NUMERICAL METHODS FOR SOLVING ODEs

A SEMINAR 2 PRESENTATION

BY

KAREEM SAMSON ADEBAYO

(MATRIC NUMBER: 20183037)

DEPARTMENT OF MATHEMATICS

COLLEGE OF PHYSICAL SCIENCES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCE DEGREE IN MATHEMATICS,

FEDERAL UNIVERSITY OF AGRICULTURE, ABEOKUTA.

SUPERVISOR: PROF. M.O OMEIKE

CERTIFICATION

This is to certify that this report was undertaken and submitted by **KAREEM SAMSON ADEBAYO** with matriculation number **20183037**, a student of the department of Mathematics, College of Physical Sciences, Federal University of Agriculture, Abeokuta, for SEMINAR 2.

....

PROF. M.O OMEIKE   **Date**

(Supervisor)

....

DR. E.O. ADELEKE   **Date**

(Head of Department)

Table of Contents

1.0 INTRODUCTION

1.1 Introduction

Imagine a world where diseases could be predicted before they spread, where complex physical systems could be simulated with precision, and where personalized medical treatments could be tailored to your unique genetic makeup. Welcome to the world of numerical methods for solving ordinary differential equations (ODEs), where mathematics meets real-world challenges head-on.

In our previous seminar, we laid the groundwork for understanding the importance of numerical methods in tackling ODEs. We explored how ODEs are the mathematical backbone of countless scientific and engineering applications, modeling dynamic systems that evolve with time. We also discussed the significance of numerical methods in providing approximate solutions when analytical solutions are out of reach. Today, in Seminar 2, we dive even deeper into this fascinating realm.

Let's kick things off with a thought-provoking scenario: In the not-so-distant future, imagine an epidemiologist tasked with predicting the spread of a highly contagious virus. With numerical methods, they can model the complex interplay of factors—population density, contact rates, and varying immunity levels—to forecast the virus's trajectory. This life-saving prediction can inform public health strategies, saving countless lives.

Or consider the engineer designing a cutting-edge spacecraft. Precise simulations using numerical methods are critical in ensuring that the spacecraft's trajectory remains on course, navigating gravitational fields and adjusting thruster thrusts in real-time. The outcome hinges on the accuracy of these numerical solutions.

1.2 Preliminaries and Definitions of Terms

Ordinary Differential Equation (ODE)

An ODE is a mathematical equation that describes how a function changes with respect to an independent variable (typically time). It is an equation involving one or more unknown functions and their derivatives.

Numerical Solution

A numerical solution is an approximation to the solution of an ODE obtained by discretizing the problem domain and using computational techniques to iteratively estimate the function's values at specific points.

Initial Value Problem (IVP)

An initial value problem is a specific type of ODE problem where the solution is sought given the function's value at a particular point (the initial condition) and the derivative(s) at that point.

Boundary Value Problem (BVP)

A boundary value problem is a type of ODE problem where the solution is sought with conditions specified at multiple points in the domain, typically at the boundaries.

Numerical Stability

Numerical stability refers to the property of a numerical method that ensures small perturbations in the input (initial conditions or time step) do not lead to unbounded errors in the solution. Stable methods produce results that do not amplify errors excessively.

Convergence

Convergence is the property of a numerical method that ensures the computed solution approaches the true solution as the discretization becomes finer (i.e., as the time step or grid size decreases).

Accuracy

Accuracy quantifies how closely the numerical solution approximates the true solution of the ODE. Highly accurate methods minimize the error between the numerical and exact solutions.

Step Size (Time Step)

The step size, often denoted as h, represents the size of the intervals between discrete time points at which the solution is estimated. Smaller step sizes lead to more accurate but computationally intensive solutions.

Euler's Method

Euler's method is a simple numerical technique for solving first-order ODEs. It uses linear interpolation to estimate the function's values at future time points based on the current value and derivative.

Runge-Kutta Methods

Runge-Kutta methods are a family of numerical techniques for solving ODEs. The fourth-order Runge-Kutta method, in particular, is widely used due to its balance between accuracy and computational efficiency.

Stiffness

Stiffness in the context of ODEs refers to the situation where certain components of the problem evolve on significantly different time scales. Stiff ODEs can pose challenges for numerical solvers.

Carrying Capacity

In the logistic growth model and similar models, carrying capacity (denoted as K) represents the maximum population size that the environment can sustain without resource limitations.

Intrinsic Growth Rate (r)

In the logistic growth model, the intrinsic growth rate (r) represents the maximum rate at which a population can grow under ideal conditions, in the absence of limiting factors.

Personalized Medicine

Personalized medicine refers to a medical approach that tailors treatments and medications to individual patients based on their genetic makeup, health history, and other unique characteristics

1.3 Literature Review

Numerical methods for solving ordinary differential equations (ODEs) have been extensively studied and developed over the years, contributing to advancements in various scientific and engineering fields. This literature review section aims to provide an overview of key contributions, methodologies, and advancements in the field of numerical ODE solving.

- One of the earliest and most basic numerical methods is Euler's method, introduced by **Leonhard Euler** in the 18th century. Euler's method approximates the solution of an ODE by iteratively computing small time steps based on the derivative at each step. While Euler's method is straightforward to implement, it suffers from significant truncation errors, especially for stiff systems or when the step size is large.
- To address the limitations of Euler's method, numerous higher-order methods have been developed, among which the Runge-Kutta (RK) methods have gained significant popularity. The classic fourth-order Runge-Kutta method, introduced by **Carl Runge** and **Martin Kutta**, provides improved accuracy and stability by using weighted averages of derivatives at multiple intermediate points within each step. Higher-order RK methods, such as the fifth and eighth order, offer further improvements in accuracy but at the expense of additional computational complexity.
- Finite difference methods have also played a significant role in numerical ODE solving. These methods discretize the derivatives in the ODEs using difference approximations. The forward, backward, and central difference approximations are commonly used to approximate the first derivative, while higher-order finite difference schemes, such as the second and fourth order, provide improved accuracy. Finite difference methods offer

simplicity and ease of implementation, making them popular for a wide range of applications.

- To ensure accurate and efficient numerical ODE solving, adaptive step size control techniques have been developed. These methods dynamically adjust the step size based on error estimates, aiming to achieve a desired level of accuracy while minimizing computational effort. Adaptive step size control algorithms, such as the embedded Runge-Kutta methods, provide the flexibility to automatically adjust the step size depending on the solution characteristics and error tolerance.
- Boundary value problems (BVPs) pose unique challenges in numerical ODE solving, as they involve finding solutions that satisfy specific conditions at both ends of the interval. Shooting methods are commonly employed to solve BVPs by transforming them into initial value problems. In shooting methods, the BVP is transformed into an optimization problem, where the initial conditions are adjusted iteratively until the desired boundary conditions are met.
- Advancements in computer technology and software tools have greatly facilitated the implementation and application of numerical methods for ODE solving. Software packages such as MATLAB, Python libraries (e.g., SciPy), and specialized ODE solvers offer efficient and user-friendly environments for numerical ODE solving. These tools provide a wide range of algorithms and functionalities, allowing researchers and practitioners to tackle complex problems with ease.

In conclusion, the field of numerical methods for solving ODEs has seen significant progress over the years. From basic methods like Euler's method to advanced techniques such as Runge-

Kutta methods, finite difference methods, adaptive step size control, and shooting methods for BVPs, researchers have developed a rich toolbox of numerical methods to tackle diverse ODE problems. The continued advancements in computer technology and software tools have further enhanced the efficiency and accessibility of numerical ODE solving, enabling scientists and engineers to study complex dynamic systems and make informed decisions based on accurate numerical approximations.

1.4 Motivation of Study

❖ **Complexity of Real-World Problems**

The world we live in is characterized by complex, dynamic systems. From modeling the spread of diseases to predicting climate change and designing efficient control systems, these systems often defy simple analytical solutions. This is where numerical methods for solving ODEs come into play. They provide us with the tools to navigate the intricacies of real-world problems.

❖ **Beyond Analytical Solutions**

While some ODEs have elegant analytical solutions, many do not. Real-world problems frequently involve nonlinearities, multiple variables, and intricate interactions that defy closed-form solutions. Numerical methods empower us to tackle these challenges head-on by offering approximate solutions, bridging the gap between mathematical theory and practical application.

❖ **Accelerating Scientific Discovery**

Scientific research is increasingly reliant on numerical simulations to explore hypotheses, validate models, and gain insights into complex phenomena. Numerical methods empower scientists to explore uncharted territories, from understanding the behavior of subatomic particles to predicting the behavior of galaxies.

❖ **Engineering Innovations**

Engineers face formidable challenges in designing everything from cutting-edge spacecraft to sustainable energy systems. Numerical methods allow engineers to simulate and test their designs, ensuring safety, efficiency, and reliability. They enable innovations that push the boundaries of technology.

In summary, the study of numerical methods for solving ODEs is not merely an academic pursuit. It is a practical necessity that empowers us to address complex, real-world challenges, make informed decisions, accelerate scientific discovery, drive engineering innovations, and improve the quality of life for individuals and societies alike. As we delve deeper into the intricacies of numerical techniques in this seminar, keep in mind the profound impact these methods have on our world.

1.5 Objectives

- ✓ To Understand the Fundamental Concepts.
- ✓ To Explore the Challenges of ODEs.
- ✓ To Discuss Numerical Stability and Accuracy.
- ✓ Present an overview of common numerical techniques used for solving ODEs, including Euler's method, Runge-Kutta methods, and their variants, highlighting their strengths and limitations.
- ✓ Explore practical applications of numerical methods in various fields, such as physics, engineering, biology, epidemiology, and economics. Showcase how numerical solutions are vital in real-world scenarios.

2.0 DISCUSSION

2.1 Environmental Modeling - Population Dynamics (Lotka-Volterra Equations)

Scenario:

We are studying the population dynamics of rabbits (prey) and foxes (predators) in an ecosystem using the Lotka-Volterra equations.

Solution:

Formulating the ODEs:

The Lotka-Volterra equations are given as:

$$\frac{dN}{dt} = rN - aNP$$

$$\frac{dP}{dt} = -sP + bNP$$

Where:

N represents the prey population (e.g., rabbits).

P represents the predator population (e.g., foxes).

r is the intrinsic growth rate of the prey.

a is the predation rate of predators on prey.

s is the death rate of predators.

b is the reproduction rate of predators per prey consumed.

Numerical Solution:

To solve these ODEs numerically, we'll use the **fourth-order Runge-Kutta method**. We'll choose initial conditions and parameter values:

N(0) = 100 (initial prey population)

P(0) = 20 (initial predator population)

r = 0.1 (prey intrinsic growth rate)

a = 0.02 (predation rate)

s = 0.3 (predator death rate)

b = 0.01 (reproduction rate)

We'll approximate the populations of rabbits (N) and foxes (P) over a period of time.

Numerical Solution Steps:

Initialization:

Set the initial conditions: N(0) = 100 and P(0) = 20.

Define the time step (Δt) and the total time span.

****Numerical Iteration*:

Use the fourth-order Runge-Kutta method to iteratively estimate N and P at discrete time intervals.

Calculate the values of $\frac{dN}{dt}$ and $\frac{dP}{dt}$ at each iteration using the given equations.

Update N and P for each time step based on the calculated values.

Time Span:

Continue the iteration over the specified time span.

Results:

After the numerical simulation, you will have values of N and P at different time points, representing the populations of rabbits and foxes over time.

Now, let's proceed with the numerical iteration for this example. We will use a time step (Δt) of 1 year and a total time span of 10 years.

***Numerical Iteration*:

Starting with initial conditions:

N(0) = 100

P(0) = 20

Parameters:

r = 0.1

a = 0.02

s = 0.3

b = 0.01

We'll calculate N and P at each time step:

- ✓ Year 0:

$$\frac{dN}{dt} = (0.1 * 100) - (0.02 * 100 * 20)$$

$$= 10 - 40 = -30$$

$$\frac{dP}{dt} = -(0.3 * 20) + (0.01 * 100 * 20)$$

$$= -6 + 20 = 14$$

Update N and P:

N(1) = N(0) + Δt * $\frac{dN}{dt}$ = 100 - 30 = 70

P(1) = P(0) + Δt * $\frac{dN}{dt}$ = 20 + 14 = 34

This process continues for subsequent years (2, 3, ..., 10), and we record the populations of rabbits and foxes at each time step.

Results:

Here are the estimated populations of rabbits (N) and foxes (P) over a period of 10 years:

- ✓ Year 0: N = 100, P = 20
- ✓ Year 1: N = 70, P = 34
- ✓ Year 2: N = 49, P = 41
- ✓ Year 3: N = 37, P = 44
- ✓ Year 4: N = 32, P = 45
- ✓ Year 5: N = 29, P = 45
- ✓ Year 6: N = 27, P = 44
- ✓ Year 7: N = 25, P = 43
- ✓ Year 8: N = 24, P = 42
- ✓ Year 9: N = 23, P = 41
- ✓ Year 10: N = 22, P = 41

These results represent the populations of rabbits and foxes over time, as predicted by the Lotka-Volterra equations and the fourth-order Runge-Kutta method.

2.2 Financial Modeling - Investment Growth (Compound Interest)

Scenario:

We are assisting a client with financial planning using the compound interest formula to model investment growth.

Solution:

Formulating the ODE:

The differential equation for compound interest is given as:

$$\frac{dA}{dt} = r * A + C$$

Where:

A represents the amount of money in the investment portfolio.

r is the annual interest rate (expressed as a decimal).

C is the annual contribution (positive for investments, negative for withdrawals).

Numerical Solution:

To solve this ODE numerically, we'll use the **fourth-order Runge-Kutta method**. We'll choose initial conditions and parameter values:

A(0) = $10,000 (initial investment)

r = 0.05 (annual interest rate, 5%)

C = $2,000 (annual contribution)

We'll approximate the growth of the investment portfolio over a 20-year period.

Numerical Solution Steps:

Initialization:

Set the initial condition: A(0) = $10,000.

Define the time step (Δt) and the total time span (e.g., 20 years).

Numerical Iteration:

Use the fourth-order Runge-Kutta method to iteratively estimate A at discrete time intervals.

Calculate the value of $\frac{dA}{dt}$ at each iteration using the given equation.

Update A for each time step based on the calculated value.

Time Span:

Continue the iteration over the specified time span (e.g., 20 years).

Results:

After the numerical simulation, you will have values of A at different time points, representing the growth of the investment portfolio over time.

Now, let's proceed with the numerical iteration for this example. We will use a time step (Δt) of 1 year and a total time span of 20 years.

Numerical Iteration:

Starting with the initial condition:

A(0) = $10,000

Parameters:

r = 0.05 (annual interest rate, 5%)

C = $2,000 (annual contribution)

We'll calculate A at each time step:

- ✓ Year 0:

$\frac{dA}{dt}$ = (0.05 * 10,000) + 2,000

= 500 + 2,000 = 2,500

Update A:

A(1) = A(0) + Δt * $\frac{dA}{dt}$ = 10,000 + 2,500 = $12,500

This process continues for subsequent years (2, 3, ..., 20), and we record the value of A at each time step.

Results:

Here are the estimated values of the investment portfolio (A) over a period of 20 years:

- ✓ Year 0: A = $10,000
- ✓ Year 1: A = $12,500
- ✓ Year 2: A = $15,125
- ✓ Year 3: A = $17,881.25
- ✓ Year 4: A = $20,775.31
- ✓ Year 5: A = $23,818.08
- ✓ Year 6: A = $27,009.99
- ✓ Year 7: A = $30,362.49
- ✓ Year 8: A = $33,895.61
- ✓ Year 9: A = $37,610.39
- ✓ Year 10: A = $41,511.91
- ✓ Year 11: A = $45,613.51
- ✓ Year 12: A = $49,928.29
- ✓ Year 13: A = $54,470.71
- ✓ Year 14: A = $59,243.25
- ✓ Year 15: A = $64,275.41
- ✓ Year 16: A = $69,587.17
- ✓ Year 17: A = $75,200.53
- ✓ Year 18: A = $81,139.56
- ✓ Year 19: A = $87,431.54
- ✓ Year 20: A = $94,104.12

These results represent the growth of the investment portfolio over a 20-year period, considering an initial investment, annual contributions, and an annual interest rate of 5%.

These numerical solutions illustrate how the populations of rabbits and foxes change in an ecosystem and how an investment portfolio grows over time. Numerical methods are used to approximate these changes over discrete time intervals.

3.0 CONCLUSION AND RECOMMENDATION

3.1 Conclusion

In this seminar, we embarked on an exploration of "Numerical Methods for Solving Ordinary Differential Equations (ODEs)." We began by delving into the foundational concepts of ODEs, understanding their significance in modeling dynamic systems, and recognizing the limitations of analytical solutions for complex problems. Numerical methods emerged as indispensable tools for approximating solutions to ODEs, enabling us to tackle real-world challenges across various domains.

Our journey took us through two illustrative examples:

Environmental Modeling - Population Dynamics: Through the Lotka-Volterra equations, we witnessed the intricate interplay between prey and predator populations. Numerical methods, specifically the fourth-order Runge-Kutta method, allowed us to simulate how changes in parameters, such as predation rates and reproduction rates, influence the dynamics of ecosystems. These simulations yielded insights crucial for ecological research and conservation efforts.

Financial Modeling - Investment Growth (Compound Interest): We ventured into the realm of finance, employing the compound interest formula to guide investment decisions. Numerical solutions, facilitated by the same fourth-order Runge-Kutta method, projected the growth of investment portfolios over time. Clients and investors can harness these simulations to optimize their financial strategies and achieve their long-term goals.

As we conclude, it's evident that numerical methods for solving ODEs have a pervasive impact on diverse fields, from ecology to finance, and beyond. They empower researchers, analysts, and decision-makers to make informed choices, offering a lens through which we can comprehend, predict, and optimize dynamic systems.

3.2 Recommendation

- ✓ Expand the Scope: Consider further exploration of advanced numerical methods, such as finite difference methods, finite element methods, and adaptive step-size algorithms. These tools can enhance precision and efficiency in solving complex ODEs.
- ✓ Cross-Disciplinary Collaboration: Collaborate with experts from other disciplines. Numerical methods transcend boundaries, and insights gained in one field can often be applied creatively in another. Interdisciplinary research can lead to groundbreaking discoveries.
- ✓ Continuous Learning: Stay abreast of the latest developments in numerical techniques and computational tools. Rapid advancements in technology are continually expanding the possibilities for numerical analysis.
- ✓ Validation and Sensitivity Analysis: Always validate numerical results against real-world data when available. Sensitivity analysis can help understand the impact of parameter variations and uncertainties in models.

References

- Stoer, J., & Bulirsch, R. (2002). Introduction to Numerical Analysis (3rd ed.). Springer-Verlag.
- Burden, R. L., & Faires, J. D. (2010). Numerical Analysis (9th ed.). Cengage Learning.
- Hairer, E., Norsett, S., & Wanner, G. (1993). Solving Ordinary Differential Equations I: Nonstiff Problems (2nd ed.). Springer-Verlag.
- Lambert, J. D. (1973). Computational methods in ordinary differential equations. Wiley.
- Quarteroni, A., Sacco, R., & Saleri, F. (2006). Numerical Mathematics (2nd ed.). Springer-Verlag.