COSC 6364: Advanced Numerical Analysis

Assignment #1

Name: Ashutosh Kumar (1854393) Due on: 28 February 2023

1 Introduction

For this assignment, we are required to implement various integration and differentiation algorithms, apply them, calculate their errors against the ground truth, and compare their accuracies. The function that we will be applying these algorithms to is

$$F(t) = Ae^{kt}\cos{(wt)}$$

where

$$\begin{aligned} A &= 1.000 \\ k &= 0.055 \\ w &= 2.0 \\ T_a &= 0.001 \\ T_b &= 2.0 \end{aligned}$$

2 Integration

2.1 Integration Algorithms & Calculation Methodology

For the integration part of this assignment, I will be implementing the following three methods in Python to calculate the integrals.

1. Midpoint Rule
2. Trapezoidal Rule
3. Simpson's Rule

All of these procedures calculate integrals by dividing the range between the lower and upper into a pre-defined number of divisions, n. I have calculated integrals by using various n values: 2^2, 2^4, 2^6, 2^8, and 2^{10}. Unsurprisingly, as the number of data points (or divisions) increase, the accuracy of the integral improves, and the error decreases.

2.2 Ground Truth for Integral

In order to calculate the errors, we first need to calculate some 'ground truth', which serves as the 'true' or the most accurate value. A ground truth must be independent of the algorithm being implemented to calculate the integral. Integrating by hand, using the rules of integrations, makes sure that the ground truth remains true as it will be independent of the three integration algorithms implemented here.

The error is calculated by taking the absolute difference between this ground truth and the integral values calculated by the three algorithms.

$$I = \int_{0.001}^{2.0} e^{0.055t}\cos{(2t)}\ dt$$

We need to integrate this function using Integration by Parts method.

Let $u = \cos(2t)$, then $du = -2\sin(2t)\ dt$. Let $dv = e^{0.055t}dt$, then $v = \frac{1}{0.055}e^{0.055t}$. Then,

$$I = \frac{1}{2.0}e^{0.055t}\cos(2t)\Big|_{0.001}^{0.055} + \int_{0.001}^{2.0}\frac{2}{0.055}e^{0.055t}\sin(2t)dt$$

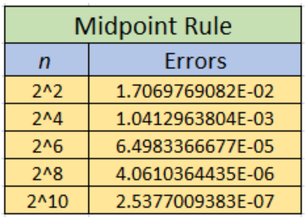
Now, let $x = \sin(2t)$, then $dx = 2\cos(2t)$. Let $dy = e^{0.055t}dt$, then $y = \frac{1}{0.055}e^{0.055t}$. Then,

$$\begin{aligned}
I &= \frac{1}{2.0}e^{0.055t}\cos(2t)\Big|_{0.001}^{0.055} + \frac{2}{0.055^2}e^{0.055t}\sin(2t)\Big|_{0.001}^{0.055} - \frac{4}{0.055^2}\int_{0.001}^{2.0} e^{0.055t}\cos(2t)\ dt \\
&= \frac{1}{2.0}e^{0.055t}\cos(2t)\Big|_{0.001}^{0.055} + \frac{2}{0.055^2}e^{0.055t}\sin(2t)\Big|_{0.001}^{0.055} - \frac{4}{0.055^2}I \\
\left(1 + \frac{4}{0.055^2}\right)I &= \frac{1}{2.0}e^{0.055t}\cos(2t)\Big|_{0.001}^{0.055} + \frac{2}{0.055^2}e^{0.055t}\sin(2t)\Big|_{0.001}^{0.055} \\
\boxed{I = -0.44684653}
\end{aligned}$$

Therefore, the ground truth for integration of our function is $\boxed{-0.44684653}$.

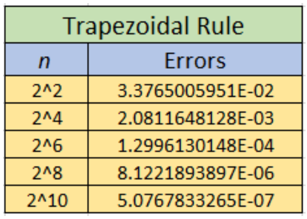
2.3 Results, Error Analysis, & Comments

The ground truth value of the integral of the function is -0.44684653. Amongst the three methods, Simpson's Rule method gives the most accurate result. The errors for the three procedures are given below 1.



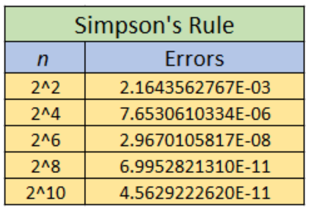
Midpoint Rule	
n	Errors
2^2	1.7069769082E-02
2^4	1.0412963804E-03
2^6	6.4983366677E-05
2^8	4.0610364435E-06
2^10	2.5377009383E-07

(a) Errors in Midpoint Rule



Trapezoidal Rule	
n	Errors
2^2	3.3765005951E-02
2^4	2.0811648128E-03
2^6	1.2996130148E-04
2^8	8.1221893897E-06
2^10	5.0767833265E-07

(b) Errors in Trapezoidal Rule



Simpson's Rule	
n	Errors
2^2	2.1643562767E-03
2^4	7.6530610334E-06
2^6	2.9670105817E-08
2^8	6.9952821310E-11
2^10	4.5629222620E-11

(c) Errors in Simpson's Rule

Figure 1: Errors in various integration procedures

The error vs. n graph is in the figure 2.

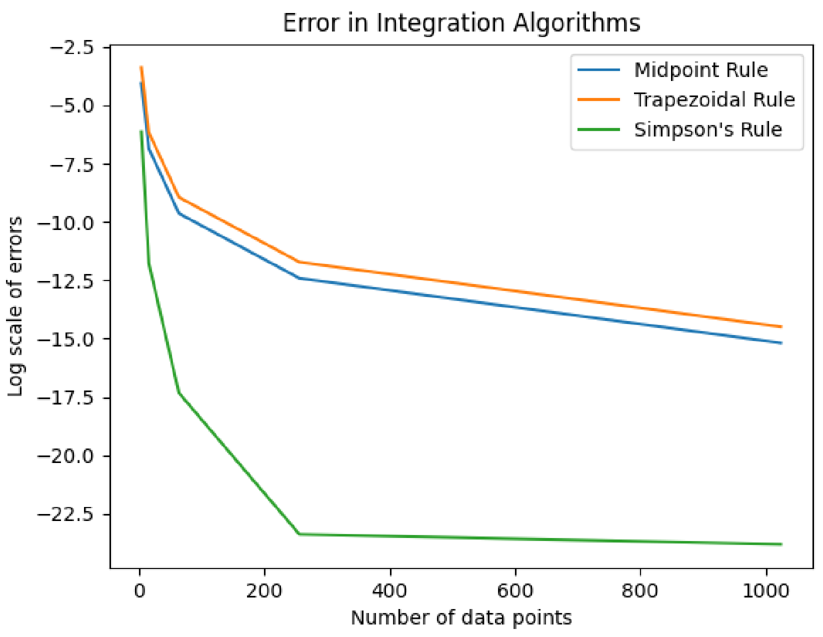


Figure 2: Error vs. n graph

As can be seen in the graph 2 above, Simpson's Rule is significantly better than the other two algorithms.

For all three algorithms, the errors go down as the number of partitions increases. However, the drop in the error for Simpson's Rule as the number of partitions go up is greater than the other two algorithms. Both Midpoint and Trapezoidal Rules follow a similar trajectory in the error (log) vs. partitions.

3 Differentiation

3.1 Differentiation Algorithms & Calculation Methodology

For the differentiation part of this assignment, I will be implementing the following two methods in Python.

1. Forward Difference
2. Central Difference

Both of these algorithms calculate the differentiation by taking in some input value and a small step, h. The smaller this step is, the better the results are, i.e., the more accurate the differentiation is. I have generated plots by both, keeping h constant, and by keeping h variable.

At each step, the differentiation value is compared with the ground truth for error calculation.

3.2 Ground Truth for Differentiation

In order to calculate the errors, we first need to calculate some 'ground truth', which serves as the 'true' or the most accurate value. A ground truth must be independent of the differentiation algorithm being implemented. Differentiating by hand using the rules of differentiation makes sure that the solution remains true, as it is independent of the differentiation algorithms I implemented.

The error is calculated by taking the absolute difference between this ground truth and the differentiation values calculated by the two algorithms.

I have differentiated the given function by hand. The result is stored in a Python function, which takes in an input value, passes it through this result, and outputs the 'ground truth' at that input point.

The differentiation is as follows:

$$
\begin{aligned}
F(t) &= e^{0.055t}\cos(2t)\\
\frac{\mathrm{d}F}{\mathrm{d}t} &= e^{0.055t}\left(\frac{\mathrm{d}\cos(2t)}{\mathrm{d}t}\right) + \cos(2t)\left(\frac{\mathrm{d}e^{0.055t}}{\mathrm{d}t}\right)\\
&= e^{0.055t}(-2\sin(2t)) + 0.055\ e^{0.055t}\cos(2t)
\end{aligned}
$$

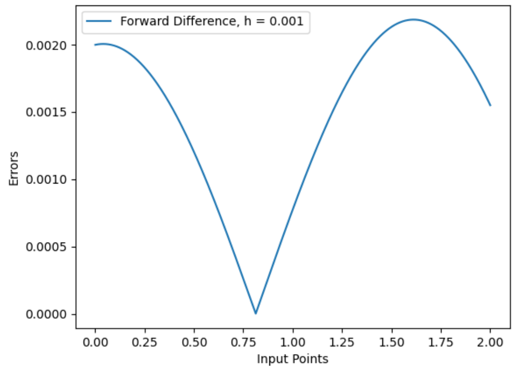
$$\boxed{\frac{\mathrm{d}F}{\mathrm{d}t} = 0.055\ e^{0.055t}\cos(2t) - 2e^{0.055t}\sin(2t)}$$

$$\boxed{\frac{\mathrm{d}^2F}{\mathrm{d}t^2} = 0.055^2\ e^{0.055t}\cos(2t) - 4e^{0.055t}\sin(2t) - (4\times 0.055)\sin(2t)}$$

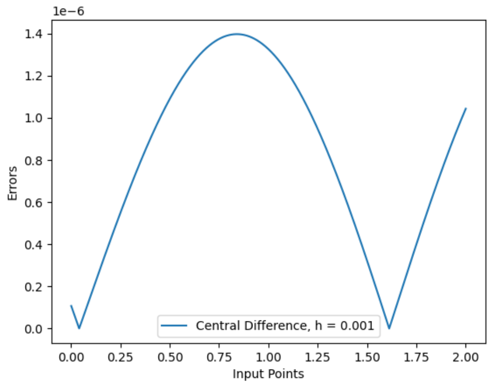
3.3 Results, Error Analysis, & Comments

3.3.1 First Order Derivative

There are three graphs below that have been generated using a constant step size ($h = 0.001$) – one shows the errors in the Forward Difference 3a method, another shows the errors in the Central Difference 3b method, and the last shows the errors of both methods in a single plot 4.



(a) Errors in Forward Difference



(b) Errors in Central Difference

Figure 3: Errors in various differentiation procedures

As can be seen in the two plots above 3a & 3b, both the differentiation algorithms have somewhat similar looking errors, however, the scales of both vary greatly. The errors seen by the Central Difference method are far, *far* smaller than those seen by the Forward Difference method.

This huge difference in the error is blatantly apparent in the plot below 4. Compared to Forward Difference, Central Difference has almost no errors.

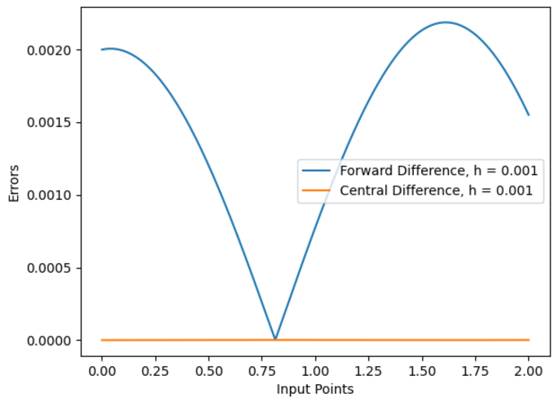
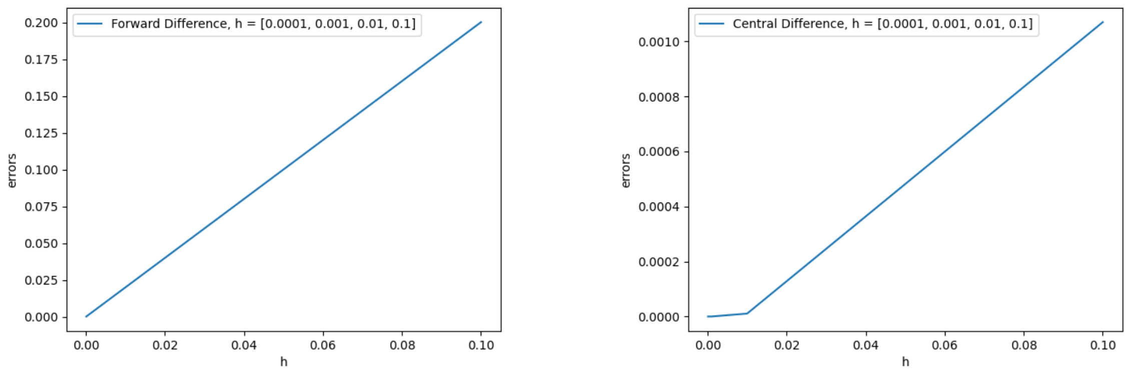


Figure 4: Errors in the two differentiation procedures

Additionally, I have also generated plots for various step sizes ($h = [0.0001, 0.001, 0.01, 0.1]$) with constant input ($= 0.001$).



(a) Errors in Forward Difference with variable h (b) Errors in Central Difference with variable h

Figure 5: Errors in various differentiation procedures with variable h

Looking at the above plots 5a & 5b, it is clear that as the value of the step size h increases, the error increases as well. Once again, we see in plot 6 that Central Difference is significantly more accurate than Forward Difference.

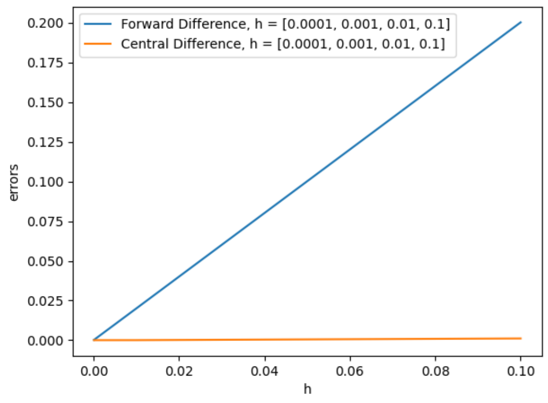
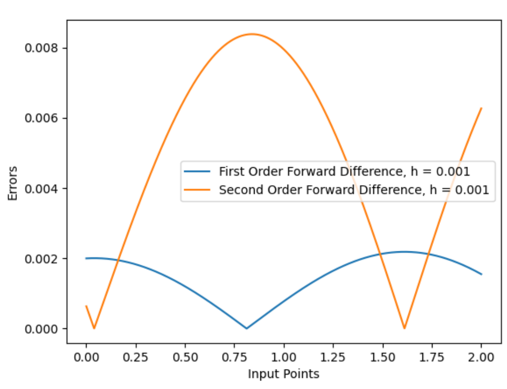


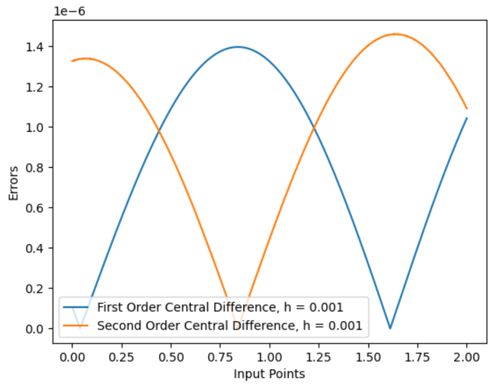
Figure 6: Errors in the two differentiation procedures

3.3.2 Second Order Derivative

Similar to the first order derivatives, we have plots for the second order derivatives.



(a) Errors in the Second Order Forward Difference with variable h for the second order



(b) Errors in the Second Order Central Difference with variable h

Figure 7: Errors in various differentiation procedures with variable h

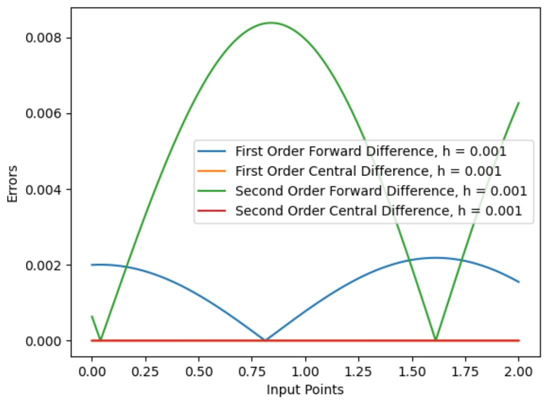


Figure 8: Errors in the two differentiation procedures for the second order

Here too, we see a similar trend. Central Difference gives us a second order derivative with smaller error. However, we can see that the second order errors are greater than the first order errors (at least for the Forward Difference method).