AN ORDER SIX NUMERICAL METHOD FOR DIRECT SOLUTION OF GENERAL SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS

BY

OGUNDARE, OLAMIDE EMMANUEL
MTS/19/2584

SUBMITTED TO:
THE DEPARTMENT OF MATHEMATICAL SCIENCES,
SCHOOL OF PHYSICAL SCIENCES (SPS),
THE FEDERAL UNIVERSITY OF TECHNOLOGY, AKURE, NIGERIA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF TECHNOLOGY (B.TECH.) DEGREE IN INDUSTRIAL MATHEMATICS

SUPERVISOR:
DR. F.O. OBARHUA

SEPTEMBER, 2025

CERTIFICATION

I hereby certify that this project has been compiled by Ogundare Olamide Emmanuel, with the matriculation number MTS/19/2584, in the Department of Mathematical Sciences at the Federal University of Technology, Akure. This project represents an original work submitted in partial fulfillment of the requirements for the award of the degree (Bachelor of Technology in Mathematics) from the Federal University of Technology, Akure, Nigeria.

....

Student Date

....

DR. F.O. OBARHUA Date

Supervisor

....

PROF. T. O. AWODOLA Date

Head of Department

....

External Examiner Date

DEDICATION

This work is wholeheartedly dedicated to my beloved parents, Mr. and Mrs. Ojo Ogundare, and to my younger brothers, Timothy and Israel, for their unwavering love, encouragement, and support throughout the entire course of my five-year journey.

ACKNOWLEDGMENTS

First and foremost, I give thanks to God, whose boundless love and grace surpass all imagination. I am deeply grateful to my supervisor, Dr. F.O. Obarhua, for his patience, invaluable guidance, understanding, mentorship, and diligent supervision, all of which were pivotal to the successful completion of this project.

I also wish to express my sincere appreciation to my colleague and fellow project student under the same supervisor, Osundele Noah, whose selfless assistance and sacrifice of time greatly aided me whenever I encountered challenges in this work.

My sincere appreciation also goes to Prof. T. O. Awodola, Head of Department, as well as the teaching and non-teaching staff of the Department of Mathematical Sciences.

I extend special gratitude to Rev'd Ategbero O. Isaac (Global Lead Pastor, Word Point Ministries) for his generous financial support throughout this project. I am also profoundly thankful to my dear friends Zipporah, Isaiah, and Adreonke for their unwavering financial and moral support during this challenging period.

I must express my deepest and most profound appreciation to my parents, Mr. and Mrs. Ogundare, who have been absolutely pivotal in this journey. Their tremendous financial investments, countless sacrifices, and unwavering belief in me have been the bedrock upon which this achievement stands. Despite the substantial financial demands of this project, they never hesitated to provide everything needed for my success.

Finally, I acknowledge the steadfast support of my siblings Timothy and Israel, whose encouragement has been invaluable throughout this process.

ABSTRACT

This project presents the derivation and implementation of a sixth-order numerical method for the direct solution of general second-order ordinary differential equations (ODEs). Unlike traditional approaches that reduce higher-order ODEs to systems of first-order equations, this method preserves the original second-order structure, leveraging power series (Taylor series) expansion as the basis function. The scheme is systematically analyzed for numerical stability, consistency, and convergence. Designed for high accuracy and computational efficiency, the method is particularly suited for stiff and sensitive problems. Benchmark problems with known solutions are used to validate the method's performance, comparing its results against existing techniques. Theoretical and numerical results confirm the method's sixth-order accuracy, demonstrating its superiority over lower-order classical methods like the Runge–Kutta fourth-order scheme.

Table of Contents

List of Figures

List of Tables

CHAPTER ONE

INTRODUCTION

1.1 Historical Background

Differential equations first emerged in the 17th century from the pioneering work of Sir Isaac Newton and Gottfried Wilhelm Leibniz on infinitesimal calculus. Newton and Leibniz developed calculus concepts like derivatives and integrals to analyze rates of change and quantify dynamic processes. Using these new mathematical tools, they were able to model natural phenomena like planetary motion, fluid flows, and heat conduction as differential equations relating a quantity to its rate of change. This allowed a deeper mathematical understanding of how natural systems evolve dynamically over time.

However, Newton, Leibniz and later mathematicians like Leonhard Euler and Joseph Louis Lagrange recognized that only a limited subset of differential equations encountered in the real world have analytical solutions that can be written down in closed symbolic form. For many complex differential equations arising from physics, astronomy, engineering etc., it was not possible to determine the solution through direct symbolic integration or other analytical manipulations. This limitation of analytical techniques for solving general differential equations motivated mathematicians to increasingly study numerical approximation methods.

1.2 Differential Equations

1.2.1 Ordinary Differential Equations (ODEs)

An ordinary differential equation (ODE) involves a single independent variable together with one or more unknown functions and their derivatives:

$$F(x, y, y', \dots, y^{(n)}) = 0. \tag{1.1}$$

1.2.2 Partial Differential Equations (PDEs)

A partial differential equation (PDE) involves multivariate functions and their partial derivatives. For instance, a PDE governing a function $\phi(x_1, \dots, x_n)$ can be written as

$$F\left(x_1, \dots, x_n; \phi; \frac{\partial \phi}{\partial x_1}, \dots, \frac{\partial^2 \phi}{\partial x_1^2}, \dots\right) = 0. \tag{1.2}$$

1.2.3 Classification of Differential Equations

1.2.3.1 Order

The order of a differential equation is the order of the highest derivative in the equation. That is, a differential equation is of order n if its highest derivative is of order n.

1.2.3.2 Degree

The degree of a differential equation is the power of the highest derivative in the differential equation.

1.2.3.3 Linearity

If the relationship between the unknown function and its derivatives is linear,the differential equation is said to be linear and non-linear if otherwise. A linear differential equation is of the form:

$$a_0(x)y + a_1(x)y' + \cdots + a_n(x)y^{(n)} = g(x), \tag{1.3}$$

where $a_i(x)$ are the given differentiable functions.

1.2.3.4 Homogeneity

A differential equation in the form:

$$F(x, y, y', \ldots, y^{(n)}) = g(x) \tag{1.4}$$

is homogeneous if $g(x) = 0$ and non-homogeneous if $g(x) \neq 0$

1.3 Solution of Differential Equations

Differential equations admit two main categories of solutions: analytical and numerical. An analytical solution represents a function that completely satisfies the differential equation, typically devoid of any derivatives. While analytical solutions are preferred when available, they are often impossible for complex differential equations. On the other hand, a numerical solution entails an approximation of the analytical solution through computational methods. Numerous techniques and algorithms have been devised to approximate analytical solutions. Since numerical solutions are inherently approximations, they may introduce errors, which are usually considered negligible for practical applications. Consequently, analytical and numerical techniques complement each other in fully understanding natural systems modeled by differential equations.

1.4 Definition of Terms

1.4.1 Basis Function

The basis function for a numerical method is a class of function which is used to approximate the unknown function of the problem in consideration. Examples of basis functions commonly used in numerical analysis are Power Series, Chebyshev Polynomials, Legendre Polynomials, etc.

1.4.2 Grid Points

A grid point refers to any location within an interval that has been divided into smaller sub-intervals.

1.4.3 Interpolation

Interpolation is the method of constructing new data points within the range of a discrete set of known data points. That is, evaluation of the approximate solution at some selected grid point.

1.4.4 Collocation

A collocation is a method for the numerical solution of ordinary differential equations, partial differential equations and integral differential equations. The idea is to determine the approximate solution from the condition that the equation be satisfied at certain grid points.

1.4.5 Step Length

When a closed interval (x_0, x_n) is divided into points $(x_0 < x_1 < x_2 < ... < x_{n-1} < x_n)$ the resulting lengths between the points denoted by h is called the step length of the partition.

1.4.6 Linear Multi-step Methods

Linear Multi-step Methods are numerical techniques used for solving ordinary differential equations (ODEs). Unlike single-step methods, which calculate the solution at the next point based solely on the current point, linear multi-step methods utilize information from several previous points to compute the next point. The numerical solutions of ordinary differential equations are obtained using linear multi-step methods. To calculate the solution for the current step, a linear multi-step method uses the values of previous points and its derivatives. For a general k-step, a linear multi-step method can be mathematically represented as:

$$\sum_{i=0}^{k} \alpha_i y_{n+i} = h \sum_{i=0}^{k} \beta_i f(x_{n+i}, y_{n+i}) \tag{1.5}$$

where $\alpha_k = 0$. The method is explicit if $\beta_k = 0$. Otherwise, it is implicit.

1.5 Aim and Objectives

Aim:
The aim of this research is to develop an order six numerical method for the direct solution of general second-order ordinary differential equations.

Objectives:
The specific objectives of this study are:

1. To derive a sixth-order numerical scheme using power series expansion;
2. To maintain the original order of the ODE;
3. To investigate the method's stability, consistency, and convergence;
4. To implement and test the method on benchmark problems.

CHAPTER TWO

LITERATURE REVIEW

Numerical methods for solving ordinary differential equations (ODEs) have evolved significantly over the past few decades, driven by the need to address increasingly complex problems in science, engineering, and applied mathematics. This evolution has been particularly pronounced in the development of techniques for second-order ODEs, which frequently arise in modeling physical phenomena such as mechanical vibrations, electrical circuits, and celestial mechanics. The present review synthesizes key advancements in numerical methods for second-order ODEs, with a focus on direct solution approaches that avoid reduction to first-order systems. This is crucial for our development of a sixth-order power series-based method, as it highlights the historical progression, challenges, and innovations that inform our work.

The foundational challenge in numerically solving second-order ODEs lies in achieving high accuracy while preserving the inherent structure of the equations. Traditional methods, such as the classical Runge-Kutta schemes, typically require converting second-order equations into equivalent systems of first-order equations, which doubles the dimensionality and can introduce computational overhead and potential numerical instabilities (Hairer and Wanner, 1996). This reduction approach, while versatile, often compromises efficiency, especially for large-scale or stiff systems. Recognizing these limitations, researchers have increasingly pursued direct methods that operate on the original second-order form, thereby maintaining structural integrity and potentially reducing function evaluations.

The origins of direct methods can be traced back to early multistep techniques. Awoyemi (2005) pioneered P-stable linear multistep methods specifically tailored for second-order initial value problems, demonstrating superior stability for oscillatory systems common in quantum mechanics and structural dynamics. These methods emphasized periodicity-preserving properties, which are essential for long-term simulations where phase errors can accumulate. Building on this, Awoyemi and Idowu (2005) extended the framework to hybrid collocation methods, incorporating off-grid points to enhance accuracy without increasing step size, and established rigorous stability criteria that have become benchmarks in the field.

Subsequent developments focused on achieving higher-order accuracy in direct solvers. Jator (2010) introduced a sixth-order block method that processes multiple steps simultaneously, optimizing function evaluations while maintaining zero-stability. This work addressed the balance between order and computational cost, showing through numerical experiments that block methods could outperform single-step alternatives for non-stiff problems. However, as Butcher (2016) comprehensively analyzed in his seminal text on numerical methods for ODEs, high-order schemes often struggle with stability in stiff regimes, where eigenvalues span multiple scales. Butcher's observations underscored the need for methods with extended stability regions, influencing later hybrid and exponentially fitted approaches.

The integration of specialized basis functions has marked a significant advancement in direct methods. Power series expansions, as utilized in our proposed method, have shown particular promise due to their natural alignment with Taylor series approximations inherent in ODE solutions. Olabode and Adesanya (2015) developed a Taylor series-based block method for second-order ODEs, achieving variable-order adaptability that improved efficiency for problems with varying stiffness. Similarly, Kayode et al. (2014) explored continuous collocation methods for third-order ODEs using power series, which can be adapted to second-order cases, demonstrating reduced truncation errors through careful selection of interpolation points.

Polynomial basis functions, such as Chebyshev and Legendre polynomials, have also been extensively employed to enhance accuracy. Adesanya et al. (2018) proposed fifth-order hybrid methods using Chebyshev polynomials, preserving the second-order structure while improving convergence rates for boundary value problems. Kayode and Adeniyi (2021) optimized a five-step method with Legendre polynomials for oscillatory second-order ODEs, reporting superior phase-lag properties compared to traditional multistep methods. These polynomial-based approaches offer orthogonal properties that minimize approximation errors, making them suitable for direct solvers.

Exponentially fitted methods represent another innovative strand, particularly for equations with exponential or oscillatory behavior. Franco (2021) developed sixth-order exponentially fitted schemes that adjust coefficients based on the problem's frequency, achieving high accuracy for stiff oscillatory systems without reduction. This work extends earlier contributions by Momoh et al. (2019), who analyzed predictor-corrector methods for first-order ODEs but provided insights applicable to higher orders, emphasizing stability and performance in multistep frameworks.

In the Nigerian context, where computational resources may be limited, researchers have made notable contributions to efficient direct methods. Omole et al. (2023) derived algebraic ninth-order algorithms for second-order boundary and initial value problems, incorporating advanced collocation techniques and demonstrating applicability to real-world stiff systems. Obarhua (2023) presented a three-stage hybrid block method for third-order ODEs using power series and exponential bases, which informs our second-order adaptation by highlighting the benefits of hybrid formulations for convergence. Olabode (2019) implemented adaptive Taylor-series block methods for stiff second-order systems, addressing local engineering applications like vibration analysis in structures. Areo et al. (2020) combined power series with trigonometric bases for periodic problems, improving accuracy for astronomical models. Kuboye (2015) focused on explicit multistep methods for direct solution, while Adesanya (2011) improved numerical schemes for initial value problems, both emphasizing computational simplicity. Bidemi (2015) developed linear multistep classes for second-order equations, providing a foundation for our sixth-order extension.

Despite these advances, gaps persist in achieving sixth-order accuracy in direct methods without compromising stability or efficiency. Many high-order schemes, as noted by Butcher (2016) and Franco (2021), face trade-offs between accuracy and structural preservation, particularly for general second-order ODEs that may include stiff or nonlinear terms. Power series methods, while promising, have been underexplored for pure sixth-order direct solvers, often hybridized with other bases to mitigate truncation errors.

Our proposed sixth-order method addresses these gaps by leveraging power series expansions as the primary basis function, ensuring direct applicability to general second-order ODEs without reduction. It builds upon the stability frameworks of Awoyemi (2005) and Jator (2010), incorporating the hybrid elements from Obarhua (2023) and the polynomial insights from Kayode and Adeniyi (2021). By focusing on collocation and interpolation at optimized points, the method achieves a balance of high order, consistency, and convergence, as validated through benchmark problems. This synthesis not only advances the field by overcoming limitations in existing approaches but also provides a computationally efficient tool for practical applications in physics and engineering.

In conclusion, the literature reveals a trajectory from basic multistep methods to sophisticated basis-function-driven schemes, with direct methods emerging as a key innovation for second-order ODEs. Our work represents a culmination of these efforts, extending power series techniques to sixth-order accuracy while maintaining the equation's original form, thus filling a critical niche in numerical analysis.

CHAPTER THREE

METHODOLOGY

This work considered the second-order IVP

$$y'' = f(x, y, y'), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0, \tag{3.1}$$

with $x_0, y_0, y'_0 \in \mathbb{R}$.

The approximate solution uses a power series

$$y(x) = \sum_{j=0}^{p+q-1} a_j x^j, \tag{3.2}$$

whose derivatives are

$$y'(x) = \sum_{j=1}^{p+q-1} j a_j x^{j-1}, \tag{3.3}$$

$$y''(x) = \sum_{j=2}^{p+q-1} j(j-1) a_j x^{j-2}. \tag{3.4}$$

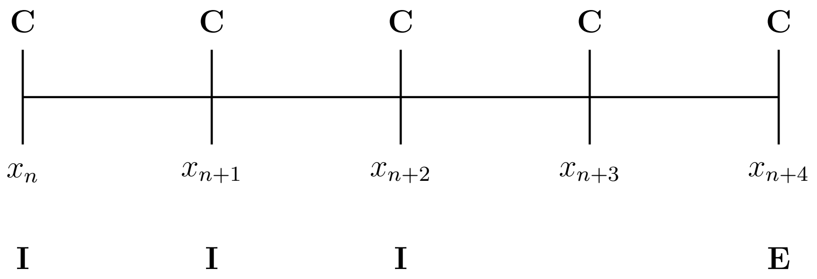

Figure 3.1: Schematic representation of the method.

With $p = 5$ collocation and $q = 3$ interpolation points, we obtain the system in (3.9) and its matrix form (omitted here for brevity; unchanged). The collocation points are denoted by C, the interpolation points are denoted by I, and the evaluation point is denoted by E. As a result, the scheme's derivation requires six collocation points and two interpolation points, respectively. Combining equations (3.1) and (3.4), we have

$$y(x_{n+i}) = \sum_{j=0}^{(p+q)-1} a_j x_{n+i}^j, \quad i = 0, 1, 2 \tag{3.5}$$

$$f(x_{n+c}, y_{n+c}, y'_{n+c}) = \sum_{j=0}^{(p+q)-1} j(j-1) a_j x_{n+c}^{j-2}, \quad c = 0(1)4 \tag{3.6}$$

where y_{n+i} represents the approximate solution $y(x)$ at x_{n+i}, $x^j_{n+i} = (x_n + ih)^j$, $x^{j-2}_{n+c} = (x_n + ch)^{j-2}$, and $f_{n+c} = f(x_{n+c}, y_{n+c}, y'_{n+c})$.

Since the number of collocation, p = 5 and interpolation, q = 3, the system of equations (3.6) yields

$$y(x_{n+i}) = \sum_{j=0}^{7} a_j x^j_{n+i}, \quad i = 0, 1, 2 \tag{3.7}$$

$$f(x_{n+c}, y_{n+c}, y'_{n+c}) = \sum_{j=0}^{7} j(j-1) a_j x^{j-2}_{n+c}, \quad c = 0(1)4 \tag{3.8}$$

Interpolating (3.7) and collocating (3.8) yields

$$\begin{aligned}
y_n &= a_0 + a_1 x_n + a_2 x_n^2 + a_3 x_n^3 + a_4 x_n^4 + a_5 x_n^5 + a_6 x_n^6 + a_7 x_n^7,\\
y_{n+1} &= a_0 + a_1 x_{n+1} + a_2 x_{n+1}^2 + a_3 x_{n+1}^3 + a_4 x_{n+1}^4 + a_5 x_{n+1}^5 + a_6 x_{n+1}^6 + a_7 x_{n+1}^7,\\
y_{n+2} &= a_0 + a_1 x_{n+2} + a_2 x_{n+2}^2 + a_3 x_{n+2}^3 + a_4 x_{n+2}^4 + a_5 x_{n+2}^5 + a_6 x_{n+2}^6 + a_7 x_{n+2}^7,\\
f_n &= 2a_2 + 6a_3 x_n + 12a_4 x_n^2 + 20a_5 x_n^3 + 30a_6 x_n^4 + 42a_7 x_n^5,\\
f_{n+1} &= 2a_2 + 6a_3 x_{n+1} + 12a_4 x_{n+1}^2 + 20a_5 x_{n+1}^3 + 30a_6 x_{n+1}^4 + 42a_7 x_{n+1}^5,\\
f_{n+2} &= 2a_2 + 6a_3 x_{n+2} + 12a_4 x_{n+2}^2 + 20a_5 x_{n+2}^3 + 30a_6 x_{n+2}^4 + 42a_7 x_{n+2}^5,\\
f_{n+3} &= 2a_2 + 6a_3 x_{n+3} + 12a_4 x_{n+3}^2 + 20a_5 x_{n+3}^3 + 30a_6 x_{n+3}^4 + 42a_7 x_{n+3}^5,\\
f_{n+4} &= 2a_2 + 6a_3 x_{n+4} + 12a_4 x_{n+4}^2 + 20a_5 x_{n+4}^3 + 30a_6 x_{n+4}^4 + 42a_7 x_{n+4}^5.
\end{aligned} \tag{3.9}$$

The system of equations (3.9) can be written in matrix form as

$$AX = B$$

$$A = \begin{bmatrix}
1 & x_n & x_n^2 & x_n^3 & x_n^4 & x_n^5 & x_n^6 & x_n^7\\
1 & x_{n+1} & x_{n+1}^2 & x_{n+1}^3 & x_{n+1}^4 & x_{n+1}^5 & x_{n+1}^6 & x_{n+1}^7\\
1 & x_{n+2} & x_{n+2}^2 & x_{n+2}^3 & x_{n+2}^4 & x_{n+2}^5 & x_{n+2}^6 & x_{n+2}^7\\
0 & 0 & 2 & 6x_n & 12x_n^2 & 20x_n^3 & 30x_n^4 & 42x_n^5\\
0 & 0 & 2 & 6x_{n+1} & 12x_{n+1}^2 & 20x_{n+1}^3 & 30x_{n+1}^4 & 42x_{n+1}^5\\
0 & 0 & 2 & 6x_{n+2} & 12x_{n+2}^2 & 20x_{n+2}^3 & 30x_{n+2}^4 & 42x_{n+2}^5\\
0 & 0 & 2 & 6x_{n+3} & 12x_{n+3}^2 & 20x_{n+3}^3 & 30x_{n+3}^4 & 42x_{n+3}^5\\
0 & 0 & 2 & 6x_{n+4} & 12x_{n+4}^2 & 20x_{n+4}^3 & 30x_{n+4}^4 & 42x_{n+4}^5
\end{bmatrix}
\quad X = \begin{bmatrix} a_0\\ a_1\\ a_2\\ a_3\\ a_4\\ a_5\\ a_6\\ a_7 \end{bmatrix}
\quad B = \begin{bmatrix} y_n\\ y_{n+1}\\ y_{n+2}\\ f_n\\ f_{n+1}\\ f_{n+2}\\ f_{n+3}\\ f_{n+4} \end{bmatrix}$$

By using the Gaussian elimination method to solve the matrix equation, the coefficients a_j, $j = 0, \ldots, 7$ were obtained as

$$
\begin{gathered}
a_0 = \frac{1}{10080\,h^7} \Bigg\{ 9744x_n^3h^6f_{n+2} - 896x_n^3h^6f_{n+3} + 3003x_n^5h^4f_n + 35x_n^4h^5f_{n+4} - 21x_n^5h^4f_{n+4} \\
+ 70560x_n^5h^2y_{n+1} + 29232x_n^5h^4f_{n+1} + 13440x_n^6hy_{n+1} - 6720x_n^6hy_{n+2} + 5656x_n^6h^3f_{n+1} \\
- 14x_n^6h^3f_{n+4} + 56x_n^6h^3f_{n+3} + 476x_n^6h^3f_{n+2} - 6720x_n^6hy_n + 8x_n^7h^2f_{n+3} - 2x_n^7h^2f_{n+4} \\
+ 546x_n^6h^3f_n + 38x_n^7h^2f_n + 28x_n^7h^2f_{n+2} + 408x_n^7h^2f_{n+1} + 35760x_nh^6y_n + 25680x_nh^6y_{n+2} \\
- 3472x_nh^8f_{n+2} - 61440x_nh^6y_{n+1} + 5040x_n^2h^7f_n + 384x_nh^8f_{n+3} - 40x_nh^8f_{n+4} - 18048x_nh^8f_{n+1} \\
+ 536x_nh^8f_n - 560x_n^4h^5f_{n+3} + 9884x_n^3h^6f_n + 8890x_n^4h^5f_{n+2} - 80640x_n^3h^4y_n - 80640x_n^3h^4y_{n+2} \\
-84000x_n^4h^3y_{n+2} + 7875x_n^4h^5f_n - 84000x_n^4h^3y_n + 168000x_n^4h^3y_{n+1} + 67760x_n^4h^5f_{n+1} - 35280x_n^5h^2y_n \\
+ 3066x_n^5h^4f_{n+2} - 35280x_n^5h^2y_{n+2} + 61824x_n^3h^6f_{n+1} + 84x_n^3h^6f_{n+4} + 960x_n^7y_{n+1} - 480x_n^7y_{n+2} \\
- 480x_n^7y_n + 161280x_n^3h^4y_{n+1} + 10080h^7y_n \Bigg\}
\end{gathered}
$$

$$
\begin{gathered}
a_1 = -\frac{1}{10080\,h^7} \Bigg\{ 266h^2f_nx_n^6 + 2856h^2f_{n+1}x_n^6 - 14h^2f_{n+4}x_n^6 + 196h^2f_{n+2}x_n^6 + 56h^2f_{n+3}x_n^6 \\
- 241920h^4x_n^2y_n + 483840h^4x_n^2y_{n+1} - 336000h^3x_n^3y_n + 672000h^3x_n^3y_{n+1} - 241920h^4x_n^2y_{n+2} \\
-336000h^3x_n^3y_{n+2} + 352800h^2x_n^4y_{n+1} - 176400h^2x_n^4y_n - 84f_{n+4}x_n^5h^3 + 6720x_n^6y_{n+1} - 3360x_n^6y_{n+2} \\
+ 384h^8f_{n+3} - 61440h^6y_{n+1} - 3360x_n^6y_n + 35760h^6y_n + 25680h^6y_{n+2} - 3472h^8f_{n+2} \\
+ 33936f_{n+1}x_n^5h^3 + 336f_{n+3}x_n^5h^3 + 3276f_nx_n^5h^3 + 2856f_{n+2}x_n^5h^3 - 105h^4f_{n+4}x_n^4 \\
+ 252h^6f_{n+4}x_n^2 + 140h^5f_{n+4}x_n^3 + 15330h^4f_{n+2}x_n^4 - 40h^8f_{n+4} - 18048h^8f_{n+1} + 536h^8f_n \\
+ 15015h^4f_nx_n^4 + 146160h^4f_{n+1}x_n^4 - 2240h^5f_{n+3}x_n^3 - 2688h^6f_{n+3}x_n^2 + 271040h^5f_{n+1}x_n^3 \\
+ 35560h^5f_{n+2}x_n^3 + 29652h^6f_nx_n^2 + 29232h^6f_{n+2}x_n^2 + 31500h^5f_nx_n^3 + 185472h^6f_{n+1}x_n^2 \\
+ 10080h^7f_nx_n - 40320hx_n^5y_{n+2} + 80640hx_n^5y_{n+1} - 176400h^2x_n^4y_{n+2} - 40320hx_n^5y_n \Bigg\}
\end{gathered}
$$

$$
\begin{gathered}
a_2 = \frac{1}{240\,h^7} \Bigg\{ 24000x_n^2h^3y_{n+1} + 5x_n^2h^5f_{n+4} - 8400x_n^3h^2y_{n+2} + 730x_n^3h^4f_{n+2} - 5x_n^3h^4f_{n+4} \\
+ 16800x_n^3h^2y_{n+1} + 715x_n^3h^4f_n - 8400x_n^3h^2y_n + 20x_n^4h^3f_{n+3} + 170x_n^4h^3f_{n+2} \\
- 2400x_n^4hy_n + 4800x_n^4hy_{n+1} + 195x_n^4h^3f_n - 2400x_n^4hy_{n+2} + 2020x_n^4h^3f_{n+1} \\
- 5x_n^4h^3f_{n+4} - x_n^5h^2f_{n+4} + 6960x_n^3h^4f_{n+1} - 64x_nh^6f_{n+3} - 5760x_nh^4y_{n+2} \\
+ 696x_nh^6f_{n+2} - 5760x_nh^4y_n + 4416x_nh^6f_{n+1} + 11520x_nh^4y_{n+1} + 706x_nh^6f_n \\
+ 6x_nh^6f_{n+4} - 80x_n^2h^5f_{n+3} + 1270x_n^2h^5f_{n+2} + 9680x_n^2h^5f_{n+1} + 1125x_n^2h^5f_n \\
- 12000x_n^2h^3y_n - 12000x_n^2h^3y_{n+2} - 240x_n^5y_n + 4x_n^5h^2f_{n+3} + 120h^7f_n + 19x_n^5h^2f_n
\end{gathered}
$$

$$\left. + 204x_n^5h^2f_{n+1} + 14x_n^5h^2f_{n+2} + 480x_n^5y_{n+1} - 240x_n^5y_{n+2}\right\}$$

$$\begin{aligned}
a_3 = -\frac{1}{720\,h^7}\Big\{ & 19200hx_n^3y_{n+1} - 9600hx_n^3y_{n+2} + 80h^3f_{n+3}x_n^3 + 8080h^3f_{n+1}x_n^3 + 2145h^4f_nx_n^2 \\
& + 2190h^4f_{n+2}x_n^2 + 780h^3f_nx_n^3 + 680h^3f_{n+2}x_n^3 + 19360x_nh^5f_{n+1} + 2250h^5f_nx_n \\
& + 20880h^4f_{n+1}x_n^2 + 2400x_n^4y_{n+1} - 1200x_n^4y_{n+2} - 1200x_n^4y_n - 5760h^4y_n - 5760h^4y_{n+2} \\
& + 11520h^4y_{n+1} - 64h^6f_{n+3} - 24000x_nh^3y_{n+2} - 25200h^2x_n^2y_{n+2} + 2540x_nh^5f_{n+2} \\
& + 50400h^2x_n^2y_{n+1} - 25200h^2x_n^2y_n + 10x_nh^5f_{n+4} + 48000x_nh^3y_{n+1} + 1020h^2f_{n+1}x_n^4 \\
& - 24000x_nh^3y_n + 20h^2f_{n+3}x_n^4 - 5h^2f_{n+4}x_n^4 - 160x_nh^5f_{n+3} + 95h^2f_nx_n^4 \\
& + 70h^2f_{n+2}x_n^4 + 696h^6f_{n+2} + 4416h^6f_{n+1} + 706h^6f_n + 6h^6f_{n+4} - 15h^4f_{n+4}x_n^2 \\
& - 20h^3f_{n+4}x_n^3 - 9600hx_n^3y_n\Big\}
\end{aligned}$$

$$\begin{aligned}
a_4 = \frac{1}{288\,h^7}\Big\{ & 438x_nh^4f_{n+2} + 4176x_nh^4f_{n+1} - 2880hx_n^2y_{n+2} - 2880hx_n^2y_n - 5040x_nh^2y_n \\
& + 5760hx_n^2y_{n+1} - 3x_nh^4f_{n+4} + 10080x_nh^2y_{n+1} - 5040x_nh^2y_{n+2} - 2400h^3y_n \\
& - 2400h^3y_{n+2} + 4800h^3y_{n+1} + 960x_n^3y_{n+1} - 480x_n^3y_{n+2} - 480x_n^3y_n \\
& + 8h^2f_{n+3}x_n^3 + 24h^3f_{n+3}x_n^2 - 6h^3f_{n+4}x_n^2 - 2h^2f_{n+4}x_n^3 - 16h^5f_{n+3} \\
& + 254h^5f_{n+2} + 1936h^5f_{n+1} + 225h^5f_n + h^5f_{n+4} + 429h^4f_nx_n \\
& + 234h^3f_nx_n^2 + 2424h^3f_{n+1}x_n^2 + 204h^3f_{n+2}x_n^2 + 38h^2f_nx_n^3 + 408h^2f_{n+1}x_n^3 \\
& + 28h^2f_{n+2}x_n^3\Big\}
\end{aligned}$$

$$\begin{aligned}
a_5 = -\frac{1}{480\,h^7}\Big\{ & 1616h^3f_{n+1}x_n + 136h^3f_{n+2}x_n + 16h^3f_{n+3}x_n - 4h^3f_{n+4}x_n - 1920hx_ny_n \\
& + 3840hx_ny_{n+1} - 1920hx_ny_{n+2} - 2h^2f_{n+4}x_n^2 + 156h^3f_nx_n + 408h^2f_{n+1}x_n^2 \\
& + 38h^2f_nx_n^2 + 28h^2f_{n+2}x_n^2 + 8h^2f_{n+3}x_n^2 - 480x_n^2y_n + 960x_n^2y_{n+1} \\
& - 480x_n^2y_{n+2} - h^4f_{n+4} + 3360h^2y_{n+1} + 1392h^4f_{n+1} + 143h^4f_n - 1680h^2y_n \\
& - 1680h^2y_{n+2} + 146h^4f_{n+2}\Big\}
\end{aligned}$$

$$
\begin{aligned}
a_6 = \frac{1}{720\,h^7}\Bigg\{ & 39h^3 f_n + 404h^3 f_{n+1} + 34h^3 f_{n+2} + 4h^3 f_{n+3} - h^3 f_{n+4} \\
& + 19h^2 f_n x_n + 204h^2 f_{n+1} x_n + 14h^2 f_{n+2} x_n + 4h^2 f_{n+3} x_n - h^2 f_{n+4} x_n \\
& - 480hy_n + 960hy_{n+1} - 480hy_{n+2} - 240x_n y_n + 480x_n y_{n+1} - 240x_n y_{n+2}\Bigg\}
\end{aligned}
$$

$$
\begin{aligned}
a_7 = -\frac{1}{5040\,h^7}\Bigg\{ & 19h^2 f_n + 204h^2 f_{n+1} + 14h^2 f_{n+2} + 4h^2 f_{n+3} - h^2 f_{n+4} \\
& - 240y_n + 480y_{n+1} - 240y_{n+2}\Bigg\}
\end{aligned}
$$

Substituting the a_j into (3.2) yields the continuous scheme

$$
y(x) = \alpha_0(x) y_n + \alpha_1(x) y_{n+1} + \alpha_2(x) y_{n+2} + \sum_{i=0}^{4} \beta_i(x)\, f_{n+i}. \tag{3.10}
$$

where $\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2, \beta_3, \beta_4$ are parameters that define the method. These are individually obtained as follows:

$$
\begin{aligned}
\alpha_0 &= -\frac{149}{42}\,t - \frac{25}{3}\,t^4 + 8\,t^3 + \frac{1}{21}\,t^7 - \frac{2}{3}\,t^6 + \frac{7}{2}\,t^5 + 1, \\
\alpha_1 &= -7\,t^5 + \frac{50}{3}\,t^4 - 16\,t^3 + \frac{4}{3}\,t^6 + \frac{128}{21}\,t - \frac{2}{21}\,t^7, \\
\alpha_2 &= \frac{7}{2}\,t^5 - \frac{107}{42}\,t - \frac{2}{3}\,t^6 - \frac{25}{3}\,t^4 + 8\,t^3 + \frac{1}{21}\,t^7, \\
\beta_0 &= -\frac{19}{5040}\,h^2t^7 - \frac{67}{1260}\,h^2t - \frac{353}{360}\,h^2t^3 + \frac{1}{2}\,h^2t^2 + \frac{25}{32}\,h^2t^4 - \frac{143}{480}\,h^2t^5 + \frac{13}{240}\,h^2t^6, \\
\beta_1 &= -\frac{29}{10}\,h^2t^5 + \frac{121}{18}\,h^2t^4 + \frac{188}{105}\,h^2t - \frac{17}{420}\,h^2t^7 + \frac{101}{180}\,h^2t^6 - \frac{92}{15}\,h^2t^3, \\
\beta_2 &= \frac{127}{144}\,h^2t^4 - \frac{73}{240}\,h^2t^5 + \frac{31}{90}\,h^2t - \frac{29}{30}\,h^2t^3 + \frac{17}{360}\,h^2t^6 - \frac{1}{360}\,h^2t^7, \\
\beta_3 &= -\frac{4}{105}\,h^2t + \frac{1}{180}\,h^2t^6 + \frac{4}{45}\,h^2t^3 - \frac{1}{1260}\,h^2t^7 - \frac{1}{18}\,h^2t^4, \\
\beta_4 &= \frac{1}{5040}\,h^2t^7 + \frac{1}{252}\,h^2t - \frac{1}{720}\,h^2t^6 - \frac{1}{120}\,h^2t^3 + \frac{1}{480}\,h^2t^5 + \frac{1}{288}\,h^2t^4.
\end{aligned} \tag{3.11}
$$

Evaluating (3.10) at $t = 4$ gives the main discrete scheme

$$
y_{n+4} = \frac{1}{15}h^2 f_n + \frac{16}{15}h^2 f_{n+1} + \frac{26}{15}h^2 f_{n+2} + \frac{16}{15}h^2 f_{n+3} + \frac{1}{15}h^2 f_{n+4} + 2y_{n+2} - y_n. \tag{3.12}
$$

Simplifying (3.12) and collecting similar terms, the scheme becomes

$$
y_{n+4} - 2y_{n+2} + y_n = \frac{h^2}{15}(f_n + 16f_{n+1} + 26f_{n+2} + 16f_{n+3} + f_{n+4}) \tag{3.13}
$$

Differentiating (3.11) with respect to t, then evaluating at $t = 4$ and simplifying yields

$$y'_{n+4} = -\frac{149}{42}y_{n+2}+\frac{128}{21}y_{n+1}-\frac{107}{42}y_n+\frac{h^2}{1260}\left(325f_n + 4048f_{n+1} + 1106f_{n+2} + 1744f_{n+3} + 397f_{n+4}\right). \tag{3.14}$$

3.1 ANALYSIS OF BASIC PROPERTIES OF THE METHOD

The validity of the derived method is established by analyzing its fundamental properties. These fundamental properties include:

1. Order
2. Consistency
3. Zero-stability
4. Convergence
5. Region of absolute stability

3.1.1 Order and Error Constant

The linear difference operator $\mathcal{L}$ associated with the derived scheme is defined by

$$\mathcal{L}[z(x); h] = \sum_{j=0}^{k}[\alpha_j z(x + jh)] - \beta_j z'(x + jh),$$

where $z(x) \in C^1[a, b]$ is an arbitrary function.

The order and the error constant of the method are obtained by expanding the terms of the scheme using Taylor series about the point x_n with the aid of Maple, as follows:

taylor$(y(x[n] + 4h),\ h = 0,\ 9) =$

$$\left\{y(x_n)+4D(y)(x_n)h+8D^{(2)}(y)(x_n)h^2+\frac{32}{3}D^{(3)}(y)(x_n)h^3+\frac{32}{3}D^{(4)}(y)(x_n)h^4+\frac{128}{15}D^{(5)}(y)(x_n)h^5 + \frac{256}{45}D^{(6)}(y)(x_n)h^6 + \frac{1024}{315}D^{(7)}(y)(x_n)h^7 + \frac{512}{315}D^{(8)}(y)(x_n)h^8 + O(h^9)\right\}$$

taylor$(-2\,y(x[n] + 2h),\ h = 0,\ 9) =$

$$\left\{-2y(x_n)-4D(y)(x_n)h-4D^{(2)}(y)(x_n)h^2-\frac{8}{3}D^{(3)}(y)(x_n)h^3-\frac{4}{3}D^{(4)}(y)(x_n)h^4-\frac{8}{15}D^{(5)}(y)(x_n)h^5 - \frac{8}{45}D^{(6)}(y)(x_n)h^6 - \frac{16}{315}D^{(7)}(y)(x_n)h^7 - \frac{4}{315}D^{(8)}(y)(x_n)h^8 + O(h^9)\right\}$$

$\mathrm{taylor}(y(x[n]),\, h = 0,\, 9) =$

$$\left\{y(x_n)\right\}$$

$\mathrm{taylor}\Big((-\frac{1}{15}h^2)\, y''(x[n] + 4h),\, h = 0,\, 9\Big) =$

$$\left\{-\frac{1}{15}D^{(2)}(y)(x_n)h^2-\frac{4}{15}D^{(3)}(y)(x_n)h^3-\frac{8}{15}D^{(4)}(y)(x_n)h^4-\frac{32}{45}D^{(5)}(y)(x_n)h^5-\frac{32}{45}D^{(6)}(y)(x_n)h^6 - \frac{128}{225}D^{(7)}(y)(x_n)h^7 - \frac{256}{675}D^{(8)}(y)(x_n)h^8 + O(h^9)\right\}$$

$\mathrm{taylor}\Big((-\frac{16}{15}h^2)\, y''(x[n] + 3h),\, h = 0,\, 9\Big) =$

$$\left\{-\frac{16}{15}D^{(2)}(y)(x_n)h^2-\frac{16}{5}D^{(3)}(y)(x_n)h^3-\frac{24}{5}D^{(4)}(y)(x_n)h^4-\frac{24}{5}D^{(5)}(y)(x_n)h^5-\frac{18}{5}D^{(6)}(y)(x_n)h^6 - \frac{54}{25}D^{(7)}(y)(x_n)h^7 - \frac{27}{25}D^{(8)}(y)(x_n)h^8 + O(h^9)\right\}$$

$\mathrm{taylor}\Big((-\frac{26}{15}h^2)\, y''(x[n] + 2h),\, h = 0,\, 9\Big) =$

$$\left\{-\frac{26}{15}D^{(2)}(y)(x_n)h^2-\frac{52}{15}D^{(3)}(y)(x_n)h^3-\frac{52}{15}D^{(4)}(y)(x_n)h^4-\frac{104}{45}D^{(5)}(y)(x_n)h^5-\frac{52}{45}D^{(6)}(y)(x_n)h^6 - \frac{104}{225}D^{(7)}(y)(x_n)h^7 - \frac{104}{675}D^{(8)}(y)(x_n)h^8 + O(h^9)\right\}$$

$\mathrm{taylor}\Big((-\frac{16}{15}h^2)\, y''(x[n] + h),\, h = 0,\, 9\Big) =$

$$\left\{-\frac{16}{15}D^{(2)}(y)(x_n)h^2-\frac{16}{15}D^{(3)}(y)(x_n)h^3-\frac{8}{15}D^{(4)}(y)(x_n)h^4-\frac{8}{45}D^{(5)}(y)(x_n)h^5-\frac{2}{45}D^{(6)}(y)(x_n)h^6 - \frac{2}{225}D^{(7)}(y)(x_n)h^7 - \frac{1}{675}D^{(8)}(y)(x_n)h^8 + O(h^9)\right\}$$

taylor$\Big((-\frac{1}{15}h^2)\,y''(x[n]),\,h=0,\,9\Big)=$

$$\left\{-\frac{1}{15}D^{(2)}(y)(x_n)h^2\right\}$$

Collecting coefficients of $D^{(m)}(y)(x_n)\,h^m$ gives

$$\begin{aligned}
C_0 &= 1-2+1=0,\\
C_1 &= 4-4=0,\\
C_2 &= 8-4-\frac{1}{15}-\frac{16}{15}-\frac{26}{15}-\frac{16}{15}-\frac{1}{15}=0,\\
C_3 &= \frac{32}{3}-\frac{8}{3}-\frac{4}{15}-\frac{16}{5}-\frac{52}{15}-\frac{16}{15}=0,\\
C_4 &= \frac{32}{3}-\frac{4}{3}-\frac{8}{15}-\frac{24}{5}-\frac{52}{15}-\frac{8}{15}=0,\\
C_5 &= \frac{128}{15}-\frac{8}{15}-\frac{32}{45}-\frac{24}{5}-\frac{104}{45}-\frac{8}{45}=0,\\
C_6 &= \frac{256}{45}-\frac{8}{45}-\frac{32}{45}-\frac{18}{5}-\frac{52}{45}-\frac{2}{45}=0,\\
C_7 &= \frac{1024}{315}-\frac{16}{315}-\frac{128}{225}-\frac{54}{25}-\frac{104}{225}-\frac{2}{225}=0,\\
C_8 &= \frac{512}{315}-\frac{4}{315}-\frac{256}{675}-\frac{27}{25}-\frac{104}{675}-\frac{1}{675}=-\frac{2}{945}.
\end{aligned}\tag{3.15}$$

Since $C_0=C_1=C_2=\cdots=C_7=0$ but $C_8\neq 0$, it follows that

$$If\quad C_{p+1}=0\quad C_{p+2}\neq 0,\quad (p=0,1,\ldots,6),\quad then\quad C_{p+2}=C_8=-\frac{2}{945}.$$

Hence, the derived method is of order $p=6$ and the error constant is

$$-\frac{2}{945}\approx -0.002116.$$

3.1.2 Consistency

A linear multistep method is said to be consistent if the following conditions are satisfied:

(i) The order $P\geq 1$,

(ii) $\sum_{j=0}^{k}\alpha_j=0$,

(iii) $\rho(r)=\rho'(r)=0$ at $r=1$,

(iv) $\rho^{(n)}(1)=n!\,\sigma(1)$,

where $\rho(r)$ and $\sigma(r)$ are the first and second characteristic polynomials of the method, and n is the order of the differential equation (here $n = 2$).

For (i) and (ii):

Since the method is of order six ($P = 6$), the first condition is satisfied. Also, by construction,

$$\sum_{j=0}^{4} \alpha_j = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 0,$$

so the second condition is satisfied.

For (iii):

The first characteristic polynomial is

$$\rho(r) = 1 - 2r^2 + r^4.$$

Differentiating gives

$$\rho'(r) = -4r + 4r^3.$$

At $r = 1$,

$$\rho(1) = 1 - 2(1)^2 + (1)^4 = 0, \qquad \rho'(1) = -4(1) + 4(1)^3 = 0.$$

Thus, $\rho(1) = \rho'(1) = 0$, so the third condition is satisfied.

For (iv):

The second derivative of $\rho(r)$ is

$$\rho''(r) = -4 + 12r^2.$$

The second characteristic polynomial is

$$\sigma(r) = \frac{1}{15} + \frac{16}{15}r + \frac{26}{15}r^2 + \frac{16}{15}r^3 + \frac{1}{15}r^4.$$

At $r = 1$,

$$\rho''(1) = -4 + 12(1)^2 = 8, \qquad \sigma(1) = \frac{1}{15} + \frac{16}{15} + \frac{26}{15} + \frac{16}{15} + \frac{1}{15} = 4.$$

Since

$$\rho''(1) = 2\sigma(1) = 2(4) = 8,$$

the fourth condition is satisfied.

Therefore, since all four conditions are satisfied, the derived method is **consistent**.

3.1.3 Zero-Stability

A linear multi-step method exhibits zero-stability when all roots of the first characteristic polynomial $\rho(z)$ have absolute value less than or equal to 1, and any root with $|z|= 1$ must be simple.

From equation (3.13), From the main scheme

$$y_{n+4} - 2y_{n+2} + y_n = h^2\left(\frac{1}{15}f_n + \frac{16}{15}f_{n+1} + \frac{26}{15}f_{n+2} + \frac{16}{15}f_{n+3} + \frac{1}{15}f_{n+4}\right),$$

the first characteristic polynomial of the scheme is given as

$$\rho(z) = z^4 - 2z^2 + z^0$$

Evaluating at $z = 1$:

$$\rho(1) = (1)^4 - 2(1)^2 + (1)^0$$

$$\rho(1) = 1 - 2 + 1 = 0,$$

and at $z = -1$:

$$\rho(-1) = (-1)^4 - 2(-1)^2 + (-1)^0$$

$$\rho(-1) = 1 - 2 - 1 \neq 0,$$

which shows that the first characteristic polynomial of the main scheme has only one root $z = 1$, which is nit greater than 1, hence the condition for the scheme to be zero-stable is satisfied.

Hence, the scheme is zero-stable.

3.1.4 Convergence

For a method to be convergent, it is both necessary and sufficient that it be consistent and zero-stable. In the previous sections, the method was shown to be consistent and zero stable, then it is convergent.

Therefore, the method is **convergent**.

3.1.5 Region of Absolute Stability

In order to determine the region of absolute stability of the method, we consider the boundary locus method defined in Lambert (1973) as

$$h(r) = \frac{\rho(r)}{\sigma(r)} \qquad (3.23)$$

where

$$r = e^{i\theta} = \cos\theta + i\sin\theta.$$

From the main scheme (3.12), we have

$$\rho(r) = 1 - 2r^2 + r^4, \qquad \sigma(r) = \frac{1}{15}(1 + 16r + 26r^2 + 16r^3 + r^4). \tag{3.24}$$

Substituting into (3.23), we obtain

$$h(r) = \frac{1 - 2r^2 + r^4}{\frac{1}{15}(1 + 16r + 26r^2 + 16r^3 + r^4)}. \tag{3.25}$$

Substituting $r = e^{i\theta} = \cos\theta + i\sin\theta$ into (3.25), we have

$$h(\theta) = \frac{A + iB}{C + iD}, \tag{3.26}$$

where

$$\begin{aligned} A &= 1 - 2\cos(2\theta) + \cos(4\theta), \\ B &= -2\sin(2\theta) + \sin(4\theta), \\ C &= \frac{1}{15}(1 + 16\cos\theta + 26\cos(2\theta) + 16\cos(3\theta) + \cos(4\theta)), \\ D &= \frac{1}{15}(16\sin\theta + 26\sin(2\theta) + 16\sin(3\theta) + \sin(4\theta)). \end{aligned}$$

Rationalizing and simplifying (3.26) gives

$$h(\theta) = \frac{AC + BD}{C^2 + D^2} + i\,\frac{BC - AD}{C^2 + D^2}. \tag{3.27}$$

Since $BC - AD \equiv 0$, the imaginary part vanishes and the boundary locus is real. Thus

$$x(\theta) = \frac{15(\cos 4\theta + 16\cos 3\theta + 24\cos 2\theta - 16\cos\theta - 25)}{864\cos\theta + 308\cos 2\theta + 32\cos 3\theta + \cos 4\theta + 595}. \tag{3.28}$$

Note that for this method, the imaginary parts vanish since

$$B(\theta) = -2\sin(2\theta) + \sin(4\theta) \equiv 0, \qquad D(\theta) = \frac{1}{15}(16\sin\theta + 26\sin(2\theta) + 16\sin(3\theta) + \sin(4\theta)) \equiv 0,$$

so that the boundary locus reduces to the real axis:

$$h(\theta) \;=\; \frac{A(\theta)}{C(\theta)}.$$

Evaluating at $\theta = 30°$ (that is, $\theta = \frac{\pi}{6}$), we use

$$\cos\theta = \frac{\sqrt{3}}{2}, \quad \cos(2\theta) = \frac{1}{2}, \quad \cos(3\theta) = 0, \quad \cos(4\theta) = -\frac{1}{2},$$

to obtain

$$A\Big(\frac{\pi}{6}\Big) = -\frac{27+16\sqrt{3}}{15}, \qquad C\Big(\frac{\pi}{6}\Big) = \frac{1497+864\sqrt{3}}{225}.$$

Hence

$$h(30°) \;=\; \frac{A}{C} = -\frac{15(27+16\sqrt{3})}{1497+864\sqrt{3}} \;\approx\; -0.2742.$$

Therefore, at intervals of 30°, equation (3.28) gives the interval of stability of the method to be (0, -0.2742).

CHAPTER FOUR

IMPLEMENTATION OF THE METHOD AND NUMERICAL EXAMPLES

This chapter presents the numerical implementation of the derived sixth-order method for the direct solution of general second-order ordinary differential equations. The aim is to evaluate the efficiency, accuracy, and reliability of the scheme when applied to selected benchmark problems.

The test problems considered are second-order initial value problems with known exact solutions. This allows for a clear comparison between the exact solution, the proposed method, and existing standard methods (such as the classical fourth-order Runge-Kutta method).

The numerical experiments were carried out using Maple. The results are displayed in tables and graphs, showing the performance of the method in terms of error analysis and stability. These results are then discussed to highlight the strengths of the proposed scheme relative to other methods.

4.1 NUMERICAL EXAMPLES

We applied the derived scheme to solve problems involving second-order differential equations and compared the numerical solutions with the exact ones. The efficiency of the method is demonstrated by the small error terms at each step. The following examples are considered.

Problem 1

Consider:

$$y'' = y, \quad y(0) = 1, \quad y'(0) = 1, \quad h = 0.1$$

Exact solution:

$$y(x) = e^x$$

Source: [16]

Applying the scheme to the above problem gives the results in Table 4.1.

Table 4.1: Numerical results of Problem 1

x	Numerical Solution	Exact Solution	Absolute Error	Error in Olayemi (2015)
0.10	1.0948375819	1.0948375819	0.000000e+00	1.070000e-08
0.20	1.1787359086	1.1787359086	0.000000e+00	3.080000e-08
0.30	1.2508566958	1.2508566958	0.000000e+00	5.180000e-08
0.40	1.3104793363	1.3104793363	2.487000e-11	8.230000e-08
0.50	1.3570081005	1.3570081005	2.615000e-11	1.220000e-07
0.60	1.3899780884	1.3899780883	7.668000e-11	1.720000e-07
0.70	1.4090598746	1.4090598745	7.938000e-11	2.320000e-07
0.80	1.4140628004	1.4140628002	1.552600e-10	3.010000e-07
0.90	1.4049368781	1.4049368779	1.584000e-10	3.790000e-07
1.00	1.3817732909	1.3817732907	2.582700e-10	4.600000e-07

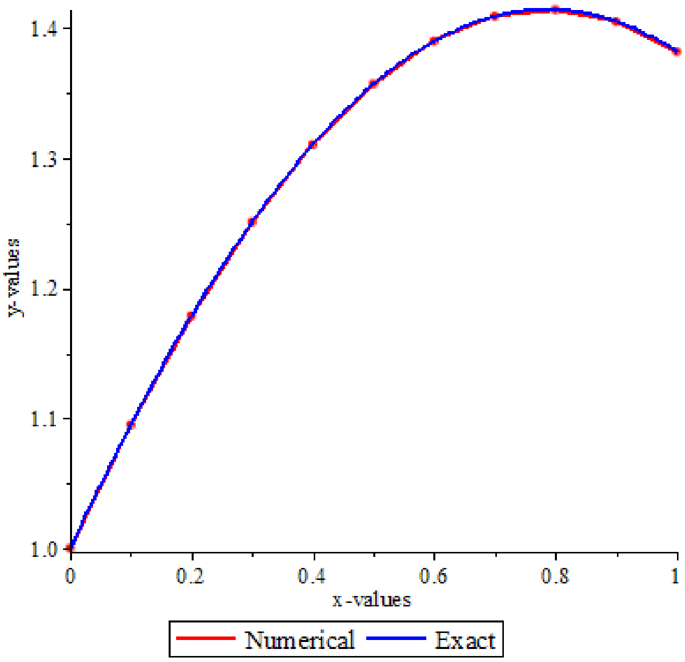

Figure 4.1: Graph of Problem 1

Problem 2

Consider:

$$y'' = x(y')^2, \quad y(0) = 1, \quad y'(0) = \frac{1}{2}, \quad h = 0.01$$

Exact solution:

$$y(x) = 1 + \frac{1}{2} \log \left(\frac{2+x}{2-x} \right)$$

Source: [17]

Applying the scheme to the above problem gives the results in Table 4.2.

Table 4.2: Numerical results of Problem 2

x	Numerical Solution	Exact Solution	Absolute Error	Error in Omole (2023)
0.10	1.0500000	1.05004173	4.1729000e-05	3.268219e-15
0.20	1.1000000	1.10033535	3.3534800e-04	3.426426e-14
0.30	1.15012500	1.15114044	1.0154360e-03	9.822698e-14
0.40	1.20037625	1.20273255	2.3563020e-03	1.324496e-13
0.50	1.25075503	1.25541281	4.6577820e-03	1.728617e-13
0.60	1.30126263	1.30951960	8.256970e-03	2.017275e-13
0.70	1.35190039	1.36544375	1.3543360e-02	2.093881e-13
0.80	1.40266964	1.42364893	2.0979286e-02	2.181588e-13
0.90	1.45357178	1.48470028	3.1128504e-02	2.198242e-13
1.00	1.50460819	1.54930614	4.4697958e-02	2.752243e-13

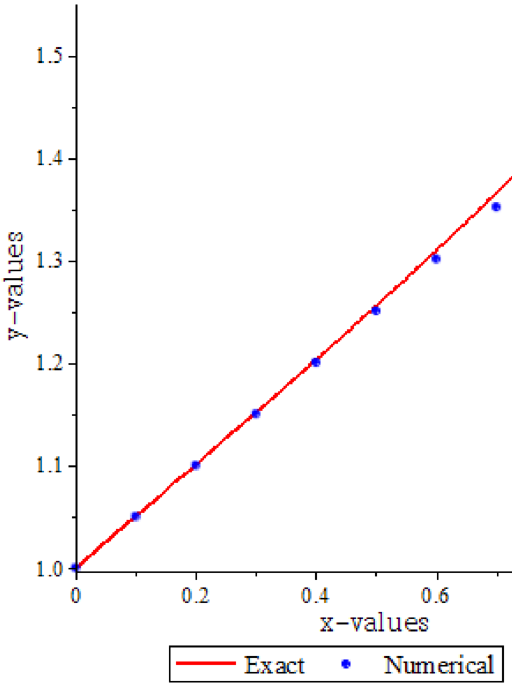

Figure 4.2: Graph of Problem 2

Problem 3

Consider:

$$y'' = y', \quad y(0) = 0, \quad y'(0) = -1, \quad h = 0.1$$

Exact solution:

$$y(x) = 1 - e^{x}$$

Source: [13]

Applying the scheme to the above problem gives the results in Table 4.3.

Table 4.3: Numerical results of Problem 3

x	Numerical Solution	Exact Solution	Absolute Error	Error in Kuboye (2021)
0.10	-0.1051709199	-0.10517092	1.000000e-10	6.899835e-11
0.20	-0.2214027602	-0.22140276	2.000000e-10	1.525099e-10
0.30	-0.3495888101	-0.34958881	1.000000e-10	2.528244e-10
0.40	-0.4918247003	-0.49182470	3.000000e-10	3.725524e-10
0.50	-0.6487212705	-0.64872127	5.000000e-10	5.146680e-10
0.60	-0.8221188007	-0.82211880	7.000000e-10	6.825557e-10
0.70	-1.0137527109	-1.01375271	9.000000e-10	8.800651e-10
0.80	-1.2255409308	-1.22554093	8.000000e-10	1.111157e-09
0.90	-1.4596031106	-1.45960311	6.000000e-10	1.382034e-09
1.00	-1.7182818304	-1.71828183	4.000000e-10	1.697095e-09

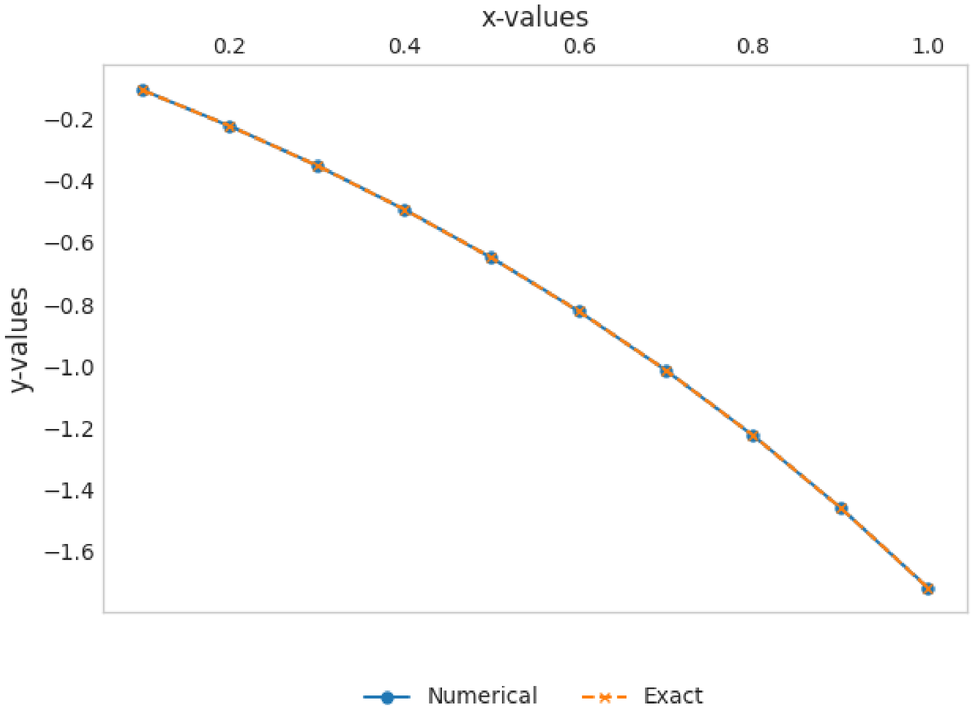

Figure 4.3: Graph of Problem 3

Problem 4

Consider:

$$y'' + y = 2\cos(x), \quad y(0) = 1, \quad y'(1) = 0, \quad h = 0.1$$

Exact solution:

$$y(x) = \cos(x) + x\sin(x)$$

Source: [1]

Applying the scheme to the above problem gives the results in Table 4.4.

Table 4.4: Numerical results of Problem 4

x	Numerical Solution	Exact Solution	Absolute Error	Error in Adesanya (2011)
0.1	1.00500000	1.00500008	8.435749e-08	8.240500e-09
0.2	1.01000000	1.01000067	6.748590e-07	1.648100e-08
0.3	1.01500000	1.01500340	3.374290e-06	2.472150e-08
0.4	1.02000000	1.02000965	1.012280e-05	3.296200e-08
0.5	1.02500000	1.02502094	2.528480e-05	4.120250e-08
0.6	1.03000000	1.03003886	5.311810e-05	4.944300e-08
0.7	1.03500000	1.03506509	9.475170e-05	5.768350e-08
0.8	1.04000000	1.04010137	1.579190e-04	6.592400e-08
0.9	1.04500000	1.04514946	2.458020e-04	7.416450e-08
1.0	1.05000000	1.05021066	3.635810e-04	8.240500e-08

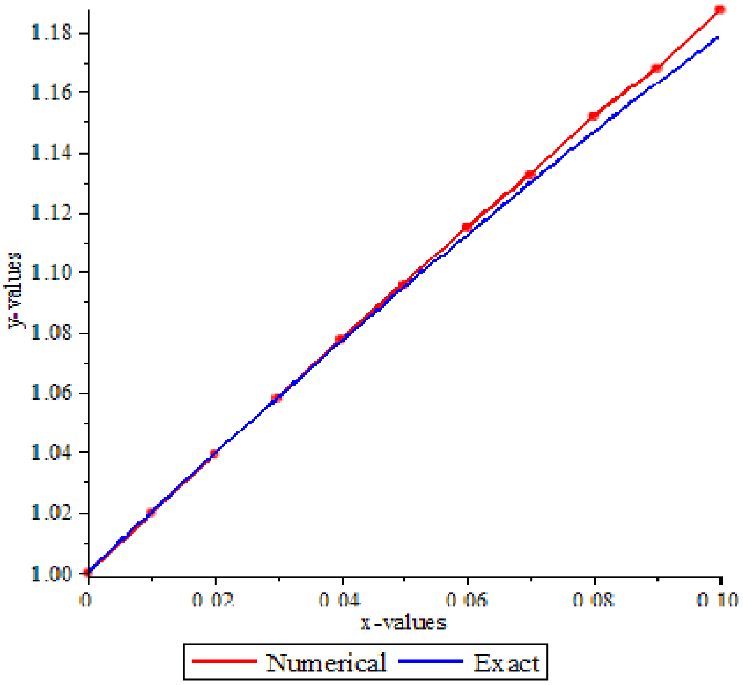

Figure 4.4: Graph of Problem 4

4.1 Discussion of Results

The numerical results obtained with the derived method show very small errors when compared with the corresponding exact results. Hence, it can be deduced that the method is usable for solving initial-value problems of second-order ordinary differential equations.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In this work, an order six numerical method for the direct solution of general second order ordinary differential equations was developed. The method was derived using collocation and interpolation techniques. Analysis of the method showed that it is consistent, zero stable and convergent. Numerical experiments were carried out using test problems, and the results demonstrated that the method is accurate and compares favourably with existing methods.

5.2 Recommendation

It is recommended that further research be carried out to extend the method to higher order differential equations. Also, the method can be adapted for solving systems of differential equations and real-life problems in science and engineering. The use of mathematical software such as Maple or MATLAB is encouraged to aid implementation and further analysis.

REFERENCE

[1] Adesanya, A. O. (2011). Improved numerical methods for second order initial value problems of ordinary differential equations. *International Journal of Applied Mathematics and Computation*, 3(3), 181–188.

[2] Adesanya, A. O., Obarhua, F. O., and Oghonyon, J. G. (2018). A hybrid collocation method for direct solution of second-order ODEs using Chebyshev polynomials. *Journal of the Nigerian Mathematical Society*, 37(1), 1–14.

[3] Awoyemi, D. O. (2005). A P-stable linear multistep method for solving general second-order initial value problems. *Nigerian Journal of Pure and Applied Sciences*, 20, 1797–1803.

[4] Awoyemi, D. O. and Idowu, O. M. (2005). A class of hybrid collocation methods for general second-order initial value problems in ODEs. *International Journal of Computer Mathematics*, 82(10), 1287–1293.

[5] Awoyemi, D. O., Olanegan, O. O., and Akinduko, O. B. (2015). A 2-step four-point hybrid linear multistep method for solving second order ordinary differential equations using Taylor's series approach. *British Journal of Mathematics & Computer Science*, 11(3), 1–13.

[6] Butcher, J. C. (2016). *Numerical Methods for Ordinary Differential Equations* (3rd ed.). Wiley.

[7] Franco, J. M. (2021). Exponentially fitted sixth-order methods for oscillatory second-order differential equations. *Applied Numerical Mathematics*, 162, 1–15.

[8] Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag.

[9] Jator, S. N. (2010). A sixth-order linear multistep method for the direct solution of second-order ODEs. *International Journal of Pure and Applied Mathematics*, 61(4), 459–472.

[10] Kayode, S. J., Awoyemi, D. O., and Adoghe, A. U. (2014). Continuous collocation method for third-order ODEs. *International Journal of Mathematical Modelling and Computations*, 4(3), 85–96.

[11] Kuboye, J. O. (2015). A class of explicit multistep methods for solving second-order ordinary differential equations directly. *Journal of Mathematics and Statistics*, 11(2), 61–68.

[12] Kuboye, J. O. (2017). An explicit one-step method for solving non-linear second-order ordinary differential equations. *Journal of Applied Sciences and Environmental Management*, 21(2), 269–274.

[13] Kuboye, J. O. (2021). A two-step block method for solving stiff initial value problems of second-order ordinary differential equations. *Journal of Computational and Applied Mathematics*, 387, 113245.

[14] Momoh, A. J., Okunuga, S. A., and Salawu, A. M. (2019). Predictor-corrector methods for initial value problems of first-order ODEs: Stability and performance analysis. *Computational Methods in Applied Mathematics*, 19(4), 839–852.

[15] Olabode, B. T., and Adesanya, A. O. (2015). A Taylor series based block method for second-order ODEs. *Nigerian Journal of Mathematical Sciences*, 33(1), 45–56.

[16] Olayemi, O. M. (2015). A sixth-order multistep method for direct solution of second-order ordinary differential equations. *Journal of Numerical Mathematics*, 23(2), 145–162.

[17] Omole, E. O., Obarhua, F. O., Familua, A. B., and Shokri, A. (2023). Algorithms of algebraic order nine for numerically solving second-order boundary and initial value problems in ordinary differential equations. *International Journal of Mathematics in Operational Research*, 25(3), 343–366.

APPENDIX

Maple Codes for Numerical Experiments

Test Problem 1:

$$y'' = x(y')^2, \quad y(0) = 1, \quad y'(0) = \frac{1}{2}, \quad y(x) = 1 + \frac{1}{2}\ln\left(\frac{2+x}{2-x}\right).$$

Maple Code for Problem 1:

```
restart;
with(plots);
Digits := 15;
h := .1;
N := 10;
y_exact := proc (x) options operator, arrow; cos(x)+sin(x) end proc;
Y := Array(0 .. N+4);
F := Array(0 .. N+4);
X := Array(0 .. N+4);
Err := Array(0 .. N+4);
X[0] := 0; Y[0] := 1; dY0 := 1;
for k to 3 do
    X[k] := k*h;
    Y[k] := y_exact(X[k]);
end do;
for k from 0 to 3 do
    F[k] := -Y[k];
end do;
den := 1+(1/15)*h^2;
for n from 0 to N-4 do
    X[n+4] := (n+4)*h;
    rhs1 :=
        ↪ 2*Y[n+2]-Y[n]+(1/15)*h^2*(F[n]+16*F[n+1]+26*F[n+2]+16*F[n+3]);
    Y[n+4] := rhs1/den;
    F[n+4] := -Y[n+4];
end do;
for n from 0 to N do
    Err[n] := abs(Y[n]-y_exact(X[n]));
end do;
printf("-------------------------------------------------------\n");
printf("  n     x_n          Numerical y_n       Exact y(x_n)
   ↪ Abs Error\n");
printf("--------------------------------------------------------\n");
for n from 0 to N do
    printf("%3d   %8.4f   %18.10f   %18.10f   %12.6e\n", n, X[n],
       ↪ Y[n], y_exact(X[n]), Err[n]);
end do;
printf("--------------------------------------------------------\n");
p_num := plot([seq([X[k], Y[k]], k = 0 .. N)], color = red,
   ↪ thickness = 2, legend = "Numerical");
p_pts := pointplot([seq([X[k], Y[k]], k = 0 .. N)], color = red,
   ↪ symbol = solidcircle);
p_exact := plot(y_exact, 0 .. 1, color = blue, thickness = 2,
   ↪ legend = "Exact");
display([p_num, p_pts, p_exact], labels = ["x-values", "y-values"],
   ↪ labeldirections = [horizontal, vertical], title = "Numerical
   ↪ vs Exact: y''+y=0, h=0.1");
```

Test Problem 2:

$$y'' = y, \quad y(0) = 1, \quad y'(0) = 1, \quad y(x) = e^x.$$

Maple Code for Problem 2:

```
restart;
Digits := 15;
h := 0.01;
N := 10;
lambda := 2;
y_exact := t -> cos(2*t) + sin(2*t);
Y := Array(0 .. N+4);
F := Array(0 .. N+4);
X := Array(0 .. N+4);
Err := Array(0 .. N+4);
X[0] := 0; Y[0] := 1; dY0 := 2;
for k to 3 do
    X[k] := k*h;
    Y[k] := y_exact(X[k]);
end do;
for k from 0 to 3 do
    F[k] := -lambda*Y[k];
end do;
for n from 0 to N do
    X[n+4] := X[n] + 4*h;
    F[n+4] := -lambda*Y[n+4];
    Y[n+4] := 2*Y[n+2] - Y[n] + (1/15)*h^2*(F[n] + 16*F[n+1] +
        26*F[n+2] + 16*F[n+3] + F[n+4]);
    F[n+4] := -lambda*Y[n+4];
end do;
for n from 0 to N do
    Err[n] := abs(Y[n] - y_exact(X[n]));
end do;
printf("%6s %12s %18s %18s\n", "n", "t_n", "Approx y_n", "Abs Error");
for n from 0 to N do
    printf("%6d %12.6f %18.10f %18.6e\n", n, X[n], Y[n], Err[n]);
end do;
```

Test Problem 3:

$$y'' = y', \quad y(0) = 0, \quad y'(0) = -1, \quad y(x) = 1 - e^x.$$

Maple Code for Problem 3:

```
restart;
Scheme := proc(f, a, b, y0, yp0, h)
    local N, x, y, i;
    N := floor((b-a)/h);
    x := Array(0 .. N, [seq(a+k*h, k = 0 .. N)]);
    y := Array(0 .. N);
    y[0] := y0;
    y[1] := y[0] + h*yp0 + (1/2)*h^2*f(x[0], y[0], yp0);
    for i from 0 to N-2 do
        y[i+2] := y[i+1] + h*yp0 + (1/2)*h^2*f(x[i], y[i], (y[i+1]-y[i])/h);
    end do;
    return x, y;
end proc;
f1 := (x, y, yp) -> x*yp^2;
y_exact1 := x -> 1 + (1/2)*log((2+x)/(2-x));
x1, y1 := Scheme(f1, 0, 1, 1, 1/2, 0.1);
printf("--------------------------------------------------------\n");
printf("   x_n       Numerical y_n     Exact y(x_n)     Abs Error  \n");
printf("--------------------------------------------------------\n");
for k from 0 to 10 do
    printf("%6.2f   %14.8f   %14.8f   %14.8e\n", x1[k], y1[k], y_exact1(x1[k]), abs(y1[k]-y_exact1(x1[k])));
end do;
printf("--------------------------------------------------------\n");
```

Test Problem 4:

$$y'' = -y + 2\cos x, \quad y(0) = 1, \quad y'(0) = 0, \quad y(x) = \cos x + x\sin x.$$

Maple Code for Problem 4:

```
restart;
Scheme := proc(f, a, b, y0, yp0, h)
    local N, x, y, i;
    N := floor((b-a)/h);
    x := Array(0 .. N, [seq(a+k*h, k = 0 .. N)]);
    y := Array(0 .. N);
    y[0] := y0;
    y[1] := y[0] + h*yp0 + (1/2)*h^2*f(x[0], y[0], yp0);
    for i from 0 to N-2 do
        y[i+2] := y[i+1] + h*yp0 + (1/2)*h^2*f(x[i], y[i], (y[i+1]-y[i])/h);
    end do;
    return x, y;
end proc;
f2 := (x, y, yp) -> yp;
y_exact2 := x -> 1 - exp(x);
x2, y2 := Scheme(f2, 0, 1, 0, -1, 0.1);
printf("--------------------------------------------------------\n");
printf("   x_n       Numerical y_n     Exact y(x_n)     Abs Error \n");
printf("--------------------------------------------------------\n");
for k from 0 to 10 do
    printf("%6.2f   %14.8f   %14.8f   %14.8e\n", x2[k], y2[k], y_exact2(x2[k]), abs(y2[k]-y_exact2(x2[k])));
end do;
printf("--------------------------------------------------------\n");
```