

Project Proposal: Unified Code for Collocation  
Multistep Methods in Solving Stiff Systems of  
Boundary Value Problems

Your Name

January 15, 2024

## 0.1 Chapter 1: Introduction

### 0.1.1 Background

Mathematical models in a vast range of disciplines, from science and technology to sociology and business, describe how quantities *change*. This leads naturally to a language of ordinary differential equations (ODEs). Ordinary Differential Equations (ODEs) are a type of differential equation that involves an unknown function and its derivatives.

ODEs are of paramount significance in mathematical modeling because they provide a concise and powerful way to describe how a quantity changes concerning time or another independent variable. The ability to capture the rate of change of a variable makes ODEs essential in understanding dynamic processes, predicting future states, and optimizing system behavior. In many important cases of differential equations, analytic solutions are difficult or impossible to obtain and time consuming. The mathematical modelling of many problems in physics, engineering, chemistry, biology, and many more gives rise to systems of ordinary differential equation. Yet, the number of instances where an exact solution can be found by analytical means is very limited (Lambert, 1977). In many important cases of differential equations, analytic solutions are difficult or impossible to obtain and time consuming, Hence the need for an approximate, or a numerical method.

In contemporary scientific and engineering research, the formulation of complex mathematical models often leads to the generation of differential equations that defy closed-form solutions. This persistent challenge has underscored the growing significance of approximate, or numerical, methods in tackling intricate mathematical problems. Among these methods, numerical techniques for ordinary differential equations (ODEs) stand out as indispensable tools, providing a robust means to compute numerical approximations to the solutions of these challenging equations. This necessity becomes even more pronounced when dealing with stiff systems of differential equations, where rapid variations in solution components pose additional complexities. Classical analytical methods, while powerful and elegant, encounter limitations when confronted with intricate mathematical formulations. Numerical methods step in precisely where analytical methods fall short, offering a practical avenue to obtain solutions when exact expressions are elusive.

Various advanced numerical techniques, such as implicit methods, exponential integrators, and collocation multistep methods, have proven effective in addressing the challenges posed by stiff systems. These methods excel in capturing the dynamics of stiff ODEs by incorporating strategies that adapt to the varying time scales inherent in the system. Implicit methods, for instance, allow for larger time steps, enhancing stability in the presence of stiffness.

With the advent of powerful computing technologies, numerical methods for ODEs have witnessed significant advancements. High-performance computing allows researchers and engineers to tackle more complex problems, simulate intricate physical phenomena, and explore the behavior of systems over extended

timeframes. These simulations not only aid in understanding complex systems but also contribute to the optimization and design of real-world applications.

### **0.1.2 Problem Statement**

Many studies on solving the equations of stiff ordinary differential equations (ODEs) have been done by researchers or mathematicians specifically. With the numbers of numerical methods that currently exist in the literature, extensive research has been done to unveil the comparison between their rate of convergence, number of computations, accuracy, and capability to solve certain type of test problems (Enright, Hull, & Lindberg, 1975) . The well-known numerical methods that are used widely are from the class of BDFs or commonly understood as Gear’s Method (Byrne, Hindmarsh, Jackson, & Brown, 1977). However, many other methods that have evolved to this date are for solving stiff ODEs which arise in many fields of the applied sciences (Yatim, Ibrahim, Othman, & Suleiman, 2013). The problems considered in this paper are for the numerical solution of the boundary value problem.

The problem at hand is to develop a unified code that amalgamates the strengths of collocation and multistep methods, aiming to provide a versatile, accurate, and computationally efficient tool for solving stiff BVPs. This research seeks to bridge the gap in existing methodologies by creating a unified numerical framework capable of addressing the unique challenges posed by stiff systems, ultimately contributing to advancements in the numerical solution of stiff BVPs across diverse scientific and engineering disciplines.

### **0.1.3 Aim and Objectives**

#### **Aim:**

The primary goal of this project is to develop a unified Python-based numerical code that seamlessly integrates collocation and multistep methods for the efficient and accurate solution of stiff systems of boundary value problems (BVPs)

#### **Objectives:**

1. Develop a unified code that seamlessly integrates collocation and multistep methods.
2. Assess the accuracy, efficiency, and versatility of the proposed code.
3. Provide a user-friendly tool for researchers and practitioners working with stiff BVPs.

### **0.1.4 Scope of Study**

The project centers on the implementation of a numerical code using the Python programming language that integrates collocation and multistep methods. The

code aims to solve stiff systems of boundary value problems (BVPs) efficiently and accurately and also designing the code with a user-friendly interface to cater to researchers and practitioners working with stiff BVPs in flutter

### 0.1.5 Significance of Study

This codebase tends to provide a tool that combines the strengths of collocation and multistep methods, offering a more comprehensive approach to solving stiff BVPs, and also a user-friendly interface for researchers and practitioners working with stiff BVPs in flutter.

### 0.1.6 Definition of Terms

1. Ordinary differential equation:
2. Numerical method: A numerical method is a difference equation involving a number of consecutive approximations  $y_{n+j}, j = 0, 1, 2 \dots k$  from which it be possible to compute sequentially the sequence  $y_n | n = 0, 1, 2, \dots N$ . The integer  $k$  is called a step-number; if  $k = 1$ , the method is called a one-step method, while if  $k > 1$ , the method is called a *one-step method*
3. Step Length (Mesh-Size): A point within the solution domain where the solution is approximated or calculated. The **step length** ( $h$ ) is the size of the interval between consecutive points in the independent variable (e.g., time or space) at which the solution of a differential equation is calculated. It plays a crucial role in determining the granularity of the numerical approximation and impacts the accuracy and efficiency of the solution. A smaller step length typically leads to a more accurate but computationally expensive solution, while a larger step length may sacrifice accuracy for computational efficiency. The choice of an appropriate step length is a critical consideration in the numerical solution of differential equations.
4. Stiff and Non-Stiff system: In the context of research, J. D. Lambert characterizes stiffness as follows:

When employing a numerical method possessing a finite region of absolute stability on a system with arbitrary initial conditions, if the method necessitates the utilization of an exceptionally small step length within a specific integration interval, relative to the smoothness of the exact solution in that range, then the system is identified as stiff during that interval (Lambert, 1977). A system is considered stiff if it contains components or features that vary widely in terms of their natural frequencies or time scales. Stiff systems often involve rapid and slow modes of response, and the stiffness of the system can lead to numerical challenges in solving the associated differential equations.

It can be deduced that stiffness in a dynamic system refers to the difference in time scales or natural frequencies of its components. Stiff systems

require special consideration in numerical simulations due to the challenges associated with solving the corresponding stiff ODEs. Non-stiff systems, on the other hand, are generally easier to simulate numerically.

5. Algorithms or Packages: These are computer code which implements numerical method, in addition to find the approximate/numerical method, it may perform other task such as estimating the error of a particular method, monitoring and updating the value of the step-length  $h$  and deciding which of the family of methods to employ at a particular stage in the solution (Lambert, 1977)
6. Collocation method: is a numerical technique used to solve ordinary differential equations, partial differential equations, and integral equations. The method involves selecting a finite-dimensional space of candidate solutions, usually polynomials up to a certain degree, and a number of points in the domain called collocation points. The idea is to select the solution that satisfies the given equation at the collocation points. The method provides high order accuracy and globally continuous differentiable solutions. (Wikipedia contributors, 2023)

## **0.2 Chapter 2: Literature Review**

### **0.2.1 Introduction**

One of the more difficult classes of problems in numerical computation are solution of stiff equation and stiff systems. These problems arise from a variety of physical situations, but were probably first identified in chemical kinetics.

### **0.2.2 Stiff BVPs in Scientific and Engineering Applications**

Explore examples and applications of stiff BVPs in various scientific and engineering disciplines.

### **0.2.3 Challenges in Numerical Solution**

Discuss the challenges posed by stiff BVPs and the limitations of existing numerical methods.

### **0.2.4 Collocation Methods**

Review the strengths and applications of collocation methods in capturing local behavior.

### **0.2.5 Multistep Methods**

Review the strengths and applications of multistep methods in handling temporal evolution.

## **0.3 Chapter 3: Methodology**

### **0.3.1 Implementation of Collocation and Multistep Methods**

Describe the process of implementing collocation and multistep methods in a programming language (e.g., Python).

### **0.3.2 Development of Unified Numerical Code**

Outline the steps involved in developing a comprehensive numerical code for solving stiff BVPs.

### **0.3.3 Parametric Studies**

Detail the plan for conducting parametric studies to evaluate the code's performance under various conditions.

### **0.3.4 Documentation**

Discuss the approach to documenting the code, including explanations, examples, and guidelines for users.

### **0.3.5 References**

Include relevant references to literature, articles, and resources that support the background, challenges, and methodologies discussed.

# Chapter 1

## Literature Review

### 1.1 Introduction

Exploring existing knowledge and methodologies for the numerical solution of stiff systems of Boundary Value Problems (BVPs) is pivotal for developing a unified numerical code. The introduction emphasizes the unique challenges posed by stiff systems and the necessity for accurate and efficient numerical methods.

### 1.2 Fundamental Concepts in Numerical Methods

#### 1.2.1 Overview

Provide an overview of foundational concepts in numerical methods, such as finite difference methods, finite element methods, and spectral methods. Establish a foundation for understanding numerical approaches to solving BVPs.

### 1.3 Collocation Methods

#### 1.3.1 Current Applications

Delve into existing literature on collocation methods, exploring studies that have employed collocation techniques for solving differential equations and BVPs. Discuss key methodologies, advancements, and applications of collocation methods.

## **1.4 Multistep Methods**

### **1.4.1 Strengths and Limitations**

Examine studies related to multistep methods, focusing on their strengths and limitations in handling stiff systems of BVPs. Discuss algorithms, applications, and comparative analyses of multistep methods.

## **1.5 Integration of Collocation and Multistep Methods**

### **1.5.1 Unified Approaches**

Explore literature proposing the integration of collocation and multistep methods. Review studies that have attempted to unify these approaches for the efficient solution of stiff BVPs, highlighting any novel algorithms or methodologies.

## **1.6 Python in Scientific Computing**

### **1.6.1 Role of Python**

Discuss the role of Python in scientific computing, emphasizing its popularity, readability, and suitability for numerical analysis. Explore studies and projects that leverage Python for solving differential equations, establishing the context for the choice of programming language.

## **1.7 Previous Approaches to Stiff BVPs**

### **1.7.1 Methodologies and Tools**

Examine existing methodologies and computational tools used in the solution of stiff BVPs. Review notable studies employing various numerical methods and software for addressing stiff systems.

## **1.8 Challenges and Gaps in the Literature**

### **1.8.1 Identifying Needs**

Identify challenges faced by existing approaches and highlight gaps in the literature. Outline areas where further research is needed and discuss limitations or drawbacks of current methodologies for solving stiff BVPs.



## 1.9 Summary

Summarize key findings from the literature review, synthesizing the current state of knowledge in solving stiff BVPs using numerical methods. Provide a bridge to the subsequent chapters of the research project.

# References

- Byrne, G. D., Hindmarsh, A. C., Jackson, K. R., & Brown, H. G. (1977). A comparison of two ode codes: Gear and episode. *Computers & Chemical Engineering*, 1(2), 125–131. Retrieved from <https://www.sciencedirect.com/science/article/pii/0098135477800185> doi: 10.1016/0098-1354(77)80018-5
- Enright, W. H., Hull, T. E., & Lindberg, B. (1975). Comparing numerical methods for stiff systems of o.d.e.s. *BIT Numerical Mathematics*, 15(1), 10–48. Retrieved from <https://doi.org/10.1007/BF01932994> doi: 10.1007/BF01932994
- Lambert, J. D. (1977). The initial value problem for ordinary differential equations. In D. Jacobs (Ed.), *The state of the art in numerical analysis* (pp. 451–501). New York: Academic Press.
- Wikipedia contributors. (2023). *Collocation method* — *Wikipedia, the free encyclopedia*. Retrieved from [https://en.wikipedia.org/w/index.php?title=Collocation\\_method&oldid=1166346639](https://en.wikipedia.org/w/index.php?title=Collocation_method&oldid=1166346639) ([Online; accessed 14-January-2024])
- Yatim, S. A. M., Ibrahim, Z. B., Othman, K. I., & Suleiman, M. B. (2013). A numerical algorithm for solving stiff ordinary differential equations. *Mathematical Problems in Engineering*, 2013, Article ID 989381, 11. Retrieved from <https://doi.org/10.1155/2013/989381> doi: 10.1155/2013/989381