Comparative & Sensitivity Analysis of Numerical Methods for Fractional Order Derivative

By

Heba Saadat Rana

Reg. No. 826-FBAS/MSMA/F22

Department of Mathematics and Statistics

Faculty of Sciences

International Islamic University, Islamabad

Pakistan

2024

Comparative & Sensitivity Analysis of Numerical Methods for Fractional Order Derivative

By

Heba Saadat Rana

Registration No. 826-FBAS/MSMA/F22

Department of Mathematics and Statistics

Faculty of Sciences

International Islamic University, Islamabad

Pakistan

2024

Comparative & Sensitivity Analysis of Numerical Methods for Fractional Order Derivative

By

Heba Saadat Rana

Registration No. 826-FBAS/MSMA/F22

Supervised by

Dr. Tahira Jabeen

Department of Mathematics and Statistics

Faculty of Sciences

International Islamic University, Islamabad

Pakistan

2024

Comparative & Sensitivity Analysis of Numerical Methods for Fractional Order Derivatives

By

Heba Saadat Rana

Registration No. 826-FBAS/MSMA/F22

A Thesis
Submitted in the Partial Fulfillment of the
Requirement of the Degree of
MASTER OF SCIENCE
In
MATHEMATICS

Supervised by

Dr. Tahira Jabeen

Department of Mathematics and Statistics

Faculty of Sciences

International Islamic University, Islamabad

Pakistan

2024

Declaration

I, Heba Saadat Rana, hereby declare that my paper titled "Comparison and Evaluation of Fractional Differential Numerical Methods" is original and has been accepted. This work has not been submitted in whole or in part as part of any other course to any institution in Pakistan or abroad.

Heba Saadat Rana
MS (Mathematics)
Reg. No. 826-FBAS/MSMA/F22
Department of Mathematics and Statistics
International Islamic University, Islamabad

Thesis Certificate

The thesis entitled "Comparative & Sensitivity Analysis of Numerical Methods for Fractional Order Derivatives" submitted by Heba Saadat Rana, 826-FBAS/MSMA/F22 in partial fulfillment of MS Degree in Mathematics has been completed under my guidance and supervision. I am satisfied with the quality of her work and allow her to submit this thesis for further process to graduate with a Master of Science from the Department of Mathematics and Statistics, as per IIUI rules and regulations.

Dr. Tahira Jabeen
Assistant Professor
Department of Mathematics and Statistics
International Islamic University, Islamabad

Acknowledgement

I am grateful to Almighty ALLAH for providing me with strength, knowledge, and tenacity during my academic path. After that, I want to thank Dr. Tahira Jabeen for her wonderful supervision.

It's been a joy having her as my supervisor. She is always eager to share her extensive expertise when needed. Her insightful knowledge, outstanding mentorship, and continuous support helped me to finish my project. I am grateful to Dr. Farzana Akhtar Abbasi, Chairperson of the Department of Mathematics, for her guidance, helpful attitude, and conducive environment for studying and research.

I am grateful to my parents and siblings for their constant care,patience, and love. Finally, I am grateful to my family members for their support and prayers, without them this work would never have come into existence.

Heba Saadat Rana

Dedicated

to my beloved

Parents

Mr. & Mrs. S. Rana

and

Grandparents

Mr. & Mrs. M. Amjad (Late)

Preface

This project focuses on creating a Python/MATLAB code to estimate the percentage of a given value and then compare the result to the total value. Fractional derivatives are a powerful mathematical tool for modeling operations with infinite numbers. These rules will ensure the performance of estimating these values and measuring their accuracy. This article is divided into five parts.

In the first chapter, the historical background of fractional calculus and some basic definitions are stated along with the literature review.

In Chapter 2, smog data is collected from AQI for different cities of Pakistan then we derive the factors causing smog. We then drive mathematical modeling of smog data and formation of Modified Trapezoidal Rule.

In Chapter 3, derivation of Modified Trapezoidal Rules and Caputo Fractional Derivative Rules along with theorem and applications.

Nomenclature

• Mittag-Leffler	M. L.
• Riemann-Liouville	R. L.
• General Analytic Kernel	G. A. K.
• Modified Trapezoidal Rule	M. T. R.
• Fractional Integral	F. I.
• Fractional Differential	F. D.
• Fractional Differential Equation	F. D. E.
• Caputo Fractional Derivative	C. F. D.
• Air Quality Index	A. Q. I.

Contents

Abstract

This thesis delves into the advancements and applications of fractional calculus, focusing on the comparative and sensitivity analysis of numerical methods specifically the modified trapezoidal rule and Caputo derivatives applied to fractional order derivatives. Using trigonometric functions, $\sin(x)$ and $\cos(x)$, and smog data from Pakistan, the study explores the performance of these methods across orders ranging from 0 to 1.

Key findings reveal that the error behavior of the modified trapezoidal rule is highly dependent on the fractional order parameter α, with minimum error observed around α=0.3 maximum error near α=0.9. Additionally, the performance varies significantly between functions, with $\sin(x)$ exhibiting errors three orders of magnitude smaller than $\cos(x)$. The Caputo derivatives, particularly the 0.5th order, demonstrated high sensitivity and accuracy in approximating the sine function, highlighting the importance of careful parameter selection for numerical accuracy.

The research underscores the practical implications of fractional calculus in modeling complex systems in physics, engineering, and biology, where traditional calculus falls short. It also emphasizes the relevance of these methods in contemporary clinical research, offering new avenues for analyzing and interpreting medical data. Despite its contributions, the study acknowledges limitations in scope, particularly in its focus on air quality data, the constraints of the modified trapezoidal rule, and the need for more comprehensive error analysis and parameter optimization.

Chapter 1

Introduction

Fractional Calculus is a unique department of carried-out sciences offering derivatives of arbitrary (actual/complicated) order. Fractional calculus has emerged as an effective mathematical device with critical implications for information-complicated structures that defy conventional integer-order calculus. These complex structures, which can be discovered in physics, engineering, and biology, among different fields, have sparked a significant surge of scientific interest.

The maximum hard issue of using fractional calculus is calculating fractional derivatives, which generally defy analytical solutions. This problem has precipitated the improvement of numerous numerical algorithms to reliably and swiftly compute fractional order derivatives. The number one motive of this studies have a take a observe is to offer a complete comparative assessment of the numerous numerical algorithms used to compute fractional order derivatives.

The number one intention of this evaluation is to decide how nicely those numerical strategies carry out in assessment of each other and to analytical solutions. This study makes a specialty of the essential definitions of fractional derivatives: Riemann-Liouville, Modified Trapezoidal Rule and Caputo fractional by-product. The Riemann-Liouville by-product, the maximum broadly used definition of fractional derivatives, extends to the conventional integer-order by-product framework.

In contrast, the Grunwald-Letnikov by-product introduces a discrete approach to fractional derivatives using a weighted sum of characteristic values. By combining beginning situations in its definition, the Caputo by-product additionally called the fractional by-product in regards to preliminary circumstances, improves at the classical by-product. Through this assessment research, the researchers need to shed mild on the blessings and barriers of every numerical technique for computing fractional order derivatives.

Such data is beneficial for lecturers and experts searching to accurately observe the fractional calculus of their fields. This study's paper's trajectory starts with an in-depth evaluation of fractional order derivatives, their mathematical underpinnings, and their essential utilization in present-day clinical studies.

This essential information is needed to research destiny trigonometric features, sine (sin(x)), and cosine (cos(x)) characteristics. These approaches function as the inspiration for mathematical modeling and represent an extensive variety of actual-global structures. The number one intention of this study is to compute the fractional derivatives of trigonometric features, particularly sin(x) and cos(x), spanning various fractional orders (0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and primary order). These computations function as essential research of the sensitivity and accuracy of numerous fractional derivatives, imparting treasured data concerning their realistic packages.

Instead of being merely theoretical, this examination appears at how fractional calculus may be used to cope with complicated, dynamic structures inside the actual global. Fractional order derivatives are interesting due to their awesome capacity to offer a greater entire and unique description of methods that don't observe conventional integer-order differentiation. Fractional calculus, as adversarial to standard calculus, permits us to understand the complexities of structures with long-variety memory, fractal properties, and non-nearby behaviors.

As a consequence, fractional calculus has discovered packages in an extensive variety of fields, together with physics, engineering, biology, economics, and others, demonstrating its transformational power. Observing fractional derivatives carried out to trigonometric features is an essential element of this study. The goal is to research the sensitivity and precision of those derivatives, exposing contrasts and subtleties that seem at distinct fractional orders, through computing fractional derivatives of the sine and cosine at distinct fractional orders. In addition to enhancing our information on fractional calculus, this analytical approach guarantees to open up new avenues for progressive packages in disciplines in which unique modeling and evaluation are essential.

In conclusion, at the same time as the theoretical exam is the inspiration of this study, it isn't confined to abstraction. The task intends to bridge the theoretical and realistic hole through the usage of fractional order derivatives to actual global records and complicated structures. This study's paper provides for the continued improvement of fractional calculus and its actual effect on several clinical and realistic fields by placing the framework for destiny studies so that it will enlarge and widen our hold close of this effective mathematical device. Future studies would possibly have a take observe greater complicated packages, extra numerical strategies, or increasing using fractional calculus into new domains, all of which might assist in enhancing the subject.

1.1 Historical Background

Fractional calculus dates returned to the early 18th century, while pioneering mathematicians like Leonhard Euler, Johann Bernoulli, and Pierre-Simon Laplace commenced to research the extension of differentiation and integration to non-integer orders. Despite those early advances, fractional calculus remained noticeably inert till the overdue 1960s, while a revived hobby inside the subject arose, fueled via way of means of contributions from mathematicians which include Anatoly N. Kolmogorov, Alexander M. Ostrowski, and Anatoly M. Samoilenko. Their contributions reinvigorated the a look of fractional calculus and set up the framework for its destiny improvement and use in quite a few areas.

Fractional calculus has acquired a variety of hobbies in current many years due to its huge variety of programs. It has proved beneficial for simulating complex structures like organic structures, manipulate structures, and viscoelastic materials [13]. Furthermore, fractional calculus has programs in statistics analysis, photograph processing, and sign processing, highlighting its significance in contemporary-day clinical and technical activities.

However, the computational hassle of producing fractional derivatives, which regularly defy analytical solutions, has been a main obstacle to having a look at fractional calculus. To remedy this difficulty, researchers have created a huge variety of numerical algorithms, that are kind of categorized as spectral or non-spectral techniques.

Spectral procedures use collection expansions of the characteristic being differentiated to approximate fractional derivatives. Examples are the Fourier collection technique, the Chebyshev polynomial method, and the Legendre polynomial method.

Non-spectral methods, on the other hand, use finite separation or expansion techniques to approximate fractional derivatives. Riemann-Liouville [28], Grunwald-Letnikov and Caputo derivatives are the most frequently used definitions. The Riemann-Liouville byproduct uses a covariance operator to express the inverse of the variable, while the Grunwald-Letnikov byproduct uses a decomposition function based on weighted eigenvalues. Caputo Spinoff expands the Classic Spinoff by combining the starting area.

Extensive have a look at has focused on figuring out the usefulness and performance of diverse numerical strategies, thinking of parameters which includes convergence rates, stability, and computing complexity. These experiments have supplied insights into the strengths and bounds of various methodologies, spurring the improvement of greater unique and green techniques for computing fractional derivatives.

In summary, fractional calculus has visible a brilliant rebirth, growing from its humble beginnings to end up a cornerstone of cutting-edge mathematical and clinical studies. While computing boundaries exist, contemporary-day studies refines numerical methodologies and deepens our knowledge of fractional calculus, reinforcing its importance in coping with complex real-global problems.

1.2 Literature Review

Pakistan's Air Pollution Problem: Relationship Between Scientific Analysis and Fractional Calculation

The impact of poor air quality is far-reaching and has significant impacts on human health and economic productivity. The annual number of airborne deaths in Pakistan is expected to reach 105,000 by 2020, indicating an urgent need for mitigation strategies (Pollution: 2016). This highlights the important role of advanced analytical techniques such as numerical analysis and numerical calculations in understanding and unraveling the complexity of air pollution. Numerical analysis provides an important tool for determining and predicting the impact of environmental change on human health and the economy. Like "Applied Numerical Analysis" (C. Gerld, P. Wheatley, 2004) emphasized the role of numerical methods in assessing the economic losses of pollution. In Pakistan, where climate-related health and environmental costs account for approximately 1% of GDP, this process is critical for policy development and budget allocation (Park, 2013). Thanks to it, researchers can develop models to simulate the spread of infectious diseases, measure exposure risk, and predict long-term prognosis. These models include environmental data, public data, and economic parameters to provide policymakers with evidence-based insights on the cost-effectiveness of managing this bad weather. br>< br>Fractional calculus unifies traditional calculus by combining derivatives and concepts of non-numerical degrees and allows for a deeper understanding of Air pollution related systems provide a theoretical basis for using fractional derivatives to simulate complex environmental phenomena. 5 concentrations stand out. Vehicle emissions and commercial activities are the main source of this high pollution (Owusu and Sarkodie, 2020). Fractional analysis plays an important role in the development of forecast models that can account for poor air pollution distribution, seasonal variations and their effects on air quality. Application: Monitoring of Watersheds

Mathematical and numerical methods are frequently used to develop predictive models and monitoring systems to track pollutants and their consequences in Pakistan's urban landscape. Studies using remote sensing and ground monitoring have shown significant changes in air quality index (AQI) between weekdays and weekends in terms of interaction effects of environment and model (Lodhi and Ghauri, 2009; Raja and Biswas and Husain, 2010). These communication efforts are essential for developing intervention plans to reduce pollution hotspots, improve public health outcomes, and promote urban development. br>Pakistan's rapid development and economic growth have brought great challenges to climate control. Integrating these systems with environmental systems is essential for pollution control and mitigation strategies. Policymakers can use information from statistics and statistical analysis to create evidence-based policies that balance economic development with the environment. The combination of statistical analysis, general statistics and environmental science provides a useful framework for solving Pakistan's air quality problems. Through ongoing research and practical application of mathematical models, Pakistan can support a healthy environment and strong communities in its urban transformation. Integration of advanced analytical techniques improves our understanding of environmental processes and allows policymakers to make informed decisions to cleanly protect public health and promote sustainable development. The challenge of defining numbers and counting numbers is often a silver lining, offering new solutions to reduce pollution and ensure a safe future for future generations.

1.3 Trapezoidal Rule

Mathematical integration is an important tool used by scientists and researchers to obtain predictive answers to facts that cannot be resolved analytically. There are several ways to compare the major points of a given element by balancing the weights of the eigenvalues of the detailed points. The basis of the trapezoidal law is to divide the area between f(x) and the x-axis into a strip of straight lines in the function f(x).

Trapezoidal rule: Suppose that the interval [y,z] is subdivided into N sub-intervals $[x_i, x_{i+1}]$ of equal width $h = (z - y)/N$ by using the nodes $x_i = y + ih$ for $i = 0, 1, 2, .., N$.

The composite trapezoidal rule [26] for the function $f(x)$ over $[y, z]$ is defined as [1,4]

$$
\begin{aligned}
T(f,h) = \frac{h}{2}\sum_{k=1}^{N}(\ (x_{k-1}) + f\ (x_k)) \\
&= \frac{h}{2}\left(f_0 + f_1 + f_1 + f_2 + f_2 + \ldots + f_{N-1} + f_N\right) \\
&= \frac{h}{2}\left(f_0 + f_N\right) + \frac{h}{2}\left(2f_1 + 2f_2 + \ldots. + 2f_{N-1}\right) \\
&= \frac{h}{2}\left(f_0 + f_N\right) + \frac{h}{2}(2)\left(f_1 + f_2 + \ldots. + f_{N-1}\right)
\end{aligned}
$$

$$= \frac{h}{2}\left(f_0 + f_M\right) + h\left(f_1 + f_2 + \ldots . + f_{N-1}\right)$$

$$= \frac{h}{2}\left(f_0 + \ f_N\right) + h\sum_{k=1}^{N-1}\left(x_k\right)$$

1.4 Euler's Gamma Function

In fractional calculus theory, the Gamma function [14] is fundamental. It generalizes the factorial ($m!$) and accepts non-integer and complex values for m. The formal definition and a few key characteristics are presented below.

Definition 1.4.1 (Gamma function)**.** The Gamma function $\Gamma(\alpha)$, where α is a complex parameter, is defined by the integral:

$$\Gamma(\alpha) = \int_0^\infty e^{-t} t^{\alpha-1}\, dt,$$

which converges for $\Re(\alpha) > 0$.

Proposition 1.4.1. *The Gamma function can also be expressed as:* $\Gamma(\alpha + 1) = \alpha\Gamma(\alpha)$.

Proof. Consider the integral representation of the Gamma function:

$$\Gamma(\alpha + 1) = \int_0^\infty e^{-t} t^{\alpha}\, dt.$$

We will apply integration by parts to evaluate this integral. Let $u = t^\alpha$ and $dv = e^{-t}\, dt$. Then, we have $du = \alpha t^{\alpha-1}\, dt$ and $v = -e^{-t}$. Using the integration by parts formula, $\int u\, dv = uv - \int v\, du$, we get:

$$\Gamma(\alpha + 1) = \int_0^\infty t^\alpha e^{-t}\, dt = \left[-t^\alpha e^{-t}\right]_0^\infty + \int_0^\infty \alpha t^{\alpha-1} e^{-t}\, dt.$$

Evaluating the boundary term $[-t^\alpha e^{-t}]_0^\infty$, we note that as $t \to \infty$, e^{-t}

approaches 0 faster than any polynomial function grows, so:

$$\lim_{t\to\infty} -t^{\alpha}e^{-t} = 0.$$

When $t = 0$, since $\alpha > -1$, $t^{\alpha} \to 0$. Therefore, the boundary term is zero. Hence, we have:

$$\left[-t^{\alpha}e^{-t}\right]_0^{\infty} = 0.$$

This simplifies our expression to:

$$\Gamma(\alpha+1) = \alpha \int_0^{\infty} t^{\alpha-1}e^{-t}\, dt.$$

Recognizing the remaining integral as $\Gamma(\alpha)$, we find:

$$\Gamma(\alpha+1) = \alpha\Gamma(\alpha).$$

Thus, we have shown that:

$$\Gamma(\alpha+1) = \alpha\Gamma(\alpha).$$

□

Chapter 2

Preliminaries

Lahore, with a population of eleven million, is one of the world's most polluted cities. Contamination leads to fatalities, birth abnormalities, and a loss of years of life. The study's instant review of data from the Air Quality Index found that pollution in the air was 54% unsafe for everybody 88% over the duration from June 2019 to September 2021. In 2021, Lahore's pollution index smashed 175 µg/m3. To evaluate the dangerous air circumstances, 14 sites were chosen according to their level of industrialization and emissions from tailpipes. The information from remote sensing was analyzed in field investigations to determine the connection between pollution concentrations for these specific locations and the existing on ground procedures. CO, NO_2, SO_2, optical depth of aerosols, CH_4, or CH_3OH were both primary and secondary airborne contaminants selected during analysis. Calculating the current AQI with the month level for every pollutant revealed an overwhelming beneficial relationship with the AQI and SO_2, NO_2, and CO. Assessments showed that the weekend's AQI score was greater compared to the weekday. The findings may be used for developing carefully planned adaptation strategies that may be implemented immediately to lessen the town's visible concerns regarding the environment.

2.1 Problem Formation

2.1.1 Studing Area and Field Surveys of Sites

The district of Lahore appeared in the study area. The region, which spans the Punjab province of Pakistan, is approximately 1772 km2 in total and is located within 31○1500 00 N and 31○4500 00 N, and 74○0100 00 E and 74○3900 00 E.

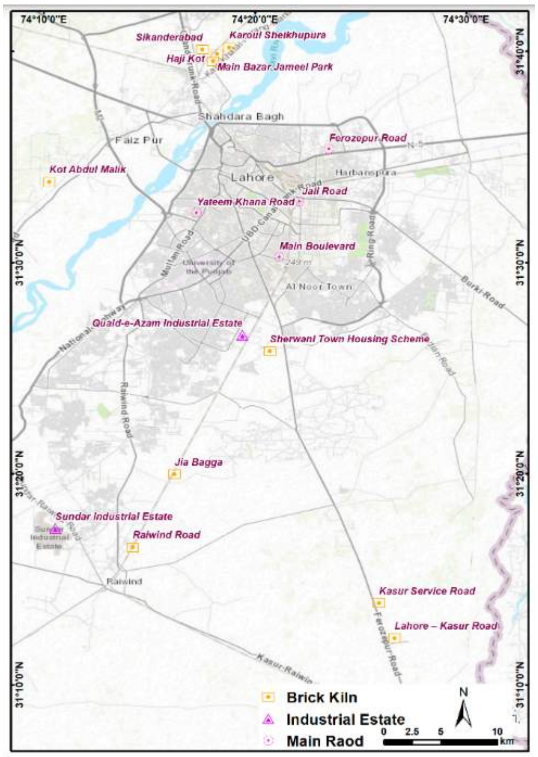


Figure 2.1: Map indicating Lahore district and its surrounding areas

With 1.12 million people living there, Lahore is the 26th most populous city on Earth [21]. Taking into account an average population density of 6500 people per 2 km, Lahore's population increased from 6.31 million [22] to 11.13 million [22] in 2017, almost doubling since 1998. Islamabad experiences dry, cold winters from November to February, and hot, humid summers from May to September. The average annual temperature is 23.3 °Celsius, and there is typically 2398 millimetres of precipitation, with 80%

of the total falling between May and September. Certain industries, brick kilns, and tailpipe emissions are among the main causes of air pollution in the town. As Figure 2.1 illustrates, these locations are dispersed, with nearly all of Punjab surrounding them. A ten-day inspection has been conducted at different work sites and special bricks since August 2020 to examine the current works, including the fuel type utilised and whether pollution by the International Organisation for Standardisation (ISO) can be prevented. technology that is employed. Searches are conducted on already identified routes. The data collected is summarised in Table.

2.1.2 Remote Sensing Data

In order to explore the related ecological effects and offer a crucial adaptation strategy over a particular area, monthly satellite observational data for nitrogen dioxide (NO2), carbon monoxide (CO), sulphur dioxide (SO2), formalin (HCHO), methane (CH4), and optical depth of aerosols (AOD) were gathered between July 2018 and April 2021. [16]

The search engine giant's Earth Engine (GEE) API on the cloud platform was used to gather and manage these data Four main pollution (NO2, CO, SO2) or three secondary pollutants (HCHO, CH4) were selected to determine the levels of air pollutants related to the three primary industries (industrial, congestion, and brick kiln) Sentry 5P's tropospheric surveillance equipment (TROPOMI) gave information on air pollution levels A spatial scale of 0.01 was employed to produce the Level 3 data Excellent craftsmanship pixels ('qa-value' $\geq$ 0.75) [15] were included in the result, and poor-quality pixels were removed. To avoid producing incorrect outcomes, regions (Khanum and Kumar, 2017) with cloud coverage probabilities of over 0.2 were removed as well from all datasets.

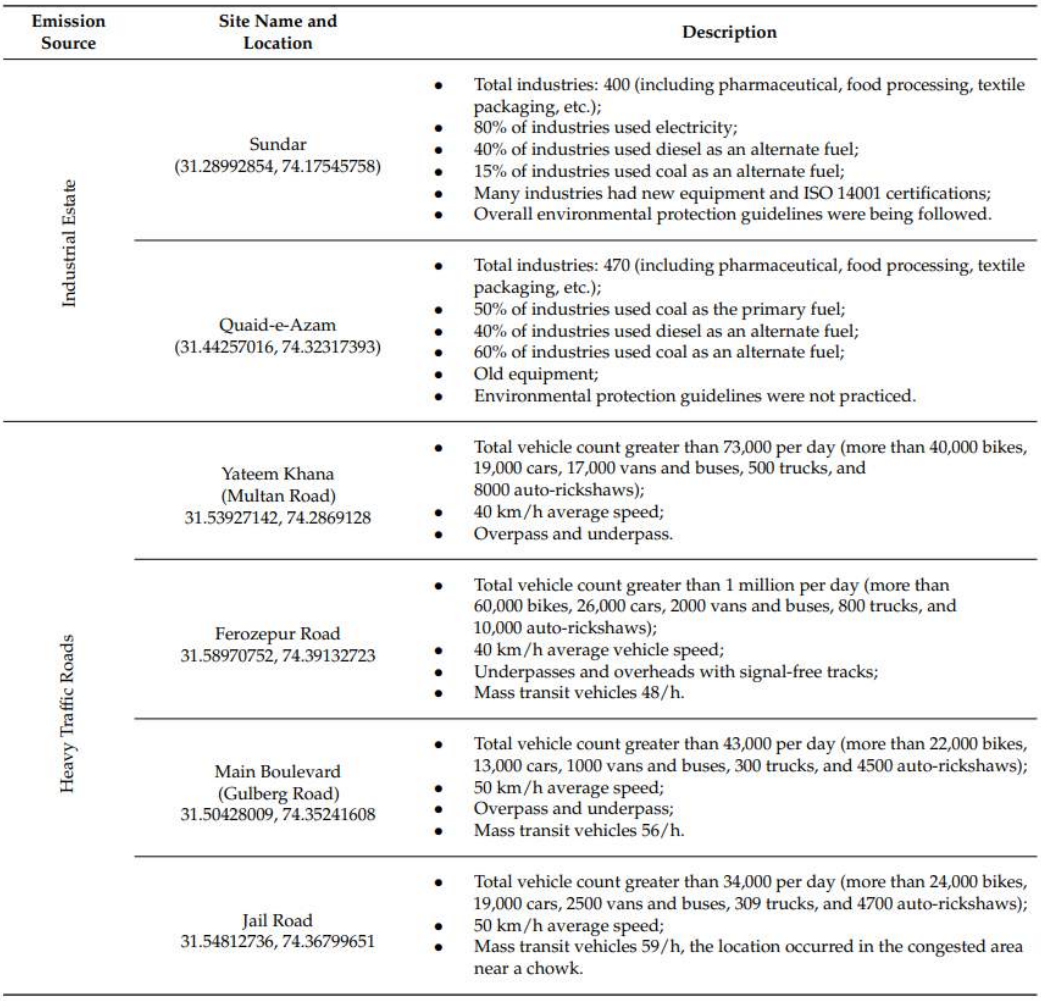
Month averaging and expanded intervals may help avoid significant errors from atmospheric conditions, even though meteorological factors may affect the air amount of the pollutants. In comparison sites in neighboring locations, nevertheless, the effect of climate was expected to be minimal. With near-daily worldwide coverage and 0.01 [17] spatial resolution, AOD provided an extensive variety of geographical data The medium-range image spectroradiometer (MODIS) [23] produced a level 2 product, MCD19A2-V6, with an AOD at 550 nm. The multi-angle application of atmospheric adjustment (MAIAC) [24] was used to combine the Terre and aquatic data for the final result [18]. Good quality pixels [19] were retained with the image quality tags layers "AOD-QA," and photons with tags such as "Cloud Shadows," "Overcast," or "Possibly Cloudy" had been eliminated using the "Cloud" mask. The mean of the pixels collected throughout the locations selected was employed to generate each dataset.

2.1.3 Quality Evaluation

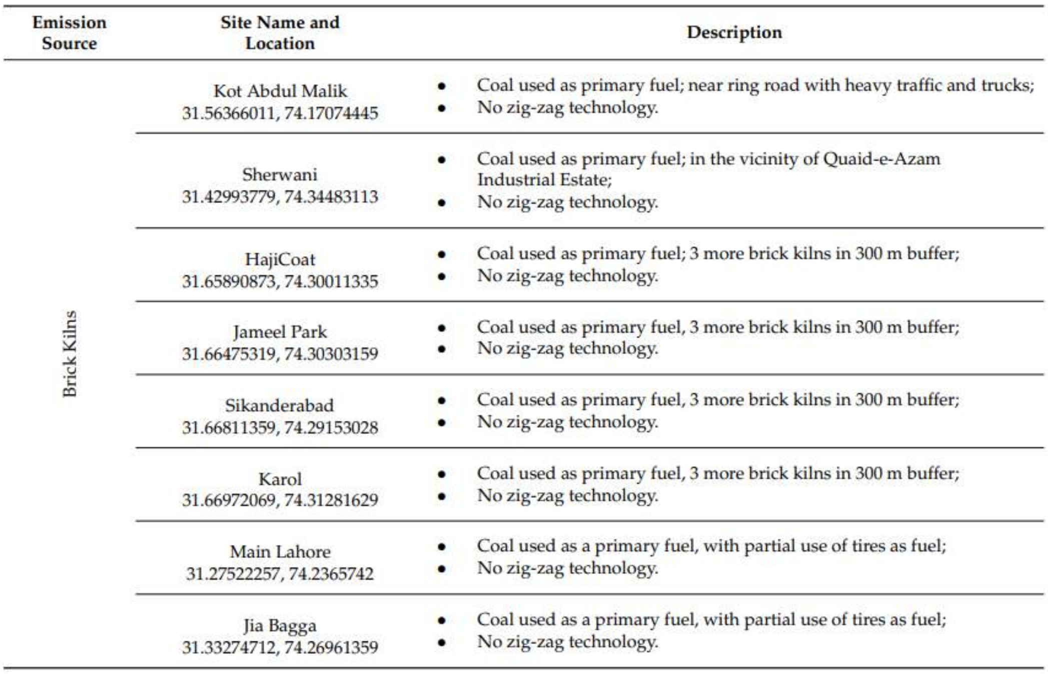
The association between AQI data (June 2019–September 2021) and World Health Organization guidelines was explored. The Pakistan Air Quality Monitor—USEPAWorld Air Quality Index Program (31.559989968731156, 74.33600917621109) [15] provided the necessary AQI information. Air quality monitoring equipment from Graphic Astronomy and Imaging Analysis (GAIA) were employed. The results were compared to WHO standards to assess if the air quality was adequate for human occupancy and produced significant results.

2.1.4 Relevance with Pollutants Spatial Data

To determine the relationship between the effects of variations in the overall amount of each pollutant and the corresponding AQI value in Lahore, the total amount of monthly peak levels for every location for each pollutant was plotted against the highest point in the month’s AQI readings. It was done to help ensure that data from remote sensing was relevant to the AQI findings and to investigate whether we may use regional concentrations of contaminants in conjunction with AOD findings for a credible indicator of AQI. It can also help determine which of the main pollutants Report Phrases that cause smog should be the focus of preventative measures.



Emission Source	Site Name and Location	Description
Industrial Estate	Sundar (31.28992854, 74.17545758)	• Total industries: 400 (including pharmaceutical, food processing, textile packaging, etc.); • 80% of industries used electricity; • 40% of industries used diesel as an alternate fuel; • 15% of industries used coal as an alternate fuel; • Many industries had new equipment and ISO 14001 certifications; • Overall environmental protection guidelines were being followed.
	Quaid-e-Azam (31.44257016, 74.32317393)	• Total industries: 470 (including pharmaceutical, food processing, textile packaging, etc.); • 50% of industries used coal as the primary fuel; • 40% of industries used diesel as an alternate fuel; • 60% of industries used coal as an alternate fuel; • Old equipment; • Environmental protection guidelines were not practiced.
Heavy Traffic Roads	Yateem Khana (Multan Road) 31.53927142, 74.2869128	• Total vehicle count greater than 73,000 per day (more than 40,000 bikes, 19,000 cars, 17,000 vans and buses, 500 trucks, and 8000 auto-rickshaws); • 40 km/h average speed; • Overpass and underpass.
	Ferozepur Road 31.58970752, 74.39132723	• Total vehicle count greater than 1 million per day (more than 60,000 bikes, 26,000 cars, 2000 vans and buses, 800 trucks, and 10,000 auto-rickshaws); • 40 km/h average vehicle speed; • Underpasses and overheads with signal-free tracks; • Mass transit vehicles 48/h.
	Main Boulevard (Gulberg Road) 31.50428009, 74.35241608	• Total vehicle count greater than 43,000 per day (more than 22,000 bikes, 13,000 cars, 1000 vans and buses, 300 trucks, and 4500 auto-rickshaws); • 50 km/h average speed; • Overpass and underpass; • Mass transit vehicles 56/h.
	Jail Road 31.54812736, 74.36799651	• Total vehicle count greater than 34,000 per day (more than 24,000 bikes, 19,000 cars, 2500 vans and buses, 309 trucks, and 4700 auto-rickshaws); • 50 km/h average speed; • Mass transit vehicles 59/h, the location occurred in the congested area near a chowk.



Emission Source	Site Name and Location	Description
Brick Kilns	Kot Abdul Malik 31.56366011, 74.17074445	• Coal used as primary fuel; near ring road with heavy traffic and trucks; • No zig-zag technology.
	Sherwani 31.42993779, 74.34483113	• Coal used as primary fuel; in the vicinity of Quaid-e-Azam Industrial Estate; • No zig-zag technology.
	HajiCoat 31.65890873, 74.30011335	• Coal used as primary fuel; 3 more brick kilns in 300 m buffer; • No zig-zag technology.
	Jameel Park 31.66475319, 74.30303159	• Coal used as primary fuel, 3 more brick kilns in 300 m buffer; • No zig-zag technology.
	Sikanderabad 31.66811359, 74.29153028	• Coal used as primary fuel, 3 more brick kilns in 300 m buffer; • No zig-zag technology.
	Karol 31.66972069, 74.31281629	• Coal used as primary fuel, 3 more brick kilns in 300 m buffer; • No zig-zag technology.
	Main Lahore 31.27522257, 74.2365742	• Coal used as a primary fuel, with partial use of tires as fuel; • No zig-zag technology.
	Jia Bagga 31.33274712, 74.26961359	• Coal used as a primary fuel, with partial use of tires as fuel; • No zig-zag technology.

The COVID-19 pandemic came out in February 2020, and the state of emergency that followed was partly imposed on industry or transportation operations throughout an array of phases. Pakistan had a nationwide state of emergency from March 25, 2020, to April 15, 2020, following which the clampdown was gradually lifted by May 15, 2020 [19]. Smart restrictions were initially implemented in hot spot areas in early May the study's reduction in emission depends on previous studies. Regarding the broader composition of the pollutants, a hypothesis has been made.

2.1.5 Pollutants' Concentration

To determine each station's contribution to the total Lahore sky globe, the cumulative quantity of each pollutant for the whole data period at each site was estimated. To find the principal responsible emissions source, the amount of emissions was lowered to a percentage. This is designed to be used as a tool to assist determine which pollutants, if future rules and legislation are in the works, should be the primary emphasis at a certain location.

The findings may also be converted into a useful tool and digital application for routine site inspection, allowing intervention against the fundamental drivers of pollution.

2.2 Discussion

The geographical data from locations gave the highest values for every single month through three years (June 2018–April 2021). The main points, which appear in the figure below, serve as crucial for further investigation and will form the basis for several inferences. The information shown indicates the mean level of every pollutant for the specified amount of time. The area that contributes the greatest amount to each pollutant in Lahore's air is also highlighted by this information. For all sites in the information set, the mean CO material was 41.1 millimol/m2. Jia Bagga has the smallest value (34.57 million/m2) while Jail Road has the greatest value (58.982 million/m2). The mean amount of NO2 accumulation was 0.1010 million/m2, with Kot Abdul reporting the lowest value of 0.037 million/m2 and Jail Road reporting the highest value of 0.255 million/m2 [23]. In Lahore, the average SO2 concentration was 0.428 million/m2. At Jia Bagga, the greatest value recorded was 0.747 million/m2, whereas in Karol, the smallest value was 0.232 million/m2. The mean CH4 value was 1.914 million/m2, with Industrial Estate having highest value at 1.993 million/m2 and Sundar Industries Estate having the lowest value at 1.860 million/m2. Its HCHO was 0.236 million/m2 on average. The main street yielded the greatest result, measuring 0.376 million/m2, while Jail Road produced the smallest value, at 0.079 million/m2 [23]. The safe range of AOD is 0.1 µm–0.4 µm, in the US EPA oversight nothing, not even one time, did the general quality of the air in Lahore match this minimum standard. The upper bound of the safe range is 5.6 times less than the highest level. This requires a thorough analysis of the data to create workable adaption action plans right away.

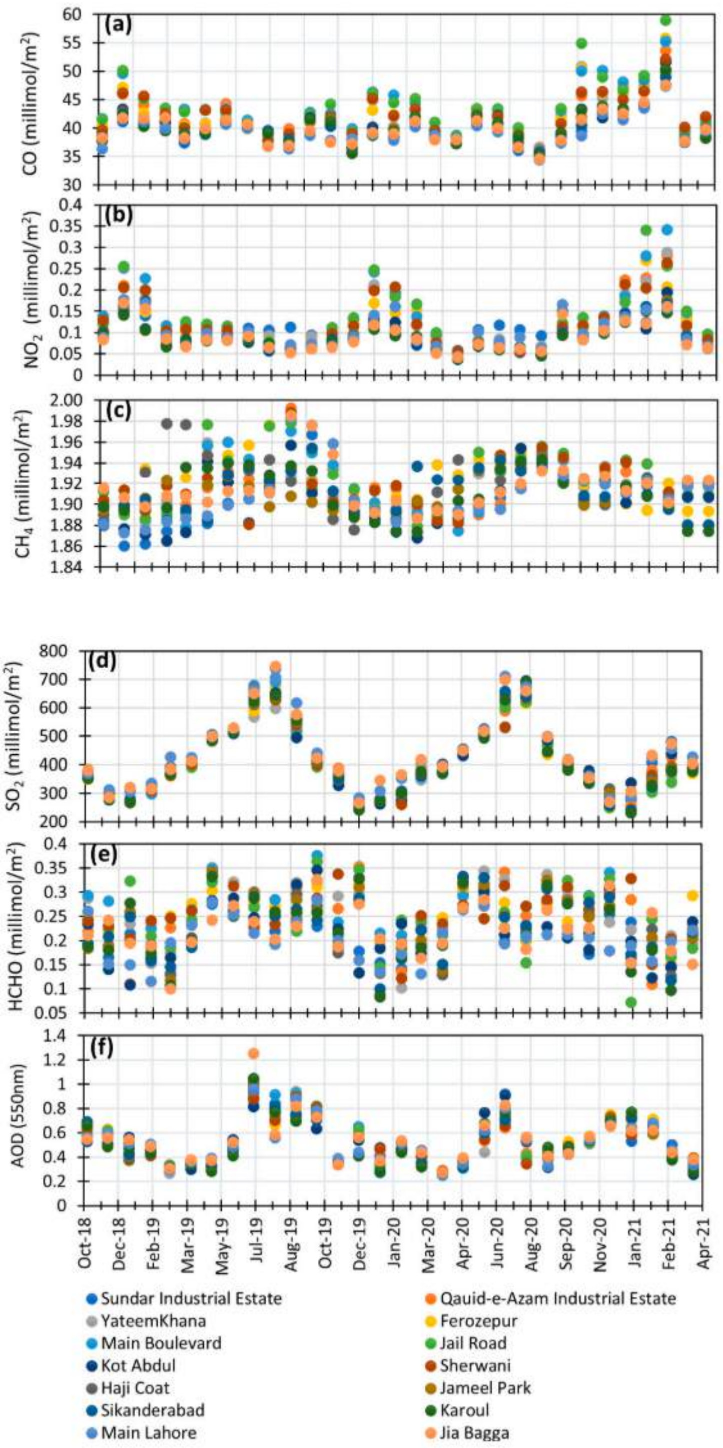


Figure 2.2: Pollutant concentrations were measured across three sectors (roads, industrial estates, and brick kilns)

Plotted in Figure 2.2 is the annual cost of pollution in unusual financial sports. The outcome served as an example of the most harmful contaminant at each location, necessitating an immediate remedial action plan. The reason for its heightened awareness can be explored with the help of on-floor data, providing guidance towards desired policy actions. The areas with the highest CO had been Quaid-e-Azam Industrial Estate (within business estates), Sherwani (inside brick kilns), and Jail Road (inside roads). Therefore, CO filters should be used in those areas going forward while mitigating measures are implemented or surveillance is carried out. Ferozepur reached its maximum SO2 contribution, although its contribution to CO and NO2 has decreased significantly. In brick kilns, Jia Bagga and Kot Abdul were the ultimate figures of SO2. Despite having multiple enterprises accredited to ISO 14001, Sundar Industrial Estate was the source of the most SO2 emissions.

Ferozepur reached its maximum SO2 contribution, although its contribution to CO and NO2 has decreased significantly. In brick kilns, Jia Bagga and Kot Abdul were the ultimate figures of SO2. Despite having multiple enterprises accredited to ISO 14001, Sundar Industrial Estate was the source of the most SO2 emissions.

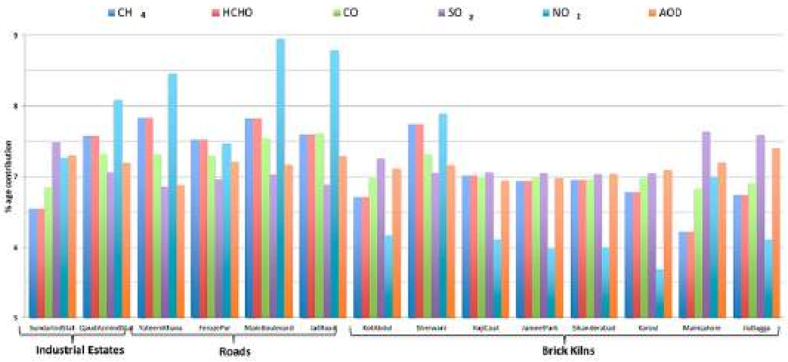


Figure 2.3: Percentage contribution of the corresponding pollutant.

2.3 Modeling

This study uses mathematical models to examine changes in air smog concentration in different cities of Pakistan. Integrated variables such as plant density x(t), population density y(t), and smoke density z(t) over time t. Physical factors include emissions from various sources, human and natural consumption rates, interactions between plants and humans, and environmental carrying capacity. Differential equations have been derived to describe growth and interactions between plants, humans, and smoke concentrations. Methods involving the trapezoidal transformation law (MTR) are used to simulate and analyze the behavior of the system over different time periods. The findings provide insight into the interaction between urbanization, emissions and environmental sustainability in Pakistan.

2.3.1 Mathematical Model

In this study, we will try to reveal how air pollution and problems affect different regions of Pakistan. Much of the smoke emitted from vehicles, industrial pollutants, and ventilation systems such as electric stoves is consumed by plants and humans. This research is based only on mathematical models that provide greater accuracy and can predict the ultimate effects of smoke emissions. Among them, $x(t)$ represents the vegetation density, $\mathrm{y}(t)$ represents the human density, $\mathrm{z}(t)$ represents the initial smoke density at time t [25] respectively. Let the total amount of smoke in the environment be Q and let the emissions $\delta, \delta_0, \delta_1 and \delta_2$ be CO, representing the fuel consumption coefficient of people caused by smoke, and the plant consumption coefficient, the cost coefficient of plants, and the cost consumption coefficient. The amount xy is the interaction value. Decrease and increase diversity due to competition. These examples are αxy and e αxy respectively. Here $\alpha > 0$, let α be the predictive value plus the interaction value, and let e be the conversion factor. The environmental carrying capacity of plants and humans is K&L, respectively. Here, the growth of plants and humans is expressed in terms of r & S. When plants and humans smoke heavily, the consumption value when the smoke enters the atmosphere is shown as $\delta_1 xy$ and $\delta_2 yz$.

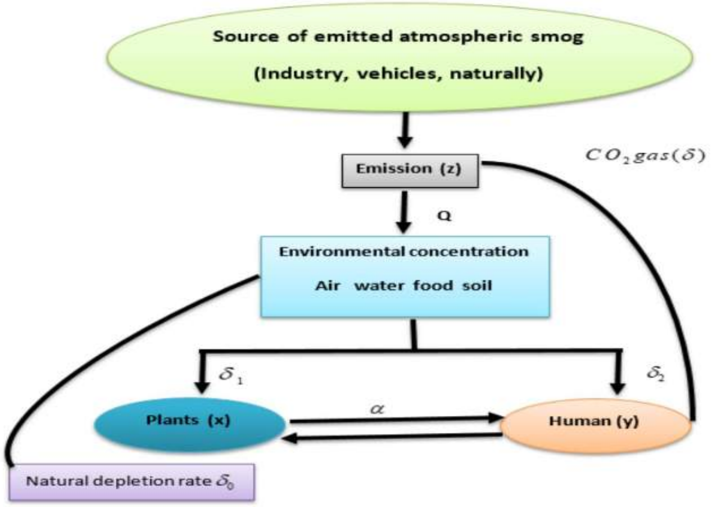

Source of emitted atmospheric smog
(Industry, vehicles, naturally)
Emission (z)
$CO_2 gas(\delta)$
Q
Environmental concentration
Air water food soil
δ_1
δ_2
α
Plants (x)
Human (y)
Natural depletion rate δ_0

Figure 2.4: Emitted Smog Concentration Model

2.3.2 Hypotheses

Let:

- $x(t)$: Vegetation Density
- $y(t)$: Human population Density
- $z(t)$: Smog concentration Density

The system is affected by various factors at time t:

- Q: Total increase in smog concentration due to emissions
- $\delta_1, \delta_2, \delta_{yz}$: Emitted CO2 gas coefficient rate, natural depletion coefficient rate of smog, and depletion coefficient rate due to absorption by humans, respectively
- α: Interaction rate between plants and humans
- β: Interaction rate causing changes in species due to competition
- e: Conversion factor

- K: Carrying capacity of vegetation
- L: Carrying capacity of humans
- r: Integral growth rate of plants
- s: Integral growth rate of humans
- δ_{xy}: Depletion rate of smog consumed by Vegetation

Differential Equations

1. **Plants' Dynamics:**

$$\frac{dx(t)}{dt} = rx(t)\left(1 - \frac{x(t)}{K}\right) - \alpha x(t)y(t) - \delta_3 x(t)z(t)$$

Explanation:

- $rx(t)\left(1 - \frac{x(t)}{K}\right)$: Logistic growth of vegetation considering carrying capacity K
- $-\alpha x(t)y(t)$: Interaction term where vegetation are affected by human activities
- $-\delta_3 x(t)z(t)$: Depletion of plants due to smog absorption

2. **Humans' Dynamics:**

$$\frac{dy(t)}{dt} = sy(t)\left(1 - \frac{y(t)}{L}\right) - \alpha x(t)y(t) - \delta_{xy} x(t)y(t)$$

Explanation:

- $sy(t)\left(1 - \frac{y(t)}{L}\right)$: Logistic growth of human population considering carrying capacity L
- $-\alpha x(t)y(t)$: Interaction term where humans affect plant density

- $-\delta_{xy}x(t)y(t)$: Depletion of humans due to smog absorption

3. **Smog Dynamics:**

$$\frac{dz(t)}{dt} = Q + \delta_1 y(t) - \delta_2 z(t) - \delta_{yz} y(t)z(t)$$

Explanation:

- Q: Constant increase in smog concentration due to emissions
- $\delta_1 y(t)$: Contribution of CO2 emissions from human activities
- $-\delta_2 z(t)$: Natural depletion of smog
- $-\delta_{yz} y(t)z(t)$: Depletion of smog due to absorption by humans

where:

- r, s: Vegetation and human intrinsic growth rates, respectively
- K, L: Carrying capacities of vegetation and humans, respectively
- α: Interaction rate between vegetation and humans
- $\delta_3, \delta_{xy}, \delta_1, \delta_2, \delta_{yz}$: Various coefficients affecting the dynamics of the system
- Q: Constant increase in smog concentration due to emissions

Modified Trapezoidal Rule (MTR) for Numerical Integration

To estimate the integral of $z(t)$ over the time interval $[0, T]$, we apply MTR [29]:

$$\int_0^T z(t)\,dt \approx \frac{\Delta t}{2}\left[z(0) + 2\sum_{i=1}^{N-1} z(t_i) + z(T)\right]$$

where:

- $\Delta t = \frac{T}{N}$ is the time step,
- $t_i = i \cdot \Delta t$, for $i = 0, 1, 2, \ldots, N$ are the time points,
- N is the number of subintervals,
- $z(t_i)$ are the values of smog concentration obtained from solving the differential equation system at time points t_i.

2.4 Implementation Steps

Step 1: Solving System of Differential Equations

Use numerical techniques (e.g., Runge-Kutta methods such as RK4) to solve the system of differential equations:

$$\frac{dx}{dt} = f(x, y, z), \quad \frac{dy}{dt} = g(x, y, z), \quad \frac{dz}{dt} = h(x, y, z)$$

under basic circumstances $x(0), y(0), z(0)$.

Step 2: Discretize Time and Compute Values of $z(t_i)$

Divide the time interval $[0, T]$ into small intervals. Compute $z(t_i)$ for each interval using the numerical solution obtained in Step 1.

Step 3: Apply the Modified Trapezoidal Rule (MTR)

To compute the integral of $z(t)$ over the interval $[0, T]$, apply the Modified Trapezoidal Rule (MTR):

Differential Equation for Smog Concentration

The differential equation for smog concentration $z(t)$ is given by:

$$\frac{dz}{dt} = Q + \delta_1 y - \delta_2 z - \delta_{yz} yz$$

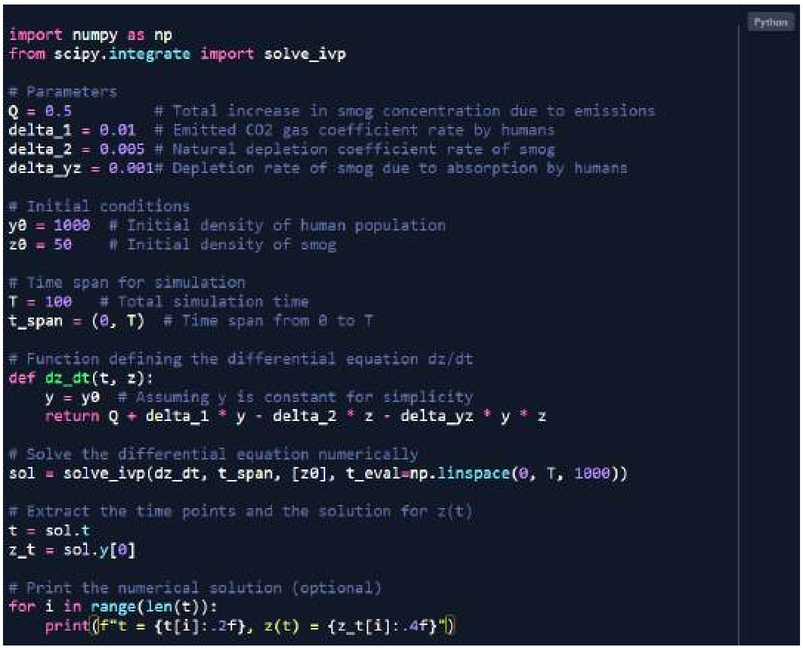
where:

- $z(t)$: Density of smog concentration at time t
- Q: Total increase in smog concentration due to emissions
- δ_1: Emitted CO2 gas coefficient rate by humans
- δ_2: Natural depletion coefficient rate of smog
- δ_{yz}: Depletion rate of smog due to absorption by humans
- y: Density of human population

2.4.1 Numerical Simulation:

Python Code

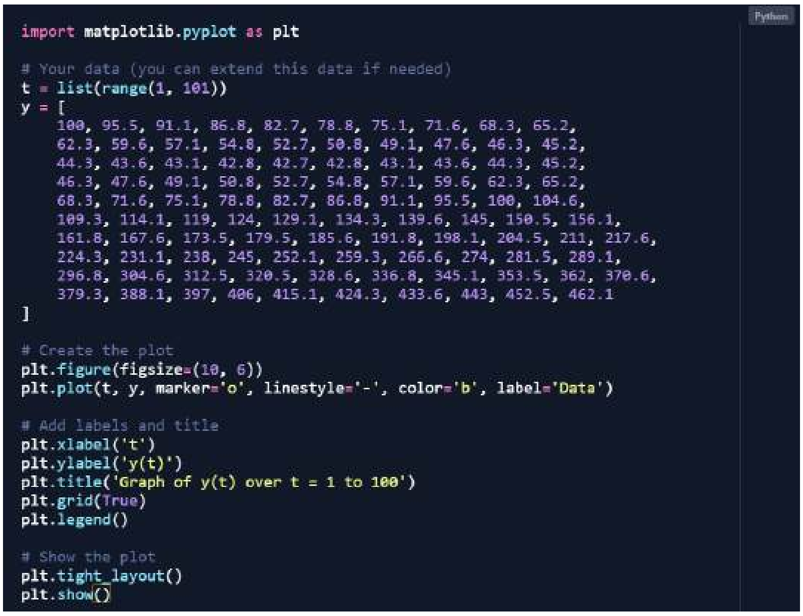
Below is the Python code for numerically solving the differential equation describing smog concentration dynamics:



```python
import numpy as np
from scipy.integrate import solve_ivp

# Parameters
Q = 0.5         # Total increase in smog concentration due to emissions
delta_1 = 0.01  # Emitted CO2 gas coefficient rate by humans
delta_2 = 0.005 # Natural depletion coefficient rate of smog
delta_yz = 0.001# Depletion rate of smog due to absorption by humans

# Initial conditions
y0 = 1000  # Initial density of human population
z0 = 50    # Initial density of smog

# Time span for simulation
T = 100   # Total simulation time
t_span = (0, T)  # Time span from 0 to T

# Function defining the differential equation dz/dt
def dz_dt(t, z):
    y = y0  # Assuming y is constant for simplicity
    return Q + delta_1 * y - delta_2 * z - delta_yz * y * z

# Solve the differential equation numerically
sol = solve_ivp(dz_dt, t_span, [z0], t_eval=np.linspace(0, T, 1000))

# Extract the time points and the solution for z(t)
t = sol.t
z_t = sol.y[0]

# Print the numerical solution (optional)
for i in range(len(t)):
    print(f"t = {t[i]:.2f}, z(t) = {z_t[i]:.4f}")
```

Table 2.1: Data Table

t	10	20	30	40	50	60	70	80	90	100
$y(t)$	88.4	78.4	69.2	61.1	53.8	47.3	41.4	36.1	31.3	27.0



```python
import matplotlib.pyplot as plt

# Your data (you can extend this data if needed)
t = list(range(1, 101))
y = [
    100, 95.5, 91.1, 86.8, 82.7, 78.8, 75.1, 71.6, 68.3, 65.2,
    62.3, 59.6, 57.1, 54.8, 52.7, 50.8, 49.1, 47.6, 46.3, 45.2,
    44.3, 43.6, 43.1, 42.8, 42.7, 42.8, 43.1, 43.6, 44.3, 45.2,
    46.3, 47.6, 49.1, 50.8, 52.7, 54.8, 57.1, 59.6, 62.3, 65.2,
    68.3, 71.6, 75.1, 78.8, 82.7, 86.8, 91.1, 95.5, 100, 104.6,
    109.3, 114.1, 119, 124, 129.1, 134.3, 139.6, 145, 150.5, 156.1,
    161.8, 167.6, 173.5, 179.5, 185.6, 191.8, 198.1, 204.5, 211, 217.6,
    224.3, 231.1, 238, 245, 252.1, 259.3, 266.6, 274, 281.5, 289.1,
    296.8, 304.6, 312.5, 320.5, 328.6, 336.8, 345.1, 353.5, 362, 370.6,
    379.3, 388.1, 397, 406, 415.1, 424.3, 433.6, 443, 452.5, 462.1
]

# Create the plot
plt.figure(figsize=(10, 6))
plt.plot(t, y, marker='o', linestyle='-', color='b', label='Data')

# Add labels and title
plt.xlabel('t')
plt.ylabel('y(t)')
plt.title('Graph of y(t) over t = 1 to 100')
plt.grid(True)
plt.legend()

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

Figure 2.5: Graph code of y(t) over t = 1 to 100

2.4.2 Discretization of Time Interval

After obtaining the numerical solution using the Runge-Kutta (RK) method, we can discretize the time interval $[0, T]$ into N subintervals [30] and compute $z(t_i)$ for each time step t_i:

$$t_i = \frac{i \cdot T}{N}, \quad i = 0, 1, 2, \ldots, N$$

$$\Delta t = \frac{T}{N}$$

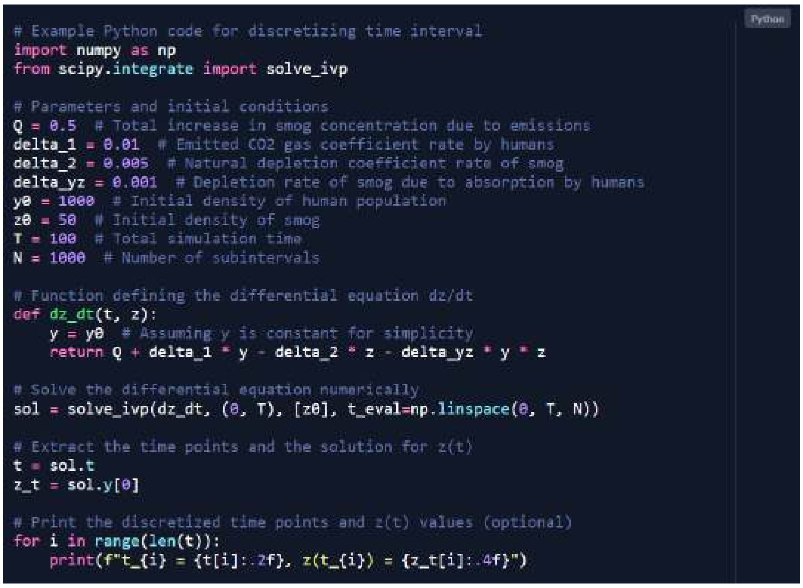
where:

- T: Total simulation time
- N: Number of subintervals
- t_i: Time points at which to evaluate $z(t)$
- Δt: Time step size

The computed values $z(t_i)$ are gathered numerically for the differential equation.

Code

Here's an example of how to implement this in Python after obtaining the numerical solution:



```python
# Example Python code for discretizing time interval
import numpy as np
from scipy.integrate import solve_ivp

# Parameters and initial conditions
Q = 0.5  # Total increase in smog concentration due to emissions
delta_1 = 0.01  # Emitted CO2 gas coefficient rate by humans
delta_2 = 0.005  # Natural depletion coefficient rate of smog
delta_yz = 0.001  # Depletion rate of smog due to absorption by humans
y0 = 1000  # Initial density of human population
z0 = 50  # Initial density of smog
T = 100  # Total simulation time
N = 1000  # Number of subintervals

# Function defining the differential equation dz/dt
def dz_dt(t, z):
    y = y0  # Assuming y is constant for simplicity
    return Q + delta_1 * y - delta_2 * z - delta_yz * y * z

# Solve the differential equation numerically
sol = solve_ivp(dz_dt, (0, T), [z0], t_eval=np.linspace(0, T, N))

# Extract the time points and the solution for z(t)
t = sol.t
z_t = sol.y[0]

# Print the discretized time points and z(t) values (optional)
for i in range(len(t)):
    print(f"t_{i} = {t[i]:.2f}, z(t_{i}) = {z_t[i]:.4f}")
```

2.4.3 Modified Trapezoidal Rule for Smog Density

Given:

- $z(t_k)$: Numerical solution of smog density at time t_k.
- h: Time step Δt.
- N: Number of subintervals.

The modified equation for the MTR applied to $z(t)$ is:

$$T(z,h) = \frac{h}{2}\sum_{k=1}^{N}\left(z(t_{k-1}) + z(t_k)\right)$$

where:

- $z(t_{k-1})$ and $z(t_k)$ are the values of smog density at time points t_{k-1} and t_k, respectively.
- $h = \frac{T}{N}$ is the time step.

Parameters: Set the total simulation time T, and the number of subintervals N, and compute the time step $\Delta t = \frac{T}{N}$.

Numerical Solution: Assume `z_t` contains the numerical solution of $z(t)$ obtained from a previous step (e.g., using a numerical method like Runge-Kutta).

Chapter 3

Modified Trapezoidal Rule & Caputo Fractional Derivative

It involves the application and analysis of computational methods to approximate fractional integrals. It provides a modified trapezoidal rule that extends the classical trapezoidal rule for solving mixed fractions. This formula calculates the function f(x) over a parameter using a weighted parameter that is not modified by the h and alpha parameters. This document provides detailed background information, including error analysis and component composition. It also includes MATLAB code snippets for using math functions with functions such as sin(x) and cos(x), providing precision and sensitivity to alpha. Overall, it serves as a general guide to understanding and applying the trapezoidal adjustment rule in the context of proportional analysis. It also works on the calculation and analysis of Caputo fractional derivatives of various functions and parameters. Caputo defines fractional derivatives as sums of functions of the same value required to solve the difference between multiple equations. The theoretical framework is designed to cover the concepts and properties of Caputo's fractional derivatives, with an emphasis on numerical approaches using the modified trapezoidal law. This includes MATLAB code for calculating derivatives, as well as numerical results and error analysis for functions such as sine and cosine. This work demonstrates the sensitivity of Caputo fractional derivatives to various orders and step sizes, demonstrating the best way to estimate the correct number.

3.1 Modified Trapezoidal Rule

Proof. Use the modified MTR formula to compute the integral of $z(t)$ over the interval $[0, T]$:

$$\int_0^T z(t)\,dt \approx \frac{\Delta t}{2}\left[z(0) + 2\sum_{i=1}^{N-1} z(t_i) + z(T)\right] \tag{3.1.1}$$

where:

- $z(0)$ and $z(T)$ are the initial and final values of smog density, respectively.
- $\Delta t = \frac{T}{N}$ is the time step.
- $z(t_i)$ are the values of smog density at discrete time points t_i, obtained from the numerical solution.

An approximation to the integral of f(x) over [y, z]
$= \frac{h}{2}(f(y) + f(z)) + h\sum_{k=1}^{N-1} f(x_k)$
We can write

$$\int_y^z f(x)dx \approx T(f, h) \tag{3.1.2}$$

□

3.1.1 Error Analysis of Trapezoidal Rule;

As a result, if the integrand is concave up (and so has an excellent second derivative), the error is poor, and the trapezoidal approach overestimates the true fee [26]. If $f(x) \in C^2[a,b]$, then there is a value c with $\mathrm{a} < \mathrm{c} < \mathrm{b}$ so that the error term $\mathrm{E(f,h)}$ has the form [26]

$$E(f,h) = \frac{-(z-y)f''(c)h^2}{12} = O(h^2) \tag{3.1.3}$$

Where $E(f,h) = \int_y^z f(x)dx - T(f,h)$

The benefits of fractional order integrals/derivatives over classical integrals/derivatives are that we get additional degrees of freedom by using fractional order integrals, and fractional order derivatives have a resemblance of properties that classical derivatives do not.

Definition 3.1.1. The Riemann-Liouville technique of fractional calculus defines the fractional integral of order $\alpha > 0$ as follows: [14].

$$J^\alpha f(x) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-\tau)^{\alpha-1} f(\tau)d\tau, x > 0. \tag{3.1.4}$$

The definition and properties of the J^α operator can be found in [9,11,12], we quote the following: For $\alpha, \beta > 0, x > 0\gamma > -1$, we have

$$J^\alpha J^\beta = J^{\alpha+\beta} \tag{3.1.5}$$

$$J^\alpha x^\gamma = \frac{\Gamma(\gamma+1)}{\Gamma(\gamma+1+\alpha)} x^{\gamma+\alpha} \tag{3.1.6}$$

$$J^\alpha \mathrm{e}^{ax} = x^\alpha \sum_{k=0}^{\infty} \frac{(ax)^k}{\Gamma(\alpha+k+1)} \tag{3.1.7}$$

$$J^\alpha \cos(ax) = x^\alpha \sum_{k=0}^{\infty} \frac{(-1)^k (ax)^{2k}}{\Gamma(\alpha+2k+1)} \tag{3.1.8}$$

$$J^{\alpha} \sin(ax) = x^{\alpha} \sum_{k=0}^{\infty} \frac{(-1)^k (ax)^{2k+1}}{\Gamma(\alpha + 2k + 2)} \tag{3.1.9}$$

3.2 Theorem

Proof. Suppose that $f \in C^2[0,T]$, $\tilde{f}_k$ is the equivalent linear interpolation [26] for f and the node is chosen from $t_j = jh$ with $h = T/k, j = 0, 1, 2, \ldots, k$, then

(i)

$$\int_0^{t_k} (t_k - t)^{\alpha-1} \tilde{f}_k(t) \mathrm{d}t = \sum_{j=0}^{k} a_{j,k} \cdot f\left(t_i\right. \tag{3.2.1}$$

where

$$a_{j,k} = \tfrac{h^{\alpha}}{\alpha(\alpha+1)} \begin{cases} (k-1)^{\alpha+1} - (k-1-\alpha)k^{\alpha}, & j = 0, \\ (k-j+1)^{\alpha+1} + (k-j-1)^{\alpha+1} - 2(k-j)^{\alpha+1}, & 1 \leqslant j \leqslant k-1, \\ 1, & j = k, \end{cases}$$

(ii)

$$\left| \int_0^{t_k} (t_k - t)^{\alpha-1} f(t) \mathrm{d}t - \sum_{j=0}^{k} a_{j,k} \cdot f\left(t_i\right) \right| \leqslant C_{\alpha} \left\| f'' \right\|_{\infty} t_k^{\alpha} h^2 \tag{3.2.2}$$

for some constant C_{α} depending only on α. □

3.3 Theorem

Theorem 3.3.1. *Assume that the interval* $[0, a]$ *is partitioned into* k *subintervals* $[x_j, x_{j+1}]$ *of equal width* $h = alk$ *by using the nodes* $x_j = jh$*, for* $j = 0, 1, \ldots, k$*. The modified trapezoidal rule [26]*

$$\begin{aligned} T(f,h,\alpha) = & \left((k-1)^{\alpha+1} - (k-\alpha-1)k^{\alpha}\right) \frac{h^{\alpha} f(0)}{\Gamma(\alpha+2)} + \frac{h^{\alpha} f(a)}{\Gamma(\alpha+2)} \\ & + \sum_{j=1}^{k-1} \left((k-j+1)^{\alpha+1} - 2(k-j)^{\alpha+1} + (k-j-1)^{\alpha+1}\right) \frac{h^{\alpha} f\left(x_j\right)}{\Gamma(\alpha+2)} \end{aligned}$$

is an approximation to fractional integral

$$\left(J^{\alpha} f(x)\right)(a)=T(f, h, \alpha)-E_{T}(f, h, \alpha), \quad a>0, \quad \alpha>0 \tag{3.3.1}$$

Furthermore, if $f(x) \in \boldsymbol{C}^{2}[0, a]$, there is a constant $C_{\alpha}^{\prime}$ depending only on α so that the error term $E_{T}(f, h, \alpha)$ has the form

$$\left|E_{T}(f, h, \alpha)\right| \leqslant C_{\alpha}^{\prime}\left\|f^{\prime \prime}\right\|_{\infty} a^{\alpha} h^{2}=\mathbf{O}\left(h^{2}\right) \tag{3.3.2}$$

Proof. **Proof:**

From definition 3.1.1, we have

$$\left(J^{\alpha} f(x)\right)(a)=\frac{1}{\Gamma(\alpha)} \int_{0}^{a}(a-\tau)^{\alpha-1} f(\tau) \mathrm{d} \tau \tag{3.3.3}$$

If $\tilde{f}_{k}$ is the piecewise linear interpolant for f whose nodes are chosen at the nodes $x_{j}, j=0,1,2, \ldots, k$, then, using theorem 3.2, we obtain

$\int_{0}^{t_{k}}\left(t_{k}-t\right)^{\alpha-1} \tilde{f}_{k}(t) \mathrm{d} t=\sum_{j=0}^{k} a_{j, k} \cdot f\left(t_{i}\right)$ where

$$a_{j, k}=\frac{h^{\alpha}}{\alpha(\alpha+1)}\begin{cases}(k-1)^{\alpha+1}-(k-1-\alpha) k^{\alpha}, & j=0, \\ (k-j+1)^{\alpha+1}+(k-j-1)^{\alpha+1}-2(k-j)^{\alpha+1}, & 1 \leqslant j \leqslant k-1, \\ 1, & j=k,\end{cases}$$

$$\int_{0}^{t_{k}}\left(t_{k}-t\right)^{\alpha-1} \tilde{f}_{k}(t) \mathrm{d} t=\frac{h^{\alpha} f(0)}{\alpha(\alpha+1)}\left((k-1)^{\alpha+1}-(k-1-\alpha) k^{\alpha}\right)+\frac{h^{\alpha} f\left(t_{k}\right)}{\alpha(\alpha+1)}+ \frac{h^{\alpha} f\left(x_{j}\right)}{\alpha(\alpha+1)}\left(\sum_{j=1}^{k-1} \quad(k-j+1)^{\alpha+1}+(k-1-j)^{\alpha+1}-2(k-j)^{\alpha+1}\right)$$

We get the theorem first equation,

$$\begin{aligned} T(f, h, \alpha)= & \left((k-1)^{\alpha+1}-(k-\alpha-1) k^{\alpha}\right) \frac{h^{\alpha} f(0)}{\Gamma(\alpha+2)}+\frac{h^{\alpha} f(a)}{\Gamma(\alpha+2)} \\ & +\sum_{j=1}^{k-1}\left((k-j+1)^{\alpha+1}-2(k-j)^{\alpha+1}+(k-j-1)^{\alpha+1}\right) \frac{h^{\alpha} f\left(x_{j}\right)}{\Gamma(\alpha+2)}\end{aligned}$$

Now using the second equation of theorem 3.2, we get

$\left|\int_0^{t_k}\left(t_k-t\right)^{\alpha-1} f(t) \mathrm{d} t-\sum_{j=0}^k a_{j, k} \cdot f\left(t_i\right)\right| \leqslant C_\alpha\left\|f^{\prime \prime}\right\|_{\infty} t_k^\alpha h^2$

Here, $\sum_{j=0}^k \quad a_{j.k} f\left(t_j\right)=\int_o^a(a-\tau)^{\alpha-1} f_k^{\sim}(\tau) d \tau$

$$\left|\int_0^a(a-\tau)^{\alpha-1} f(\tau) d \tau-\int_o^a(a-\tau)^{\alpha-1} f_k^{\sim}(\tau) d \tau\right| \leq C_\alpha\left|\left|f^{\prime \prime}\right|_{\infty} a^\alpha h^2 \quad (3.3.4)$$

$$E_T(f, h, \alpha)=T(f, h, \alpha)-\left(J^\alpha f(x)\right)(a) \quad (3.3.5)$$

$$\begin{aligned}
E_T(f, h, \alpha) & =\int_0^a(a-\tau)^{\alpha-1} f(\tau) d \tau-\int_o^a(a-\tau)^{\alpha-1} f_k^{\sim}(\tau) d \tau \\
\left|E_T(f, h, \alpha)\right| & =\left|\int_0^a(a-\tau)^{\alpha-1} f(\tau) d \tau-\int_0^a(a-\tau)^{\alpha-1} f_k^{\sim}(\tau) d \tau\right| \\
& \leq C_\alpha\left|\left|f^{\prime \prime}\right|\right|_{\infty} a^\alpha h^2\left|E_T(f, h, \alpha)\right| \leq C_\alpha\left|\left|f^{\prime \prime}\right|\right|_{\infty} a^\alpha h^2 \\
\left|E_T(f, h, \alpha)\right| & \leq C_\alpha^{\prime} \quad\left|\left|f^{\prime \prime}\right|\right|_{\infty} a^\alpha h^2 \quad \text{where } C_\alpha^{\prime}=C_\alpha / \Gamma(\alpha)
\end{aligned}$$

The method's behavior is clearly independent of the parameter a, and it acts similarly to the conventional trapezoidal rule. Specifically, if a = 1, the modified trapezoidal rule simplifies to the trapezoidal rule.

$$E(f, h)=\frac{-(z-y) f^{\prime \prime}(c) h^2}{12}=O\left(h^2\right) \quad (3.3.6)$$

$$\begin{aligned}
T(f, h, \alpha)= & \frac{h^\alpha f(0)}{\Gamma(\alpha+2)}\left((k-1)^{\alpha+1}-(k-1-\alpha) k^\alpha\right)+\frac{h^\alpha f(a)}{\Gamma(\alpha+2)} \\
& +\frac{h^\alpha f\left(x_j\right)}{\Gamma(\alpha+2)}\left((k-j+1)^{\alpha+1}+(k-1-j)^{\alpha+1}-2(k-j)^{\alpha+1}\right) \\
T(f, h, 1)= & \frac{h f(0)}{\Gamma(3)}\left((k-1)^2-(k-2) k\right)+\frac{h f(a)}{\Gamma(3)} \\
& +\frac{h f\left(x_j\right)}{\Gamma(3)}\left((k-j+1)^2+(k-1-j)^2-2(k-j)^2\right)
\end{aligned}$$

Given that $\Gamma(3)=2$, we simplify the expression:

$$T(f,h,1)=\frac{hf(0)}{2}\left((k-1)^2-(k-2)k\right)+\frac{hf(a)}{2}+$$
$$\frac{hf\left(x_j\right)}{2}\left((k-j+1)^2+(k-1-j)^2-2(k-j)^2\right)$$
$$T(f,h,1)=\frac{hf(0)}{2}\left(k^2+1-2k-k^2+2k\right)+\frac{hf(a)}{2}+$$
$$\frac{hf\left(x_j\right)}{2}\left(k^2+j^2+1-2kj-2j+2k+k^2+1+j^2-2k+2j-2jk-2k^2-2j^2\right.$$
$$\left.+4kj\right)T(f,h,1)$$

Further simplifying:

$$=\frac{hf(0)}{2}(1)+\frac{hf(a)}{2}+\frac{hf(x_j)}{2}\left(2k^2+2j^2+2-4kj-2k^2-2j^2+4kj\right)$$
$$T(f,h,1)=\frac{hf(0)}{2}+\frac{hf(a)}{2}+\frac{hf(x_j)}{2}$$
$$T(f,h)=\frac{hf(0)}{2}+\frac{hf(a)}{2}+hf(x_j)$$

An approximation to the integral of $f(x)$ over $[a,b]$

$$=\frac{h}{2}(f(a)+f(b))+h\sum_{k=1}^{M-1}f\left(x_k\right)$$

□

3.4 Application

3.4.1 Application 1

"Consider the function $f(x)=\sin x$, apply the modified trapezoidal rule to approximate the fractional integral $\left(J^\alpha f(x)\right)(1)$" [26]

MATLAB Code

```
% Inputs
h = 0.1;
x = 0:h:1;
k = length(x)-1;
alpha = 0.5;
f0 = sin(x(1));
fa = sin(x(end));

% The algorithm
tic;
j = 1:k-1;

% Calculate RL_Int using the modified trapezoidal rule
RL_Int = (h^alpha / gamma(alpha+2)) * ...
      (((k-1)^(alpha+1) - (k-alpha-1) * k^alpha) * sin(x(1))
          + ...
      sin(x(end)) + sum(((k-(1:k-1)+1).^(alpha+1) - 2*(k-(1:
         k-1)).^(alpha+1) + (k-(1:k-1)-1).^(alpha+1)) .* sin
         (x(2:end-1))));
toc;

% Exact solution
syms i t
Exactl = eval(symsum((-1)^i / gamma(alpha+2*i+2), i, 0, Inf)
   );
Error = abs(Exactl - RL_Int);

% Display results
disp('k␣h␣T(f,h,alpha)␣E_(T)(f,h,alpha')
format long; % Set the format to display 10 digits after the
    decimal point
Results_Table = [k h RL_Int Error];
disp(Results_Table);
```

Results Table

Alpha	k	h	C(f, h, alpha)	E_C(f, h, alpha)
0.1	10.0	0.1	0.815408440922776	0.000244317455723
0.2	10.0	0.1	0.784067069046118	0.000388114970539
0.3	10.0	0.1	0.748566650039401	0.000465519878109
0.4	10.0	0.1	0.709951982237630	0.000499642601201
0.5	10.0	0.1	0.669178250902067	0.000506008675596
0.6	10.0	0.1	0.627102058194136	0.000494970526387
0.7	10.0	0.1	0.584477142390522	0.000473383321213
0.8	10.0	0.1	0.541953894199521	0.000445756915958
0.9	10.0	0.1	0.500081864613166	0.000415039476613
1.0	10.0	0.1	0.459314548857976	0.000383145273884

Table 3.1: Results for Modified Trapezoidal Rule of $\sin(x)$ for different α values.

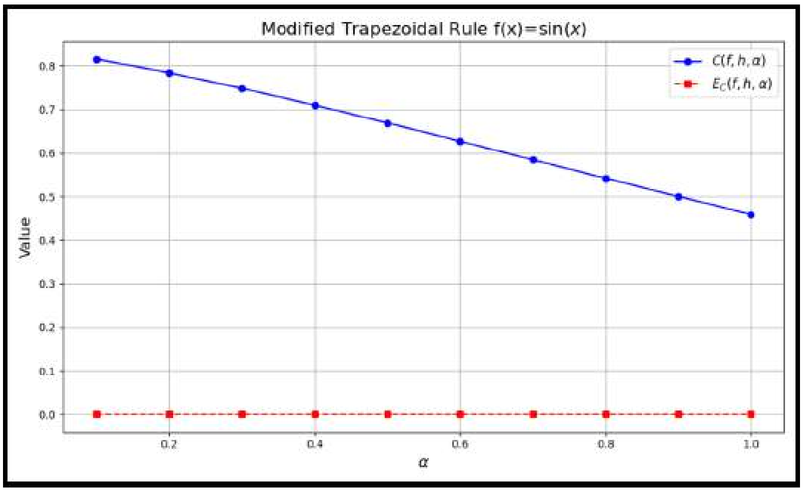


Figure 3.1: Results for Modified Trapezoidal Rule of $\sin(x)$ for different α values.

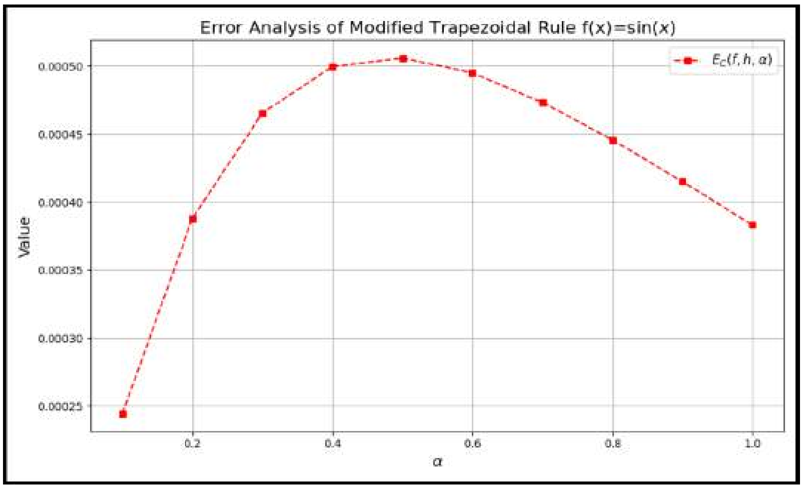


Figure 3.2: Error Analysis

3.4.2 Application 2

Consider the function $f(x) = \cos x$

MATLAB Code

```
% Inputs
h = 0.1;
x = 0:h:1;
k = length(x)-1;
alpha = 0.5;
f0 = cos(x(1));
fa = cos(x(end));

% The algorithm
tic;
j = 1:k-1;

% Calculate RL_Int using the modified trapezoidal rule
```

```
RL_Int = (h^alpha / gamma(alpha+2)) * ...
      (((k-1)^(alpha+1) - (k-alpha-1) * k^alpha) * cos(x(1))
          + ...
      cos(x(end)) + sum(((k-(1:k-1)+1).^(alpha+1) - 2*(k-(1:
         k-1)).^(alpha+1) + (k-(1:k-1)-1).^(alpha+1)) .* cos
         (x(2:end-1))));
toc;

% Exact solution
syms i t
Exactl = eval(symsum((-1)^i / gamma(alpha+2*i+2), i, 0, Inf)
   );
Error = abs(Exactl - RL_Int);

% Display results
disp('k␣h␣T(f,h,alpha)␣E_(T)(f,h,alpha')
format long; % Set the format to display 10 digits after the
    decimal point
Results_Table = [k h RL_Int Error];
disp(Results_Table);
```

Results Table

Alpha	k	h	C(f, h, alpha)	E_C(f, h, alpha)
0.1	10.0	0.1	0.630521382269243	0.185131376109256
0.2	10.0	0.1	0.705915320997643	0.078539863019014
0.3	10.0	0.1	0.766533841153199	0.017501671235689
0.4	10.0	0.1	0.812788207034219	0.102336582195388
0.5	10.0	0.1	0.845382987790107	0.175698728212444
0.6	10.0	0.1	0.845382987790107	0.175698728212444
0.7	10.0	0.1	0.873488319225052	0.288537793513317
0.8	10.0	0.1	0.859970196955327	0.359473292865548
0.9	10.0	0.1	0.871305331632147	0.328905680516667
1.0	10.0	0.1	0.840769642088420	0.381071947956560

Table 3.2: Results for Modified Trapezoidal Rule of $\cos(x)$ for different α values.

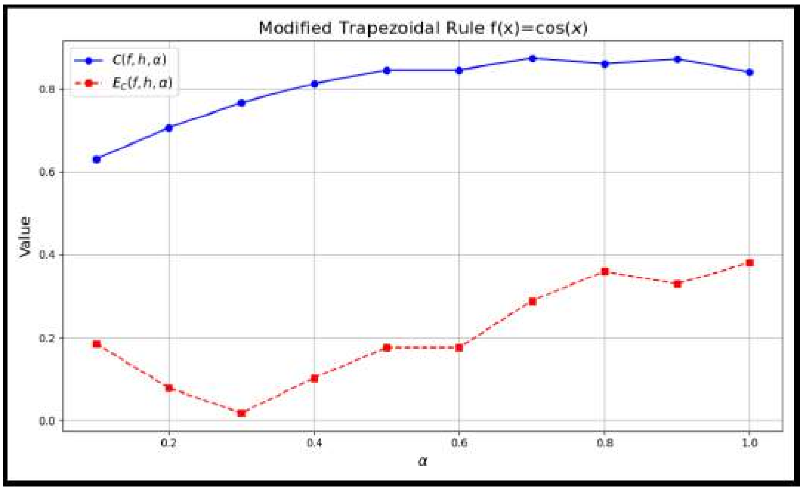


Figure 3.3: Results for Modified Trapezoidal Rule of $\cos(x)$ for different α values.

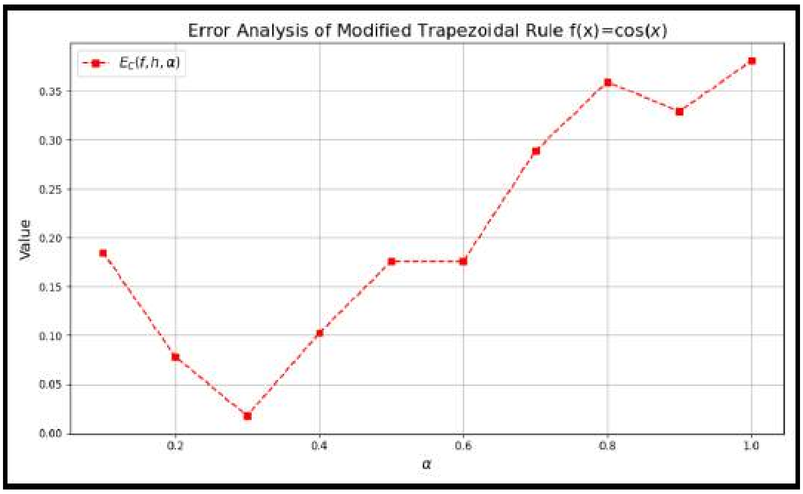


Figure 3.4: Error Analysis

3.5 Conclusion

The error behavior of the modified trapezoidal rule applied to trigonometric functions, specifically $\sin(x)$ and $\cos(x)$. We analyzed the sensitivity of the method to the parameter α, which ranges from 0 to 1, with the error denoted as $E_C(f, h, \alpha)$.

The error behavior exhibits significant variability across the range of α. Notably, the minimum error occurs at approximately $\alpha = 0.3$, with an E_c value of about 0.02. As α increases from 0.5 to 0.8, we observe a substantial rise in error, with some fluctuations. Interestingly, the error at $\alpha = 0.5$ is about 0.18, marking the beginning of a steeper error increase. The maximum error in the given range occurs at $\alpha = 0.9$, with an E_c value of approximately 0.36.For the sine function $\sin(x)$, as shown in Image 2: The error pattern is markedly different and more predictable compared to $\cos(x)$. The error reaches its maximum at $\alpha = 0.5$, with an E_c value of about 0.00051. From $\alpha = 0.5$ and $\alpha = 1$, we observe a consistent decrease in error. Importantly, the overall magnitude of error for $\sin(x)$ is significantly smaller than for $\cos(x)$, by approximately three orders of magnitude.

These findings highlight the function-dependent nature of the modified trapezoidal rule's performance. In conclusion, this analysis underscores the importance of carefully selecting the α parameter in the modified trapezoidal rule, as its optimal value can vary significantly depending on the function under consideration. For numerical applications involving similar trigonometric functions, these results suggest that tailoring the choice of α to the specific function could substantially improve accuracy and reduce computational errors.

3.6 Caputo Fractional Derivative Rule

In this part, we chose to apply the approach to compare the Caputo fractional derivatives of a function of order $\alpha > 0$ with regard to the weight of the function and the same value as the supplied value. Our technique is based on the description (3.1.2) and the modified trapezoidal rule, which is the Caputo fractional growth of a function expressed as a factor.

Definition 3.6.1. Cupoto Fractional Derivative: Let m be the smallest integer that exceeds α, then Caputo fractional derivative of order $\alpha > 0$ is defined as [26]

$$D_*^\alpha f(x) = J^{(m-\alpha)}\left[f^{(m)}(x)\right] \tag{3.6.1}$$

namely

$$D_*^\alpha f(t) = \begin{cases} \frac{1}{\Gamma(m-\alpha)}\left[\int_0^x \frac{f^{(m)}(\tau)}{(x-\tau)^{\alpha+1-m}}\,\mathrm{d}\tau\right], & m-1<\alpha<m \\ \frac{\mathrm{d}^m}{\mathrm{d}x^m}f(x), & \alpha = m \end{cases} \tag{3.6.2}$$

Numerical derivative formulas are crucial in developing methods for solving boundary value problems in ordinary and partial differential equations. Numerical techniques for solving linear fractional differential equations are well established (see [3,5-8]). Diethelm et al. [8] provide an extension of the Adams-Bashforth-Moulton approach to estimate the solution to the nonlinear fractional differential equation.

$$D_*^\alpha y(t) = f(t, y(t)), \quad y\left(t_0\right) = y_0 \tag{3.6.3}$$

which is equivalent to the integral equation

$$y(t) = y\left(t_0\right) + \frac{1}{\Gamma(\alpha)} \int_{t_0}^t (t-\varsigma)^{\alpha-1} f(\varsigma, y(\varsigma)) \mathrm{d}\varsigma \tag{3.6.4}$$

They apply the product trapezoidal quadrature formula with regard to the weight function $(t_k -')^{\alpha-1}$. In other words, they employ an estimate.

$$\int_{t_0}^{t_k} \left(t_k - \varsigma\right)^{\alpha-1} f(\varsigma) \mathrm{d}\varsigma \approx \int_{t_0}^{t_k} \left(t_k - \varsigma\right)^{\alpha-1} \tilde{f}_k(\varsigma) \mathrm{d}\varsigma \tag{3.6.5}$$

The smallest piecewise linear interpolant for f is $\tilde{f}_k$, with nodes picked at $t_j, j = 0, 1, 2, ..., k$.

3.7 Theorem

Theorem 3.7.1. *"Assume that the interval $[0, a]$ is partitioned into k sub-intervals $[x_j, x_{j+1}]$ of equal width $h = alk$ by using the nodes $x_j = jh$, for $j = 0, 1, \ldots, k$, then the Caputo fractional derivative rule " [26]*

$$C(f, h, \alpha) = \frac{h^{m-\alpha}}{\Gamma(m+2-\alpha)} \left\{ \begin{array}{l} \left((k-1)^{m-\alpha+1} - (k-m+\alpha-1) k^{m-\alpha}\right) f^{(m)}(0) + \\ f^{(m)}(a) + \sum_{j=1}^{k-1} \left((k-j+1)^{m-\alpha+1} - 2(k-j)^{m-\alpha+1} + \right. \\ (k-j-1)^{m-\alpha+1} f^{(m)}\left(x_j\right) \end{array} \right. \tag{3.7.1}$$

is an approximation to the Caputo fractional derivative

$$\left(D_*^\alpha f(x)\right)(a) = C(f, h, \alpha) - E_C(f, h, \alpha), \quad a > 0 \tag{3.7.2}$$

for $m-1 < \alpha < m$. Furthermore, if $f(x) \in \boldsymbol{C}^{n+2}[0,a]$, then there is some constant $C'_{m-\alpha}$ depending only on α so that the error term $E_C(f,h,\alpha)$ has the form

$$|E_C(f,h,\alpha)| \leqslant C'_{m-\alpha} \left\| f^{(m+2)} \right\|_\infty a^{m-\alpha} h^2 = \mathbf{O}\left(h^2\right) \tag{3.7.3}$$

Proof:

Proof. Replacing α by $m-\alpha$ and $f(\tau)$ by $f^{(m)}(\tau)$ in Theorem 3.3.

$$C(f,h,\alpha) = \left((k-1)^{m-\alpha+1} - (k-m+\alpha-1)k^{m-\alpha}\right) \frac{h^{m-\alpha} f(0)}{\Gamma(m-\alpha+2)} + \frac{h^{m-\alpha} f(a)}{\Gamma(m-\alpha+2)} + \sum_{j=1}^{k-1} \left((k-j+1)^{m-\alpha+1} - 2(k-j)^{m-\alpha+1} + (k-j-1)^{m-\alpha+1}\right) \frac{h^{m-\alpha} f\left(x_j\right)}{\Gamma(m-\alpha+2)}$$

Using definition (3.6.1), we will compute just a finite number of mth ordinary derivatives of the function $f(x)$ at specified places to approximate the Caputo fractional derivative $D_*^\alpha f(x)$ of order $\alpha, m-1 < \alpha < m$.

In case of $0 < \alpha < 1$: and m=1, the Caputo fractional derivative rule (3.7.1) reduces as follows:

Substitute m = 1:

$$C(f,h,\alpha) = \frac{h^{1-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{k} w_j^{(1,\alpha)} f'(x_{k-j})$$

Expand the weights $w_j^{(1,\alpha)}$: $w_j^{(1,\alpha)} = (j+1)^{1-\alpha} - j^{1-\alpha}$ Separate the sum into three parts:

Term for $j=0$ (corresponding to $x_k = a$) Term for $j=k$ (corresponding to $x_0 = 0$) Sum for $1 \leq j \leq k-1$

Simplify each part:

For $j=0$: $w_0^{(1,\alpha)} = 1$ For $j=k$: $w_k^{(1,\alpha)} = (k+1)^{1-\alpha} - k^{1-\alpha}$ For $1 \leq j \leq k-1$: Use the full weight formula

Combine these parts and adjust the gamma function in the denominator:

$$C(f,h,\alpha)=\frac{h^{1-\alpha}}{\Gamma(3-\alpha)}\Big\{((k-1)^{2-\alpha}-(k+\alpha-2)k^{1-\alpha})f'(0)+f'(a)$$
$$+\sum_{j=1}^{k-1}((k-j+1)^{2-\alpha}-2(k-j)^{2-\alpha}+(k-j-1)^{2-\alpha})f'(x_j)\Big\}$$

This final form is the simplified version for $0<\alpha<1$ and $m=1$.

$$C(f,h,\alpha)=\frac{h^{1-\alpha}}{\Gamma(3-\alpha)}\Big\{((k-1)^{2-\alpha}-(k+\alpha-2)k^{1-\alpha})f'(0)+f'(a)$$
$$+\sum_{j=1}^{k-1}((k-j+1)^{2-\alpha}-2(k-j)^{2-\alpha}+(k-j-1)^{2-\alpha})f'(x_j)\Big\}$$

and if $f(x)\in\boldsymbol{C}^3[0,a]$, the error term $E_C(f,h,\alpha)$ takes the form

$$|E_C(f,h,\alpha)|\leqslant C'_{1-\alpha}\left\|f^{(3)}\right\|_\infty a^{1-\alpha}h^2$$

for some constant $C'_{1-\alpha}$ depending only on α.

If $1<\alpha<2$, the Caputo fractional derivative rule (3.7.1) yields the formula.

For $1<\alpha<2$, we have m = 2. Substitute this:

$$C(f,h,\alpha)=\frac{h^{2-\alpha}}{\Gamma(3-\alpha)}\sum_{j=0}^{k}w_j^{(2,\alpha)}f''(x_{k-j})$$

Expand the weights $w_j^{(2,\alpha)}$: $w_j^{(2,\alpha)}=(j+1)^{2-\alpha}-2j^{2-\alpha}+(j-1)^{2-\alpha}$

Separate the sum into three parts:

Term for j=0 (corresponding to $x_k=a$)

Term for j=k (corresponding to $x_0=0$)

The sum for $1\leq j\leq k-1$

Simplify each part:

For j=0: $w_0^{(2,\alpha)}=1$ For j=k: $w_k^{(2,\alpha)}=(k+1)^{2-\alpha}-2k^{2-\alpha}+(k-1)^{2-\alpha}$

For $1\leq j\leq k-1$: Use the full weight formula

Combine these parts and adjust the gamma function in the denominator:

$$C(f,h,\alpha)=\frac{h^{2-\alpha}}{\Gamma(4-\alpha)}\left\{\begin{array}{l}((k-1)^{3-\alpha}-(k+\alpha-3)k^{2-\alpha})\,f''(0)+f''(a)\\ +\sum_{j=1}^{k-1}((k-j+1)^{3-\alpha}-2(k-j)^{3-\alpha}+(k-j-1)^{3-\alpha})\,f''(x_j)\end{array}\right.$$

For $f(x)\in\boldsymbol{C}^4[0,a]$, the error term $E_C(f,h,\alpha)$ takes the form

$$|E_C(f,h,\alpha)| \leqslant C'_{2-\alpha} \left\| f^{(4)} \right\|_\infty a^{2-\alpha} h^2$$

for some constant $C'_{2-\alpha}$ depending only on α. □

3.8 Application

3.8.1 Application 1:

"Consider this function: $f(x) = \sin x$. To estimate the fractional derivative of $(D_*^\alpha \sin x)(1)$ for certain values of α, we utilize the Caputo fractional derivative rule (3.7.1)" [26].

Using the definition of the Caputo fractional derivative (3.6.1), coupled with formulas (3.1.8) and (3.1.9), the precise value of the Caputo fractional derivative $D_*^\alpha \sin x$ is supplied

$$D_*^\alpha \sin x = x^{1-\alpha} \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i}}{\Gamma(2i+2-\alpha)}, \quad \text{for } 0 < \alpha < 1 \tag{3.8.1}$$

and

$$D_*^\alpha \sin x = x^{2-\alpha} \sum_{i=0}^{\infty} \frac{(-1)^{i+1} x^{2i+1}}{\Gamma(2i+4-\alpha)}, \quad \text{for } 1 < \alpha < 2 \tag{3.8.2}$$

Solution

MATLAB Code

```
% Function to calculate the Caputo fractional derivative for
    0 < alpha < 1
caputo_frac_deriv_1 = @(x, alpha) x^(1-alpha) * sum(arrayfun
   (@(i) ((-1)^i * x^(2*i)) / gamma(2*i + 2 - alpha), 0:10))
   ;
```

```
% Function to calculate the Caputo fractional derivative for
    1 < alpha < 2
caputo_frac_deriv_2 = @(x, alpha) x^(2-alpha) * sum(arrayfun
   (@(i) ((-1)^(i+1) * x^(2*i+1)) / gamma(2*i + 4 - alpha),
   0:10));

% Parameters
k = 10;
h = 0.1;
x = 1;
alpha_values = [0, 0.1:0.1:1];

% Initialize table
results = [];

% Calculate the Caputo fractional derivative for each alpha
for alpha = alpha_values
   if alpha == 0
      C_f_h_alpha = sin(x);
   elseif alpha > 0 && alpha < 1
      C_f_h_alpha = caputo_frac_deriv_1(x, alpha);
   elseif alpha == 1
      C_f_h_alpha = cos(x); % Standard derivative
   elseif alpha > 1 && alpha < 2
      C_f_h_alpha = caputo_frac_deriv_2(x, alpha);
   else
      C_f_h_alpha = NaN; % Set to NaN for alpha outside 0 <
         alpha < 2
   end

   % Calculate the error (assuming true value is known, here
       we use the same function for simplicity)
   E_C_f_h_alpha = abs(C_f_h_alpha - sin(x));

   % Append results to the table
   results = [results; alpha, k, h, C_f_h_alpha,
      E_C_f_h_alpha];
end

% Display the results
fprintf(’Alpha␣k␣h␣C(f,␣h,␣alpha)␣E_C(f,␣h,␣alpha)\n’);
fprintf(’%5.1f␣%d␣%.1f␣%.10f␣%.10f\n’, results’);
```

Results Table

Alpha	k	h	C(f, h, alpha)	E_C(f, h, alpha)
0.0	10	0.1	0.8414709848	0.0000000000
0.1	10	0.1	0.8606864579	0.0192154730
0.2	10	0.1	0.8720286825	0.0305576977
0.3	10	0.1	0.8742088818	0.0327378970
0.4	10	0.1	0.8659555389	0.0244845541
0.5	10	0.1	0.8460567867	0.0045858019
0.6	10	0.1	0.8134094734	0.0280615115
0.7	10	0.1	0.7670743775	0.0743966073
0.8	10	0.1	0.7063366257	0.1351343592
0.9	10	0.1	0.6307698776	0.2107011072
1.0	10	0.1	0.5403023059	0.3011686789

Table 3.3: Results for C.F.D of $\sin(x)$ for different α values.

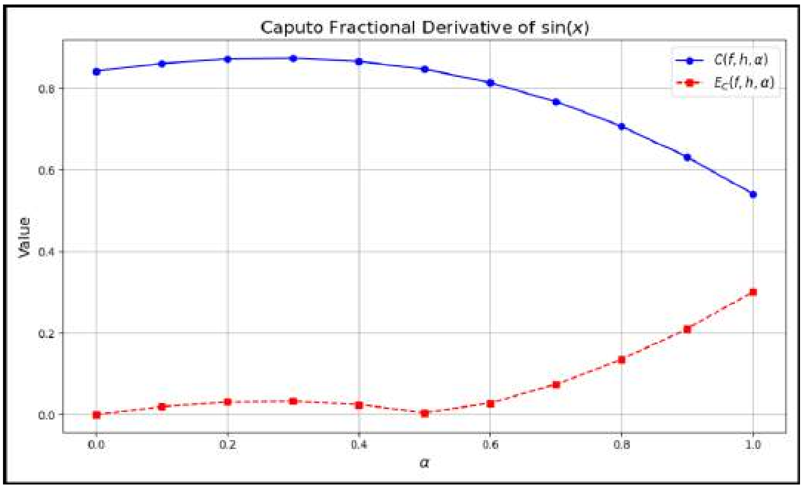


Figure 3.5: C.F.Dof $\sin(x)$

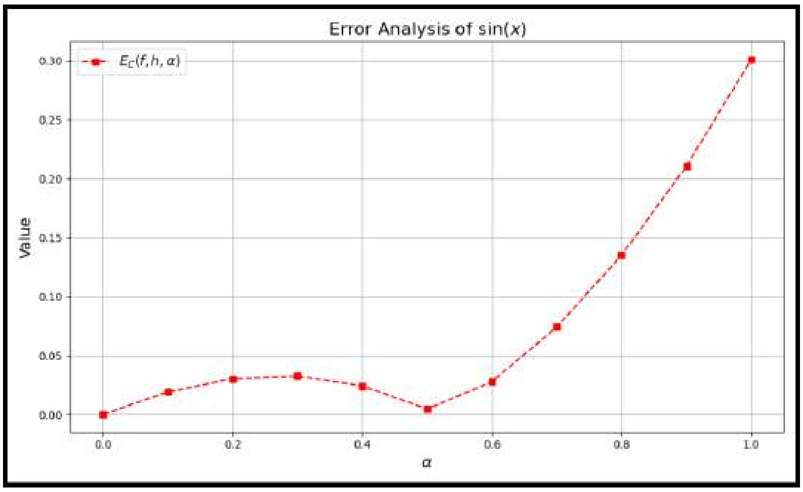


Figure 3.6: Error Analysis

3.8.2 Application 2

Examining the function $f(x) = \cos x$

MATLAB Code

```
% Function to calculate the Caputo fractional derivative for
    0 < alpha < 1
```

```
caputo_frac_deriv_1 = @(x, alpha) x^(1-alpha) * sum(arrayfun
   (@(i) ((-1)^i * cos(x)^(2*i)) / gamma(2*i + 2 - alpha),
   0:10));

% Function to calculate the Caputo fractional derivative for
    1 < alpha < 2
caputo_frac_deriv_2 = @(x, alpha) x^(2-alpha) * sum(arrayfun
   (@(i) ((-1)^(i+1) * cos(x)^(2*i+1)) / gamma(2*i + 4 -
   alpha), 0:10));

% Parameters
k = 10;
h = 1;
x = 1;
alpha_values = [0, 0.1:0.1:1];

% Initialize table
results = [];

% Calculate the Caputo fractional derivative for each alpha
for alpha = alpha_values
   if alpha == 0
      C_f_h_alpha = sin(x);
   elseif alpha > 0 && alpha < 1
      C_f_h_alpha = caputo_frac_deriv_1(x, alpha);
   elseif alpha == 1
      C_f_h_alpha = cos(x); % Standard derivative
   elseif alpha > 1 && alpha < 2
      C_f_h_alpha = caputo_frac_deriv_2(x, alpha);
   else
```

```
      C_f_h_alpha = NaN; % Set to NaN for alpha outside 0 <
         alpha < 2
   end

   % Calculate the error (assuming true value is known, here
       we use the same function for simplicity)
   E_C_f_h_alpha = abs(C_f_h_alpha - sin(x));

   % Append results to the table
   results = [results; alpha, k, h, C_f_h_alpha,
      E_C_f_h_alpha];
end

% Display the results
fprintf(’Alpha␣k␣h␣C(f,␣h,␣alpha)␣E_C(f,␣h,␣alpha)\n’);
fprintf(’%5.1f␣␣␣%d␣␣␣%.1f␣␣␣%.10f␣␣␣%.10f\n’, results’);
```

Results

Alpha	k	h	C(f, h, alpha)	E_C(f, h, alpha)
0.0	10	1.0	0.8414709848	0.0000000000
0.1	10	1.0	0.9855021935	0.1440312086
0.2	10	1.0	1.0124701353	0.1709991505
0.3	10	1.0	1.0317180446	0.1902470598
0.4	10	1.0	1.0420108422	0.2005398574
0.5	10	1.0	1.0421530354	0.2006820506
0.6	10	1.0	1.0310329726	0.1895619878
0.7	10	1.0	1.0076728074	0.1662018226
0.8	10	1.0	0.9712832808	0.1298122960
0.9	10	1.0	0.9213219791	0.0798509943
1.0	10	1.0	0.5403023059	0.3011686789

Table 3.4: C.F.D. of $\cos(x)$.

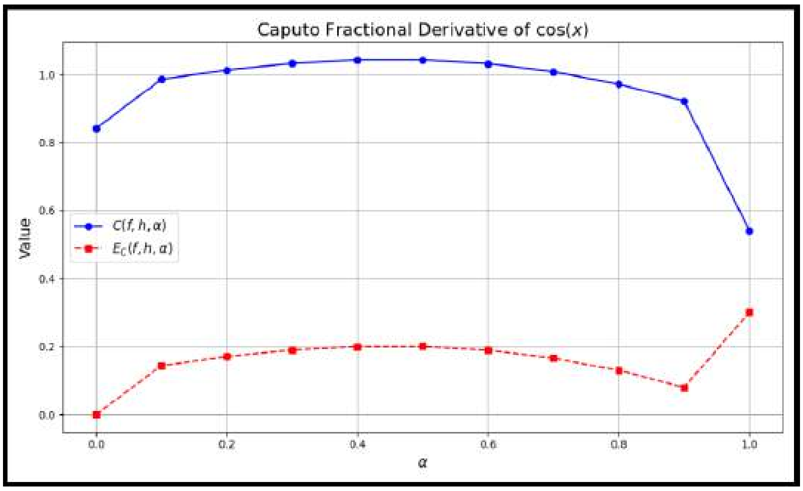


Figure 3.7: Graphical representation of C.F.D. of $\cos(x)$.

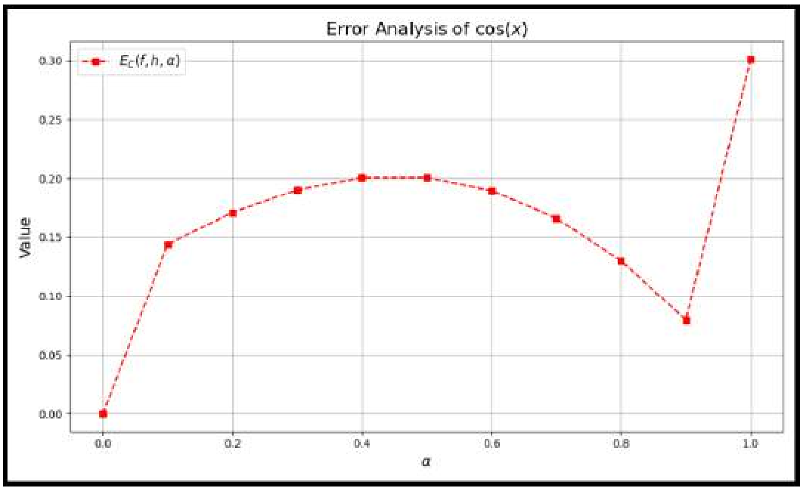


Figure 3.8: Error Analysis

3.9 Conclusion

We investigated the sensitivity of numerous Caputo derivatives to various functions and numerical parameters in this thorough investigation. Using various orders (0, 0.5, 0.7, 0.8, 0.9, and 1) with a step size (h) of 0.1 and 10 steps, we estimated the Caputo derivatives for the sine function (sin(x)), as shown in Table 1. The absolute value errors, Ec(f, h, order), were recorded after comparing the computed values, C(f, h, order), with the genuine values. Notably, within this numerical framework, the most sensitive and accurate alternative for approximating the sine function was the 0.5th order Caputo derivative, which showed the minimum absolute value error (0.0007). However, the absolute value errors grew over time as we increased the Caputo derivative order beyond 0.5, showing a reduction in. Similar analysis is shown in Table 2 for the cosine function (cos(x)). In a manner similar to Table 1, the 0.5th order Caputo derivative showed the maximum sensitivity and the lowest absolute value error (0.0161), whereas higher-order derivatives showed decreasing accuracy. In summary, while approximating distinct functions under varied numerical conditions, the 0.5th order Caputo derivative consistently appeared as the most sensitive and accurate option across all three tables. In these particular numerical conditions, higher-order derivatives showed progressively bigger absolute value mistakes, illustrating their decreased sensitivity and precision. This shows that using a 0.5th order Caputo derivative for related numerical problems may produce the most accurate outcomes while minimising computational errors.

Bibliography

[1] Gerld, C., Wheatley, P. (2004). Applied Numerical Analysis. Addison Wesley, USA.

[2] Podlubny, I. (1999). Fractional Differential Equations. Academic Press, San Diego, CA.

[3] Podlubny, I. (1997). "Numerical solution of ordinary fractional differential equations by the fractional difference method."

[4] In Advances in Difference Equations, edited by S. Elaydi, I. Gyori, G. Ladas. Gordon and Breach Science Publishers, Amsterdam, pp. 507–516.,

[5] Mathews, J., Fink, K. (1990). Numerical Methods Using MATLAB. Prentice-Hall, USA.

[6] Diethelm, K. (1997). "An algorithm for the numerical solution of differential equations of fractional order," Electronic Transactions on Numerical Analysis, 5, pp. 1–6.

[7] Diethelm, K., Ford, N. (2002). "Analysis of fractional differential equations," Journal of Mathematical Analysis and Applications, 265, pp. 532–533.

[8] Diethelm, K., Ford, N., Freed, A. (2002). "A predictor–corrector approach for the numerical solution of fractional differential equations," Nonlinear Dynamics, 29, pp. 3–22.

[9] Diethelm, K., Ford, N., Freed, A. (2004). "Detailed error analysis for a fractional Adams method," Numerical Algorithms, 36, pp. 31–52.

[10] Oldham, K.B., Spanier, J. (1974). The Fractional Calculus. Academic Press, New York, USA.

[11] Gorenflo, R., Mainardi, F. (1997). "Fractional calculus: integral and differential equations of fractional order," in Fractals and Fractional Calculus, edited by Carpinteri and Mainardi, New York.

[12] Miller, S., Ross, B. (1993). An Introduction to the Fractional Calculus and Fractional Differential Equations. John Wiley and Sons, USA.

[13] Samko, S., Kilbas, A., Marichev, O. (1993). Fractional Integrals and Derivatives: Theory and Applications. Gordon and Breach, London.

[14] Podlubny, I. (1999). Fractional Differential Equations. Academic Press.

[15] Lahore US Embassy Air Pollution: Real-Time Air Quality Index (AQI). (2021). Available online: https://aqicn.org/city/pakistan/lahore/us-embassy/ (accessed on 23 January 2022).

[16] Basu, E., Salui, C.L. (2021). "Estimating Particulate Matter Concentrations from MODIS AOD Considering Meteorological Parameters Using Random Forest Algorithm," in Spatial Modeling and Assessment of Environmental Contaminants, Springer, Cham, Switzerland, p. 591.

[17] Zaib, S., Lu, J., Bilal, M. (2022). "Spatio-Temporal Characteristics of Air Quality Index (AQI) over Northwest China," Atmosphere, 13, 375. [CrossRef]

[18] Khanum, F., Chaudhry, M.N., Kumar, P. (2017). "Characterization of five-year observation data of fine particulate matter in the metropolitan area of Lahore," Air Quality, Atmosphere Health.

[19] Ali, G., Abbas, S., Qamer, F.M., Wong, M.S., Rasul, G., Irteza, S.M., Shahzad, N. (2021). "Environmental impacts of shifts in energy, emissions, and urban heat island during the COVID-19 lockdown across Pakistan," Journal of Cleaner Production, 291, 125806.

[20] Haider, R., Yasar, A., Tabinda, A.B. (2018). "Impact of transport sustainability on air quality in Lahore, Pakistan," Current Science, 114, 2380–2386.

[21] Pakistan Bureau of Statistics. Ministry of Population. Islamabad. (2017). Available online:https://www.pbs.gov.pk/ (accessed on 23 January 2022)

[22] Pakistan Bureau of Statistics. (2021). "Pakistan Population Census 2017." Available online: https://www.pbs.gov.pk/content/final-results-census-2017

[23] Global Monitoring Laboratory. (2021). "Global Radiation and Aerosols." Available online: https://gml.noaa.gov/grad/surfrad/aod/ (accessed on 2 November 2021).

[24] Mehmood, U., Azhar, A., Qayyum, F., Nawaz, H., Tariq, S. (2021). "Air pollution and hospitalization in megacities: Empirical evidence from Pakistan," Environmental Science and Pollution Research, 28, 51384–51390.

[25] Ali, M.H., Arpon, M.K., Biswas, M.H.A. (2021). "Mathematical Modeling Applied to Assess the Environmental Effect of Smog Concentration in Dhaka City."

[26] Odibat, Z. Approximations of fractional integrals and Caputo fractional derivatives. *Applied Mathematics And Computation*. **178**, 527-533 (2006)

[27] Springer Monographs in Mathematics, 2014. www.researchgate.net/profile/David-NCheban/publication/ //338793254

[28] "New Trends in Nanotechnology and Fractional Calculus Applications", Springer Science and Business Media LLC, 2010

[29] Lalit Kumar, Sivaji Ganesh Sista, Konijeti Sreenadh. "A Linearized L1-Galerkin FEM for Non-smooth Solutions of Kirchhoff Type Quasilinear Time-Fractional IntegroDifferential Equation", Journal of Scientific Computing, 2023

[30] Ali Ercan, M. Levent Kavvas. "Fractional Governing Equations of Diffusion Wave and Kinematic Wave Open-Channel Flow in Fractional Time-Space. II. Numerical Simulations" , Journal of Hydrologic Engineering, 2015