin series remainder

        return (-4 * (x**4)) / math.factorial(4)

    else:  # Taylor series remainder centered at x0
```

```
        return (12.2997292826 * (x - x0)**4) / math.factorial(4)


# calculate the actual error between the function and the Taylor approximation
def calc_error(x, x0=0):
    return f(x) - T(x, x0)


# example calculations
x_values = [1, 2]


# nicely formatted output function
def display_output(x):
    print(f"\n{'='*40}")
    print(f"Results for x = {x}:")
    print(f"{'-'*40}")
    print(f"f(x):                  {f(x):.6f}")
    print(f"Maclaurin T(x):        {T(x):.6f}")
    print(f"Taylor T(x) at x0=2:    {T(x, 2):.6f}")
    print(f"Maclaurin remainder R(x): {R(x):.6f}")
    print(f"Taylor remainder R(x) at x0=2: {R(x, 2):.6f}")
    print(f"Maclaurin error:        {calc_error(x):.6f}")
    print(f"Taylor error at x0=2:   {calc_error(x, 2):.6f}")
    print(f"{'='*40}\n")
```

```
# printing
for x in x_values:
    display_output(x)
```

Output:

```
Results for x = 1:
----------------------------------------
f(x):                   1.468694
Maclaurin T(x):         1.666667
Taylor T(x) at x0=2:    1.214639
Maclaurin remainder R(x): -0.166667
Taylor remainder R(x) at x0=2: 0.512489
Maclaurin error:        -0.197973
Taylor error at x0=2:   0.254055
========================================

Results for x = 2:
----------------------------------------
f(x):                   -3.074932
Maclaurin T(x):         0.333333
Taylor T(x) at x0=2:    -3.074932
Maclaurin remainder R(x): -2.666667
```

Taylor remainder R(x) at x0=2: 0.000000

Maclaurin error: -3.408266

Taylor error at x0=2:   0.000000

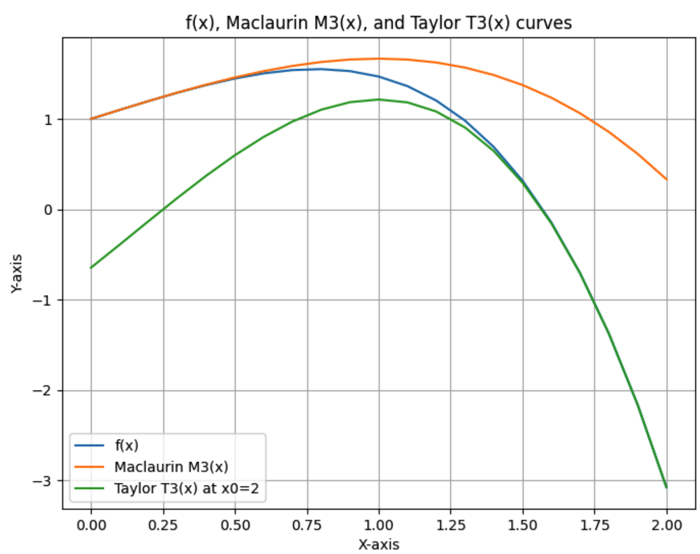
==

The output from the computer analysis has matched the report. Hence proving that the computer analysis is right.

The Three functions (f (x), M3(x), T3(x)):

```
# set up x-values from 0 to 2 with steps of 0.1
xs = [x / 10 for x in range(0, 21)]

# compute the values for f(x), Maclaurin T(x), and Taylor T(x) at x0=2
fs = [f(x) for x in xs]
m3s = [T(x) for x in xs]  # Maclaurin polynomial values
t3s = [T(x, 2) for x in xs]  # Taylor polynomial values at x0 = 2

# calculate the approximate and actual errors for Maclaurin and Taylor series
m3_approx_error = [R(x) for x in xs]
m3_calc_error = [calc_error(x) for x in xs]
t3_approx_error = [R(x, 2) for x in xs]
t3_calc_error = [calc_error(x, 2) for x in xs]

# plot f(x), Maclaurin M_3(x), and Taylor T_3(x) curves
curves = {"f(x)": fs, "Maclaurin M3(x)": m3s, "Taylor T3(x) at x0=2": t3s}

plt.figure(figsize=(8, 6))
for curve_name, curve_data in curves.items():
    plt.plot(xs, curve_data, label=curve_name)
```

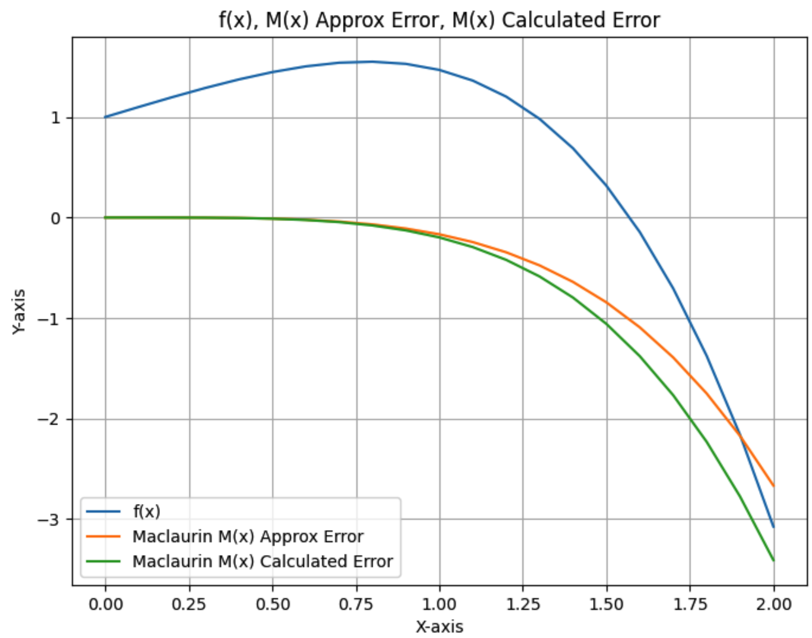
```
# label the axes and add title and grid

plt.xlabel("X-axis")

plt.ylabel("Y-axis")

plt.title("f(x), Maclaurin M3(x), and Taylor T3(x) curves")

plt.legend()

plt.grid(True)


# display the plot

plt.show()
```



This graph compares f (x), the Maclaurin polynomial $M_3(x)$ centered at 0, and the Taylor polynomial $T_3(x)$ centered at $x_3 = 0$. It highlights how well each polynomial approximates f (x) near their respective center points.

Approximated and Calculated error for $M_3(x)$:

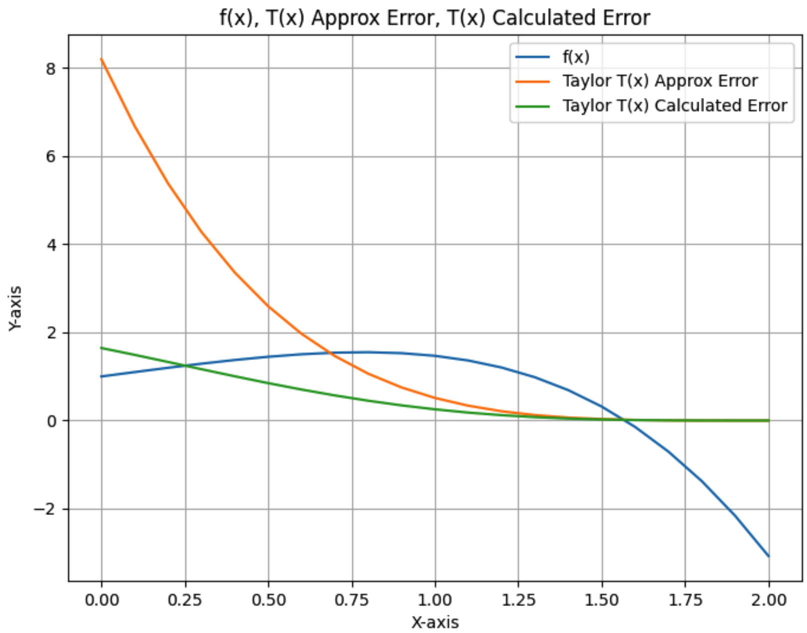
```
# define the data for f(x), Maclaurin approximation error, and calculated error
curves = {
    "f(x)": fs,
    "Maclaurin M(x) Approx Error": m3_approx_error,
    "Maclaurin M(x) Calculated Error": m3_calc_error
}

# create the plot
plt.figure(figsize=(8, 6))
for curve_name, curve_data in curves.items():
    plt.plot(xs, curve_data, label=curve_name)

# label the axes and set up title, grid, and legend
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("f(x), M(x) Approx Error, M(x) Calculated Error")
plt.legend()
plt.grid(True)

# show the plot
plt.show()
```



This graph shows f(x) alongside the theoretical Maclaurin approximation error and the actual calculated error. It visualizes the accuracy of the Maclaurin polynomial approximation, indicating the deviation between the true function and the polynomial around $x_0 = 0$

Approximated and Calculated error for $T_3(x)$:

```
# define the data for f(x), Taylor approximation error, and calculated error
curves = {
    "f(x)": fs,
    "Taylor T(x) Approx Error": t3_approx_error,
    "Taylor T(x) Calculated Error": t3_calc_error
}

# create the plot
plt.figure(figsize=(8, 6))
for curve_name, curve_data in curves.items():
    plt.plot(xs, curve_data, label=curve_name)

# label the axes and set up title, grid, and legend
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("f(x), T(x) Approx Error, T(x) Calculated Error")
plt.legend()
plt.grid(True)

# show the plot
plt.show()
```



This graph presents f (x) along with the theoretical and calculated errors for the 3rd-degree Taylor polynomial centered at $x_0 = 2$. It emphasizes the approximation accuracy of the Taylor polynomial in predicting f (x) near $x = 2$.

7. Conclusion

To approximate f (x) = e^x cos(x), the third-degree Maclaurin polynomial $M_3(x)$ and the third-degree Taylor polynomial $T_3(x)$ were compared in this analysis. When x = 0, the Maclaurin polynomial performed well; however, as x increased, its accuracy declined. Likewise, the Taylor polynomial yielded precise approximations in the vicinity of x = 2, but its efficacy decreased in the further distance from its center. Although both polynomials were helpful in the local area, they showed notable errors at higher values of x, highlighting the significance of selecting a suitable center for Taylor series approximations.

8. References

- Desmos: https://www.desmos.com/calculator/qu1ueoxxlr
- Teatime Numerical Analysis
- LibreTexts Mathematics.

  https://math.libretexts.org/Courses/Cosumnes_River_College/Math_401%3A_Calculus_II_-_Integral_Calculus/04%3A_Power_Series/4.03%3A_Taylor_and_Maclaurin_Series
- ChatGpt