Exponentially Fitted Difference Method for Solving Second Order Singularly Perturbed Volterra Integro-Differential Equations

Yidnekachew Abebe Wondmu

A Thesis Submitted to the Department of Applied Mathematics
College of Applied Natural Sciences
Presented in Partial Fulfillment of the Requirement for the Degree of
Master's in Applied Mathematics(Numerical Analysis)

Office of Graduate Studies
Adama Science and Technology University

June, 2025
Adama, Ethiopia

Exponentially Fitted Difference Method for Solving Second Order Singularly Perturbed Volterra Integro-Differential Equations

Yidnekachew Abebe Wondmu

Major-Advisor: Mesfin Mekuria Woldaregay (Ph.D)
Co-Advisor: Tekle Gemechu Dinka (Ph.D)

A Thesis Submitted to the Department of Applied Mathematics
College of Applied Natural Sciences

Presented in Partial Fulfillment of the Requirement for the Degree of
Master's in Applied Mathematics (Numerical Analysis)

Office of Graduate Studies
Adama Science and Technology University

June, 2025
Adama, Ethiopia

Declaration

I hereby declare that this Master's Thesis entitled **"Exponentially Fitted Difference Method for Solving Second Order Singularly Perturbed Volterra Integro-Differential Equations"** is my original work. That is, it has not been submitted for the award of an academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Name of Student:	**Signature:**	**Date:**

Recommendation of Advisors

We, the advisor(s) of this thesis, hereby certify that we have read the revised version of the thesis entitled **"Exponentially Fitted Difference Method for Solving Second Order Singularly Perturbed Volterra Integro-Differential Equations"** prepared under our guidance by **Yidnekachew Abebe Wondmu** submitted in partial fulfillment of the requirements for the degree of Master's of Science in Applied Mathematics. Therefore, we recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Major Advisor:	Signature:	Date:
Co-advisor:	Signature:	Date:

Approval Page

We, the advisor of the thesis entitled **"Exponentially Fitted Difference Method for Solving Second Order Singularly Perturbed Volterra Integro-Differential Equations"** by **Yidnekachew Abebe Wondmu**, hereby certify that the recommendations and suggetions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Major Advisor:	Signature:	Date:
Co-advisor:	Signature:	Date:

We, the undersigned, members of the Board of Examiners of the thesis by Yidnekachew Abebe Wondmu, have read and evaluated the thesis proposal entitled **"Exponentially fitted difference method for solving second order singularly perturbed Volterra Integro-Differential equations"** and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement for the degree of Master of Science in Applied Mathematics (Numerical Analysis).

Chairperson:	Signature:	Date:
Internal Examiner:	Signature:	Date:
External Examiner:	Signature:	Date:

Final approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head :	Signature:	Date:
School Dean:	Signature :	Date:
Office of Postgraduate Studies Dean:	Signature:	Date:

ACKNOWLEDGEMENT

I wish to extend my utmost gratitude to God for his continuous support, blessings, and guidance throughout this journey of mine. I'm especially in debt to my advisor, Dr. Mesfin Mekuria, whose assistance, keen insights, and support have been immensely valuable. I extend my sincere appreciation to my co-advisor, Dr. Tekle Gemechu, for his unwavering guidance and motivation during this work.

A warm thank you goes out to my mother, Betelhem Abera; her unconditional support, understanding, and faith in me have given me the ability to overcome every obstacle. I also wish to express my appreciation towards the faculty of Applied Mathematics and Adama Science and Technology University for allowing me to pursue my studies.

Contents

Declaration i

Recommendation of Advisors ii

Approval Page iii

Acknowledgement iv

Table of Contents v

List of Tables vii

List of Figures viii

List of Acronyms ix

Abstract x

1 Introduction 1
- 1.1 Background of the Study . . . 1
- 1.2 Introduction to Volterra Integral Equations . . . 3
- 1.3 Statement of the Problem . . . 5
- 1.4 Objectives of the Study . . . 6
  - 1.4.1 General Objective . . . 6
  - 1.4.2 Specific Objectives . . . 6
- 1.5 Significance of Study . . . 6
- 1.6 Delimitation of Study . . . 7

2 Literature Review 8

3 Methodology 12
- 3.1 Study Design . . . 12
- 3.2 Mathematical Procedures . . . 12

4 **Discripiton and Analysis of the Method** **13**
4.1 Properties of Continuous Problem 13
4.2 The Fitting Parameter and Discretization 17
4.2.1 The Fitting Parameter 17
4.2.2 Discretization of the Differential Terms 20
4.2.3 Composite Trapezoidal Rule for Integral Approximation 21
4.3 Stability analysis of the scheme 25
4.4 Uniform Convergence of the Scheme 32

5 **Result and Discussions** **36**

Conclusion **45**

References **47**

List of Tables

5.1 Maximum absolute error of the exponentially finite difference method for Example 5.1. 40
5.2 Convergence rate of the exponentially finite difference method applied to Example 5.1. 40
5.3 Maximum absolute error of the exponentially finite difference method for Example 5.2. 40
5.4 Convergence rate of the exponentially finite difference method for Example 5.2. 40
5.5 Comparison of maximum point-wise absolute errors using the proposed method and the method without the fitting parameter for Example 5.1 . 41
5.6 Comparison of rates of convergence using the proposed method and the method without the fitting parameter for Example 5.1 41
5.7 Comparison of maximum point-wise absolute errors using the proposed method and the method without the fitting parameter for Example 5.2. . 42
5.8 Comparison of rates of convergence using the proposed method and the method without the fitting parameter for Example 5.2. 42
5.9 Comparison of maximum absolute errors between the Hybrid scheme and EFDM for Example 5.1. 43
5.10 Comparison of convergence rates between the Hybrid scheme in and EFDM for Example 5.1. 43

List of Figures

5.1 Solution behavior of Example 5.1 with the boundary layer effects as ε diminishes, with N=500. 37

5.2 Point-wise absolute error for Example 5.1 with a fixed mesh size of $N = 500$ as ε decreases. 37

5.3 Point-wise absolute error using the proposed method for Example 5.1 with increasing mesh size N, and fixed $\varepsilon = 2^{-05}$. 38

5.4 Solution behavior for Example 5.2 with the boundary layer effects as ε diminishes, with N=128. 38

5.5 Point-wise absolute error using the proposed method for Example 5.2 with a fixed mesh size of $N = 128$ as ε decreases. 39

5.6 Point-wise absolute error using the proposed method for Example 5.2 with increasing mesh size N, and fixed $\varepsilon = 10^{-02}$. 39

LIST OF ACRONYMS

EFDM Exponentially Fitted Finite Difference Method

SPP Singularly Perturbed Problem

SPVIDE Singularly Perturbed Volterra Integro-Differential Equation

FDM Finite Difference Method

FIDE Fredholm integro-differential equation

IDEs Integro-differential equations

VIDE Volterra integral differential equation

BVP Boundary value problem

ABSTRACT

In this thesis, we develop and analyze a uniformly convergent numerical scheme for solving second-order singularly perturbed linear Volterra integro-differential equations, which are characterized by the presence of a small perturbation parameter ε that leads to the formation of boundary layers in the solution. These boundary layers cause severe challenges for standard numerical methods, as they result in sharp gradients near the boundaries that are difficult to resolve accurately without excessive computational effort. To address these challenges, we propose a robust parameter-uniform numerical method that effectively captures the behavior of the solution across different scales. The differential part of the equation is discretized using an exponentially fitted difference scheme, which is specifically designed to handle the exponential nature of the boundary layer solutions by incorporating the perturbation parameter into the finite difference operator. This ensures that the numerical method remains stable and accurate even on coarse meshes. Meanwhile, the integral part of the equation is approximated using the Composite Trapezoidal Rule, which provides second-order accuracy for smooth integrands and maintains consistency with the overall discretization strategy. A rigorous stability and convergence analysis is conducted to establish that the proposed scheme is uniformly convergent with respect to the perturbation parameter ε, meaning that the accuracy of the numerical solution does not deteriorate as ε approaches zero. To validate the theoretical results and demonstrate the effectiveness of the method, two test examples with known exact or highly accurate approximate solutions are considered. The maximum absolute errors and rates of convergence are computed for various values of ε and different mesh sizes, and the results are presented in tabular form. These numerical experiments confirm that the proposed scheme achieves ε-uniform convergence and efficiently resolves the sharp boundary layers, thereby illustrating its reliability and superiority over conventional methods that fail to maintain accuracy in the presence of strong boundary layer effects.

Keywords: Singular perturbation, Volterra integro-differential equation, Exponentially fitted method, Composite trapezoidal Rule, Numerical method, Boundary layer.

CHAPTER ONE
Introduction

1.1 Background of the Study

Differential equations are essential mathematical tools that are widely employed to model dynamic systems and processes in various scientific and engineering disciplines. These equations establish relationships between physical quantities and their rate of change through derivatives, providing a robust framework for analyzing complex phenomena. Their versatility makes them indispensable in fields such as physics, biology, and engineering, where they underpin models for natural and man-made systems.

Building on this foundation, integro-differential equations(IDEs) extends the framework of differential equations by incorporating both differential and integral operators. This combination enables IDEs to tackle more complex problems across diverse applications in physics, biology, and engineering (Manebo et al., 2024). IDEs are classified according to characteristics such as linearity, type of integral operator, and overall structure of the equation (Zimmerman, 2004).

Linear IDEs involve linear expressions of the unknown function and its derivatives. For instance, the general form of a linear IDE is given by:

$$\frac{du(t)}{dt} + p(t)u(t) = \int_a^b K(t,s)u(s)\,ds + f(t), \tag{1.1}$$

where $K(t,s)$ represents the kernel, and $f(t)$ is a given function.

In contrast, nonlinear IDEs introduce additional complexities, as the unknown function $u(t)$, its derivatives, or the kernel $K(t,s)$ may appear in nonlinear forms (Sladek et al., 2005). These equations can further be categorized as Volterra or Fredholm IDEs based on the limits of the integral operator. Volterra integro-differential equations involve integrals with variable upper limits, as :

$$\frac{du(t)}{dt} = \int_a^t K(t,s)u(s)\,ds + f(t), \tag{1.2}$$

where the upper limit is a variable t, and $K(t,s)$ is the kernel. Fredholm integro-differential equation, on the other hand, are characterized by fixed integration limits (Hofmann et al., 2008).

A key component of integro-differential equations is the kernel, which defines the relationship between variables within the integral. Depending on their properties, kernels can be classified into several types:

- Separable Kernel: The kernel can be expressed as a product of functions, $K(t,s) = \varphi(t)\psi(s)$.
- Non-separable Kernel: The kernel cannot be factored into separate functions of t and s.
- Symmetric Kernel: The kernel satisfies $K(t,s) = K(s,t)$.
- Asymmetric Kernel: The kernel does not satisfy symmetry, $K(t,s) \neq K(s,t)$.
- Singular Kernel: Contains singularities (e.g., division by zero) at specific points but remains integrable in the domain.
- Non-singular Kernel: Smooth and well-defined throughout the domain (Messina & Vecchio, 2020).

Furthermore, integro-differential equations can be categorized based on the role of the integral term.

Convolution Equations: The integral term represents a convolution of the unknown function with the kernel.

Non-convolution Equations: The integral term exhibits dependencies other than convolution (Messina & Vecchio, 2020).

This comprehensive classification of integro-differential equations facilitates the selection of appropriate mathematical and numerical methods for solving them. By tailoring techniques to specific types of IDEs, researchers can effectively address complex problems across various scientific domains.

A specialized subset of differential equations is singular perturbation problems (SPPs), which are characterized by the presence of a small positive parameter ε multiplying the highest order derivative term. This small parameter often leads to sharp variations in the solution within narrow regions of the domain, commonly referred to as layers, which can be classified as boundary or interior layers (Roos, 2008). These problems are prevalent in various scientific disciplines, including fluid dynamics, quantum mechanics, geophysical systems, reaction-diffusion processes, and mathematical genetics (Prandtl, 1905). For instance, in fluid dynamics, the two-dimensional Navier-Stokes equations exhibit singular perturbation behavior for large Reynolds numbers, highlighting the analytical challenges posed by such systems. Similarly, in mathematical genetics, singular perturbation problems appear in the time-independent Fokker-Planck equation, underscoring their cross-disciplinary relevance.

SPPs are a type of differential equation, either ordinary or partial, exhibiting a small perturbation parameter ε resulting in a steep change in the solutions or derivatives, i.e., boundary layers. As $\varepsilon \to 0$, the solutions could converge to discontinuous limits, making solutions completely distinct from ordinary differential equations, which may allow smooth asymptotic expansions into powers of ε (Roos, 2008). The introduction of this small perturbation parameter induces stiffness and leads to steep variation in a very small region, which creates numerical instability in classical numerical techniques. Despite extensive research efforts towards analytical, semi-analytical, and numerical studies of SPPs, solving conventional, or even simply accurate, SPP solutions is still an issue.

In order to deal with these problems, Babu et al. (2024) ε-uniform convergence of difference schemes has been developed primarily through two primary methods; Fitted Operator Finite Difference Method (FOFDM), and Fitted Mesh Finite Difference Method (FMFDM). Both methods aim to fit finite difference methods to the specific characteristics of the singularly perturbed problem, but these approaches are conceptually very different. FOFDM modifies the differential operator in itself as a means to accurately approximate what is happening with a solution near the layers, while making it a more efficient scheme needed for relatively easy problems. FMFDM modifies the mesh, i.e., the computational grid (typically by refining the grid in the regions where there are steep gradients), attempting to honor the boundary layer with some degree of success, and providing a more efficient method with complicated geometries, or multi-dimensional domains.

Lastly, hybrid approaches, or methods that include principles of both FOFDM and FMFDM, are also considered that promote accuracy and robustness; the final method will depend upon the type of singularly perturbed problem, accuracy required, and expediency of computational resources.

1.2 Introduction to Volterra Integral Equations

Volterra integral equations are a class of functional equations, where the unknown function appears under the integral sign and the upper limit of the integral is a variable. This characteristic distinguishes Volterra equations from other integral equations like Fredholm equations and Volterra-Fredholm equations, where the limits of integration are constant (Wazwaz, 2015). Volterra equations are widely used in various scientific fields such as demography, biology, chemistry, and insurance mathematics, owing to their ability to model systems where the present state depends on both the current and past values of the function.

In the context of Volterra equations, there are two primary forms: the first kind and the second kind. For a closed and bounded interval $I := [0, T]$, Volterra integral equations of the first and second kinds in the unknown function $u(t)$ are given by Burton (2005):

$$\int_{t_0}^{t} K(t,s,u(s))\,ds = f(t) \quad \text{(First Kind)} \tag{1.3}$$

and

$$u(t) = f(t) + \int_{t_0}^{t} K(t,s,u(s))\,ds \quad \text{(Second Kind)} \tag{1.4}$$

where $f(t)$ is a known function, $K(t,s,u(s))$ is the kernel, and t_0 is a constant. The key distinction between the two types of equations lies in the explicit appearance of the unknown function outside the integral in the second-kind equation, while it appears only inside the integral in the first-kind equation. Notably, when the limits of integration are constant, these equations become **Fredholm integral equations**, which are conceptually similar to boundary value problems in ordinary differential equations (ODEs) (Moore, 1979).

Volterra integral equations (VIEs), particularly in their second-kind formulation, form the basis for more complex models in various disciplines. For example, Volterra integro differential equations combine both differential and integral operators, providing a powerful tool for analyzing systems where the current state is influenced by both its derivative and cumulative past effects. The general form of a Volterra integro-differential equation is:

$$u^{(n)}(t) = f(t) + \int_{t_0}^{t} K(t,s,u(s))\,ds \tag{1.5}$$

where $u^{(n)}(t) = \frac{d^n u(t)}{dt^n}$ represents the n-th derivative of the unknown function $u(t)$, and the initial conditions $u(t_0), u'(t_0), \ldots, u^{(n-1)}(t_0)$ are given. If the limits of the integral are constants, this equation becomes a Fredholm integro differential equations (Ebrahimi & Rashidinia, 2014). And also the Volterra-Fredholm integro-differential equations arise from parabolic boundary value problems.

The Volterra-Fredholm integro-differential equations appear in two types, namely.

$$u^{(n)}(t) = f(t) + \lambda_1 \int_{0}^{t} K_1(t,s)u(s)\,ds + \lambda_2 \int_{a}^{b} K_2(t,s)u(s)\,ds$$

and the mixed form

$$u^{(n)}(t) = f(t) + \lambda_1 \int_{0}^{t} \int_{a}^{b} K(t,s)u(s)\,ds$$

The first type contains split integrals, and the second type contains mixed integrals such that the Fredholm integral is the inferior one, and Volterra is the exterior integral(Zhang, Zhang, & Gan, 2024).

The linearity or non-linearity of Volterra equations significantly affects the analytical and numerical methods used for solving them. Specifically: - If kernel $K(t,s,v) = K(t,s)v$, the equation is linear. - If the kernel $K(t,s,v)$ becomes unbounded at some points or if one of the integration limits is infinite, the equation is classified as singular, which requires more specialized methods for solution.

For singularly perturbed Volterra integro-differential equation (SPVIDE), solutions may exhibit steep gradients or sharp transitions over narrow regions of the domain, known as boundary or interior layers. This behavior is typical when small perturbation parameters, such as ε, influence the system's dynamics. A common form of such an equation is as follows:

$$\varepsilon u''(t) + a(t)u'(t) - b(t)u(t) - \lambda \int_0^t K(t,s)u(s)\,ds = f(t), \tag{1.6}$$

where ε is a small perturbation parameter. These equations can be linked to the corresponding reduced Volterra integro-differential equation by using special conditions. However, solving such equations often requires the development of tailored discretization techniques to handle the singular perturbations effectively (Kumar et al., 2024).

In conclusion, Volterra integral equations, particularly in their integro-differential form, are powerful mathematical tools that model complex systems influenced by both their history and future rates of change. An understanding of their structure, linearity, and singularities is essential for selecting appropriate numerical methods for their solution.

1.3 Statement of the Problem

It is well recognized that when the perturbation parameter ε is small, conventional numerical methods often fail to deliver stable and accurate solutions for singularly perturbed problems. This makes it essential to design numerical schemes whose performance does not deteriorate as ε decreases specifically, methods that exhibit ε-uniform convergence (Kadalbajoo et al., 2006).

Ensuring both accuracy and convergence is crucial because these types of problems are inherently challenging. The solution behavior is strongly influenced by the size of the perturbation parameter as well as the mesh size used in computation (Kadalbajoo & Gupta, 2010). Traditional numerical approaches tend to produce unstable or oscillatory results when ε is very small, leading to unreliable outputs or even divergence in the solution (Woldaregay & Duressa, 2019). To handle the sharp changes in SPVIDE solutions, classical methods often require a very fine mesh.

However, this leads to high computational cost and increased risk of round-off errors. Although many numerical techniques have been developed for SPVIDEs, only a few use exponentially fitted finite difference schemes, which are well-suited for capturing rapid solution changes. Similarly, methods based on asymptotic expansions, despite their potential, are rarely applied. This gap motivates our study. To overcome these challenges, we propose an exponentially fitted finite difference scheme combined with a composite trapezoidal integration rule on a uniform mesh. This approach aims to offer a stable and efficient method for solving second-order linear SPVIDEs, especially when the perturbation parameter ε is small.

1.4 Objectives of the Study

1.4.1 General Objective

The main goal of this study is to develop an accurate and uniformly convergent numerical method for solving the Second-Order Linear Singularly Perturbed Volterra Integro-Differential Equation (SPVIDE).

1.4.2 Specific Objectives

The specific objectives of this study are:

- to construct an exponentially fitted difference method together with a composite trapezoidal rule on the uniform mesh for the numerical solution of second-order linear singularly perturbed Volterra integro-differential equation (SPVIDE).
- to analyze the stability of the method.
- to establish the uniform convergence of the method.

1.5 Significance of Study

The result obtained from this study may:

- serve as reference material for scholars who work in this area.
- provide a numerical method for solving Second-Order Linear Singularly Perturbed Volterra Integral-Differential Equation (SPVIDE).

1.6 Delimitation of Study

The study is delimited to solving a second-order singularly perturbed Volterra integro-differential equation (SPVIDE), which is given by

$$\varepsilon u''(x) + a(x)u'(x) - b(x)u(x) - \lambda \int_0^x K(x,s)u(s)\,ds = f(x), \quad x \in (0,l), \tag{1.7}$$

with the boundary conditions

$$u(0) = A, \quad u(l) = B, \tag{1.8}$$

where $\varepsilon \in (0,1]$ is a perturbation parameter, λ is a real parameter, and l is some fixed number.

We assume that $K(x,s)$ kernel function of integral term, $u(x)$ unknown funcition, A and B are boundary value at $x = 0$ and $x = l$ respectively, $a(x) \geq 0$, $\forall\, x \in [0,l]$, $f(x)$ and $b(x)$ are sufficiently smooth functions satisfying certain regularity conditions to be specified.

CHAPTER TWO

Literature Review

In this section, we look into various studies from the literature that concentrate on the numerical analysis of singularly perturbed Volterra integro-differential equation (SP-VIDE). These types of equations are important in modeling a range of complex phenomena in engineering and applied sciences. Because of their challenging nature, especially due to the presence of small perturbation parameters, there has been considerable effort by researchers to build reliable theoretical insights and effective numerical techniques for solving them.

Several recent studies have focused on exploring different numerical approaches, such as quadrature techniques, finite difference and finite element schemes, variational iteration, homotopy perturbation, and spline-based collocation methods for solving integral and integro-differential equations of Volterra, Fredholm, Fredholm-Volterra, and Volterra-Abel types (Aigo, 2013; B. C. Iragi & Munyakazi, 2020).

When the aforementioned classes of problems are modified such that the terms involving the highest-order derivatives are scaled by a small parameter, they are referred to as singularly perturbed Volterra, Fredholm, Fredholm-Volterra, and Volterra-Abel integral and integro-differential equations. These equations often exhibit rapid changes within boundary or interior layers, which pose significant challenges to conventional numerical methods. This challenge has led to the development of specialized techniques capable of maintaining stability across all values of the perturbation parameter. Considerable progress has been made in this area, particularly in the study of singularly perturbed Volterra integro-differential equations (B. Iragi, 2017).

Cakir and Gunes (2022) proposed a fitted operator finite difference method to solve singularly perturbed Volterra–Fredholm integro-differential equations. They designed the method on a non-uniform mesh, using interpolating quadrature rules together with linear basis functions. The authors analyzed its stability and accuracy in the discrete maximum norm. To support their method, they carried out several numerical experiments, which showed that the approach performs well and seems quite reliable. Based on the results, the method turns out to be first-order accurate.

Aigo (2013) developed a numerical method for solving second-kind Volterra integral equations by applying repeated Simpson's and trapezoidal quadrature rules. The paper introduced a recurrence relation, and through it, an approximate solution was achieved, showing fairly good convergence.

B. C. Iragi and Munyakazi (2020) introduced a finite difference approach to tackle a class of singularly perturbed Volterra integro-differential equation. Their method was built using integral identities, along with carefully chosen quadrature rules that include weights and remainder terms in integral form. They applied an Euler-type finite difference operator over a piecewise-uniform Shishkin mesh. In their study, they demonstrated that the scheme achieves first-order uniform convergence concerning both the perturbation parameter and the mesh size.

Erdoğan and Amiraliyev (2012) looked into a singularly perturbed initial value problem involving a second-order linear delay differential equation. To handle the problem, they applied an exponentially fitted difference method on a uniform mesh. In the end, their results showed that the scheme was uniformly convergent with respect to both the small perturbation parameter and the mesh size.

Liu et al. (2024) came up with a second-order finite difference method on a graded mesh to solve VIDE, which involves a weakly singular kernel. They used the two-step backward differentiation formula (BDF2) to handle the first derivative and applied a basic first-order interpolation approach to approximate the integral part. They also performed a stability analysis, which helped prove that the method converges under the discrete maximum norm.

Panda et al. (2021) developed a post-processing method to tackle a SPVIDE, using a piecewise uniform Shishkin mesh. They also applied a fitted mesh finite difference scheme, where the integral term was handled with a composite trapezoidal rule, and the derivative part was approximated using a finite difference operator. The approach manages to achieve a uniform reference operator specifically for the derivative component.

Çimen and Yatar (2020) worked on an initial value problem involving a first-order linear Volterra delay integro-differential equation. To find an approximate solution, they came up with a new difference method based on a fitted finite difference approach over a uniform mesh. The scheme was developed using the method of integral identities and makes use of exponential basis functions, along with interpolating quadrature formulas that include a remainder term in integral form. They also showed that the method achieves first-order convergence in the discrete maximum norm.

Hadi (2023) studied a generalized nonlinear Volterra integral equation of the second kind, where the kernel depends on the unknown function evaluated at two different arguments. To tackle this, the author introduced a collocation method that uses piecewise Hermite polynomials. The approach involved Hermite polynomials of degree $2m-1$, with m being a natural number, and applied appropriate quadrature rules. This setup led to an accuracy of order $2m$. It was also pointed out that choosing the right collocation points is crucial, as they must meet a stability condition to guarantee a reliable numerical result.

Salama and Evans (2001) developed an exponentially fitted numerical scheme, demonstrating fourth-order accuracy for fixed perturbation parameter ε, and supported their method with a stability analysis. Lodge et al. (1978) examined various qualitative properties of nonlinear SPVIDEs and established both existence and uniqueness of the solution. Meanwhile, Şevgin (2014) proposed a uniformly convergent numerical method on a graded mesh, achieving first-order accuracy in the maximum norm. These works collectively highlight the importance of tailored numerical strategies for SPVIDEs, especially in handling boundary layers and ensuring stability and convergence.

Angell and Olmstead (1987) introduced a formal method for deriving asymptotic expansions of solutions to singularly perturbed Volterra equations and illustrated the approach through several examples. In a related study, Angell and Olmstead (1985) applied a formal asymptotic technique to analyze the leading-order behavior of a SPVIDE that describes the stretching of a polymer filament. Meanwhile, Bijura (2006) identified the presence of initial layers whose thickness does not follow the typical order of $O(\varepsilon)$ as $\varepsilon \to 0$, and constructed approximate solutions using initial layer theory.

Koto (2002) looked into the stability of a Runge-Kutta method applied to Volterra delay integro-differential equations with a constant delay. When it comes to singularly perturbed cases, they also discussed the challenges that come up in numerical computations and pointed to various strategies for handling them, drawing on the techniques explored in (Miranker, 2001). In a related work, Wu and Gan (2009) analyzed a SPVIDE with delay. They used a linear multistep method to study the behavior of the numerical errors and provided global error estimates for an A(α)-stable multistep method that incorporates a convergent quadrature rule.

Kudu et al. (2016) proposed an implicit finite difference method based on a piecewise-uniform Shishkin-type mesh for tackling a singularly perturbed delay integro-differential equation. The method followed the approach described in (Amiraliyev & Şevgin, 2006). They demonstrated that, under suitable conditions, the method is stable with respect to ε and achieves a convergence rate of $O\left(N^{-1}\ln N\right)$.

Priyadarshana et al. (2025)Robust adaptive grid-based moving mesh algorithms are discussed for singularly perturbed second-order convection-diffusion type VIDEs. Two different kinds of monitor functions are used to establish the numerical schemes. A well-studied first-order monitor function is used to construct Scheme I, which is a combination of the upwind scheme and the left rectangle rule of first-order accuracy. Further, the use of a comparatively new second-order monitor function is examined to elevate the accuracy of Scheme-I via the Richardson extrapolation technique.

Erdoğan (2023) studies SPVIDEs that involve a delay term. Taking into account the characteristics of the exact solution, the authors developed a hybrid finite difference scheme, applying suitable quadrature rules on a Shishkin-type mesh. Using truncation error analysis along with a discrete version of Grönwall's inequality, they showed that the proposed scheme achieves near second-order accuracy in the discrete maximum norm.

Zhongdi and Lifeng (2006) explored the convergence behavior of a finite difference method on a Shishkin mesh, using the midpoint difference operator along with trapezoidal integration. They managed to derive an a priori error estimate that remains uniform with respect to ε, and showed that the method reaches close to second-order accuracy. In contrast, Amiraliyev and Şevgin (2006) approached the same problem with a fitted finite difference technique. Their method incorporated a fitting factor introduced through the method of integral identities, using exponential basis functions and specially weighted quadrature rules that included remainder terms in integral form. They proved that this approach achieves first-order uniform convergence, in the maximum norm, with respect to ε.

To the best of our knowledge, previous works on SPVIDE have not utilized exponentially fitted finite difference methods for their numerical solutions. Additionally, the application of exponentially fitted finite difference methods derived from asymptotic approaches has not been thoroughly investigated for solving SPVIDE. In this study, we address a second order linear SPVIDE.

CHAPTER THREE
Methodology

3.1 Study Design

The study applies both the documentation review and numerical experimentation.

3.2 Mathematical Procedures

To attain the study's objective, we follow the following steps.

1. Domain discretization of the problem.
2. Construction of an exponentially fitted finite difference method for the problem.
3. Assembling a system of equations.
4. Analyze the stability and convergence of the method.
5. Writing MATLAB code for the proposed scheme.
6. Validating the method using numerical examples.

By following this methodology, the research effectively solves the singularly perturbed linear Volterra integro-differential equations, ensuring the numerical solution's stability and accuracy.

CHAPTER FOUR

Discripiton and Analysis of the Method

4.1 Properties of Continuous Problem

In this part, we take a closer look at the qualitative behavior of the solution to equations (1.7)–(1.8), along with its derivatives. Understanding these properties is essential for analyzing the convergence of the numerical methods we're using.

Lemma 4.1. *For the boundary value problem given in (1.7)-(1.8), the following bounds apply to the solution $u(x)$:*

$$\|u\|_{\infty,I} \leq C_o, \tag{4.1}$$

and for the k^{th} derivatives of u(x), with $k = 1, 2, 3, 4$ and $x \in [0, l]$,we have

$$|u^{(k)}(x)| \leq C\left(1 + \varepsilon^{-k} e^{-\frac{\alpha x}{\varepsilon}}\right), \tag{4.2}$$

where $C_o = (|u_0| + |u_1| + \alpha^{-1}q)\, e^{\alpha^{-1}\beta l}$.

Here, $\alpha = \min_{x\in I} a(x) > 0$, $\beta = \max_{x\in I} b(x) > 0$, and $q(x) = f(x) + \lambda \int_0^x K(x, s)u(s)\, ds$.

Proof. Let

$$\varepsilon u''(x) + a(x)u'(x) - b(x)u(x) - \lambda \int_0^x K(x, s)u(s)\, ds = f(x),$$

Now, consider $u'(x) = v(x)$, then:

$$\varepsilon v'(x) + a(x)v(x) = F(x) \tag{4.3}$$

where

$$q(x) = f(x) + \lambda \int_0^x K(x, s)u(s)\, ds.$$

$$F(x) = q(x) + b(x)u(x)$$

By solving (4.3), we can get

$$u'(x)e^{\int_0^x \frac{a(\delta)}{\varepsilon} d\delta} = u'(0) + \frac{1}{\varepsilon}\int_0^x F(\zeta)e^{\int_0^\zeta \frac{a(\delta)}{\varepsilon} d\delta} d\zeta.$$

On simplifying, it yields.

$$u'(x) = u'(0)e^{-\frac{1}{\varepsilon}\int_0^x a(\delta)d\delta} + \frac{1}{\varepsilon}\int_0^x F(\zeta)e^{-\frac{1}{\varepsilon}\int_\zeta^x a(\delta)d\delta}d\zeta. \tag{4.4}$$

By integrating once more over the integral $(0, x)$, we arrive at the following expression

$$u(x) = u(0) + \int_0^x u'(0)e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu + \frac{1}{\varepsilon}\int_0^x d\nu \int_0^\nu F(\zeta)e^{-\frac{1}{\varepsilon}\int_\zeta^\nu a(\delta)d\delta}d\zeta.$$

After adjusting the limits of integration,

$$u(x) = u_0 + u'(0)\int_0^x e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu + \frac{1}{\varepsilon}\int_0^x F(\zeta)d\zeta \int_\zeta^x e^{-\frac{1}{\varepsilon}\int_\zeta^\nu a(\delta)d\delta}d\nu. \tag{4.5}$$

Applying $u(l) = u_1$ in (4.5), we have

$$u'(0) = \frac{u_1 - u_0 - \frac{1}{\varepsilon}\int_0^l F(\zeta)d\zeta \int_\zeta^l e^{-\frac{1}{\varepsilon}\int_\zeta^\nu a(\delta)d\delta}d\nu}{\int_0^l e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu}. \tag{4.6}$$

Using (4.6) in (4.5), we get

$$\begin{aligned}
u(x) &= u_0 + \left(\frac{u_1 - u_0 - \frac{1}{\varepsilon}\int_0^l F(\zeta)d\zeta \int_\zeta^l e^{-\frac{1}{\varepsilon}\int_\zeta^\nu a(\delta)d\delta}d\nu}{\int_0^l e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu}\right)\int_0^x e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu \\
&\quad + \frac{1}{\varepsilon}\int_0^x F(\zeta)d\zeta \int_\zeta^x e^{-\frac{1}{\varepsilon}\int_\zeta^\nu a(\delta)d\delta}d\nu \\
&= u_0\left(1 - \frac{\int_0^x e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu}{\int_0^l e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu}\right) + u_1\left(\frac{\int_0^x e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu}{\int_0^l e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu}\right) \\
&\quad - \frac{1}{\varepsilon}\int_0^l F(\zeta)d\zeta \int_\zeta^l e^{-\frac{1}{\varepsilon}\int_\zeta^\nu a(\delta)d\delta}d\nu \left(\frac{\int_0^x e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu}{\int_0^l e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)d\delta}d\nu}\right) \\
&\quad + \frac{1}{\varepsilon}\int_0^x F(\zeta)d\zeta \int_\zeta^x e^{-\frac{1}{\varepsilon}\int_\zeta^\nu a(\delta)d\delta}d\nu
\end{aligned}$$

Using the estimate $a(\delta) \geq \alpha > 0$, we can write:

$$u(x) \leq u_0\left(1 - \frac{\int_0^x e^{-\frac{\alpha\nu}{\varepsilon}}d\nu}{\int_0^l e^{-\frac{\alpha\nu}{\varepsilon}}d\nu}\right) + u_1\left(\frac{\int_0^x e^{-\frac{\alpha\nu}{\varepsilon}}d\nu}{\int_0^l e^{-\frac{\alpha\nu}{\varepsilon}}d\nu}\right) + \frac{1}{\varepsilon}\int_0^x F(\zeta)d\zeta \int_\zeta^x e^{-\frac{\alpha(\nu-\zeta)}{\varepsilon}}d\nu$$

$$-\frac{1}{\varepsilon}\left[\int_0^l F(\zeta)d\zeta \int_\zeta^l e^{-\frac{\alpha(\nu-\zeta)}{\varepsilon}}d\nu \left(\frac{\int_0^x e^{-\frac{\alpha\nu}{\varepsilon}}d\nu}{\int_0^l e^{-\frac{\alpha\nu}{\varepsilon}}d\nu}\right)\right].$$

Now, compute the integrals:

$$\int_0^x e^{-\frac{\alpha\nu}{\varepsilon}}d\nu = \frac{-\varepsilon}{\alpha}\left(e^{-\frac{\alpha x}{\varepsilon}} - 1\right), \quad \int_0^l e^{-\frac{\alpha\nu}{\varepsilon}}d\nu = \frac{-\varepsilon}{\alpha}\left(e^{-\frac{\alpha l}{\varepsilon}} - 1\right),$$

$$\int_\zeta^x e^{-\frac{\alpha(\nu-\zeta)}{\varepsilon}} = \frac{-\varepsilon}{\alpha}\left(e^{-\frac{\alpha(x-\zeta)}{\varepsilon}}-1\right), \quad \int_\zeta^l e^{-\frac{\alpha(\nu-\zeta)}{\varepsilon}} d\nu = \frac{-\varepsilon}{\alpha}\left(e^{-\frac{\alpha(l-\zeta)}{\varepsilon}}-1\right)$$

So we have the estimate:

$$u(x) \le u_0\left(1-\frac{e^{-\frac{\alpha x}{\varepsilon}}-1}{e^{-\frac{\alpha l}{\varepsilon}}-1}\right) + u_1\left(\frac{e^{-\frac{\alpha x}{\varepsilon}}-1}{e^{-\frac{\alpha l}{\varepsilon}}-1}\right) - \frac{1}{\alpha}\left(\int_0^x e^{-\frac{\alpha(x-\zeta)}{\varepsilon}}-1\right)F(\zeta)d\zeta$$

$$+\frac{1}{\alpha}\int_0^l \left(e^{-\frac{\alpha(l-\zeta)}{\varepsilon}}-1\right)\left(\frac{e^{-\frac{\alpha x}{\varepsilon}}-1}{e^{-\frac{\alpha l}{\varepsilon}}-1}\right)F(\zeta)d\zeta.$$

Taking $F(\zeta) = q(\zeta) + b(\zeta)u(\zeta)$, we have

$$\begin{aligned}
|u(x)| \le |u_0| + |u_1| - \alpha^{-1}\int_0^x &\left(e^{-\frac{\alpha(x-\zeta)}{\varepsilon}}-1\right)|q(\zeta)|\,d\zeta \\
&- \alpha^{-1}\int_0^x \left(e^{-\frac{\alpha(x-\zeta)}{\varepsilon}}-1\right)|b(\zeta)||u(\zeta)|\,d\zeta \\
&+ \alpha^{-1}\int_0^l \left(e^{-\frac{\alpha(l-\zeta)}{\varepsilon}}-1\right)\left(\frac{e^{-\frac{\alpha x}{\varepsilon}}-1}{e^{-\frac{\alpha l}{\varepsilon}}-1}\right)|q(\zeta)|\,d\zeta \\
&+ \alpha^{-1}\int_0^l \left(e^{-\frac{\alpha(l-\zeta)}{\varepsilon}}-1\right)\left(\frac{e^{-\frac{\alpha x}{\varepsilon}}-1}{e^{-\frac{\alpha l}{\varepsilon}}-1}\right)|b(\zeta)||u(\zeta)|\,d\zeta.
\end{aligned}$$

Now using bounds:

$$|u(x)| \le |u_0| + |u_1| + \alpha^{-1}\|q\| + \alpha^{-1}\|b(\zeta)\|\int_0^x |u(\zeta)|\,d\zeta.$$

Using the bound $\|b(\zeta)\| \le \beta$, we can write

$$|u(x)| \le |u_0| + |u_1| + \alpha^{-1}\|q\| + \alpha^{-1}\beta\int_0^x |u(s)|\,ds.$$

Now, by applying Gronwall's inequality, we end up with:

$$|u(x)| \le \left(|u_0| + |u_1| + \alpha^{-1}\|q\|\right)e^{\frac{\beta l}{\alpha}}. \tag{4.7}$$

This proves inequality (4.1).

Now, consider (4.4) and (4.6) for determining the bounds in (4.6), we have

$$\int_0^l e^{-\frac{1}{\varepsilon}\int_0^\nu a(\delta)\,d\delta}d\nu \le \int_0^l e^{-\frac{\alpha\nu}{\varepsilon}}d\nu = \frac{-\varepsilon}{\alpha}\left(e^{-\frac{\alpha l}{\varepsilon}}-1\right) \le C_1\varepsilon, \tag{4.8}$$

and

$$\begin{aligned}
\frac{1}{\varepsilon}\int_0^l F(\zeta)d\zeta\int_\zeta^l e^{-\frac{1}{\varepsilon}\int_\zeta^\nu a(\delta)\,d\delta}d\nu &\le \frac{1}{\varepsilon}\int_0^l F(\zeta)d\zeta\int_\zeta^l e^{-\frac{\alpha(\nu-\zeta)}{\varepsilon}}d\nu \\
&\le \frac{1}{\varepsilon}\int_0^l F(\zeta)d\zeta\cdot\left[\frac{\varepsilon}{\alpha}\left(1-e^{-\frac{\alpha(l-\zeta)}{\varepsilon}}\right)\right] \\
&\le \alpha^{-1}\int_0^l |F(\zeta)|\,d\zeta
\end{aligned}$$

$$\leq \alpha^{-1}\|q\| + \alpha^{-1}\beta \leq C_2. \tag{4.9}$$

Substituting (4.8) and (4.9) into (4.6), we reach

$$|u'(0)| \leq \frac{|u_1| - |u_0| + C_2}{C_1\varepsilon} = C_1^{-1}\left(\frac{u_1 - u_0 + C_2}{\varepsilon}\right) \leq \frac{C_3}{\varepsilon}. \tag{4.10}$$

Finally, using (4.10) in (4.4), we obtain

$$\begin{aligned}
|u'(x)| &\leq |\frac{C_3}{\varepsilon}e^{-\frac{1}{\varepsilon}\int_0^x a(\delta)\,d\delta} - \frac{1}{\varepsilon}\int_0^x F(\zeta)e^{-\frac{1}{\varepsilon}\int_\zeta^x a(\delta)\,d\delta}\,d\zeta| \\
&\leq \frac{C_3}{\varepsilon}e^{-\frac{\alpha x}{\varepsilon}} + \frac{1}{\varepsilon}\int_0^x |q(\zeta)|e^{-\frac{1}{\varepsilon}\int_\zeta^x |a(\delta)|\,d\delta}\,d\zeta + \frac{\beta}{\varepsilon}\int_0^x \left(\int_0^\zeta |u(s)|\,ds\right)e^{-\frac{\alpha(x-\zeta)}{\varepsilon}}\,d\zeta \\
&\leq \frac{C_3}{\varepsilon}e^{-\frac{\alpha x}{\varepsilon}} + \frac{\|q\|}{\alpha}\left(1 - e^{-\frac{\alpha x}{\varepsilon}}\right) + \frac{C_4\beta}{\alpha}\left(1 - e^{-\frac{\alpha x}{\varepsilon}}\right) \\
&\leq \frac{C_3}{\varepsilon}e^{-\frac{\alpha x}{\varepsilon}} + \alpha^{-1}\left(\|q\| + C_4\beta\right)\left(1 - e^{-\frac{\alpha x}{\varepsilon}}\right) \\
&\leq C\left(1 + \varepsilon^{-1}e^{-\frac{\alpha x}{\varepsilon}}\right),
\end{aligned}$$

The case $k = 1$ in (4.2) has now been proven.

To show that inequality (4.2) holds for $k = 2$, we begin by differentiating equation (4.1), which results in:

$$\varepsilon u'''(x) + a(x)u''(x) = N(x), \quad x \in I, \tag{4.11}$$

where the function $N(x)$ is defined as:

$$\begin{aligned}
N(x) = f'(x) - a'(x)u'(x) + b(x)u'(x) + b'(x)u(x) \\
+ K(x,x)u(x) + \int_0^x \frac{\partial}{\partial x}K(x,s)u(s)\,ds.
\end{aligned}$$

From the previous result in (4.1), we already know:

$$|u''(x)| \leq \frac{C}{\varepsilon^2}.$$

Now, integrating equation (4.11) over the interval $(0, x)$ gives:

$$u''(x) = u''(0)e^{-\frac{1}{\varepsilon}\int_0^x a(\delta)\,d\delta} + \frac{1}{\varepsilon}\int_0^x N(\zeta)e^{-\frac{1}{\varepsilon}\int_\zeta^x a(\delta)\,d\delta}\,d\zeta.$$

To estimate $u''(x)$, we apply the triangle inequality:

$$|u''(x)| \leq \frac{C}{\varepsilon^2}e^{-\frac{\alpha x}{\varepsilon}} + \frac{1}{\varepsilon}\int_0^x |N(\zeta)|\,e^{-\frac{1}{\varepsilon}\int_\zeta^x a(\delta)\,d\delta}\,d\zeta.$$

Assuming the kernel function $K(x,s)u(s)$ and its partial derivatives are continuous, the second term can be bounded as:

$$\frac{1}{\varepsilon}\int_0^x |N(\zeta)|\, e^{-\frac{1}{\varepsilon}\int_\zeta^x a(\delta)\, d\delta}\, d\zeta \le C.$$

Combining the bounds, we arrive at:

$$|u''(x)| \le \varepsilon^{-2} C e^{-\frac{\alpha x}{\varepsilon}} + C,$$

which completes the proof for the case $k = 2$ in inequality (4.2).
By differentiating equation (4.1) further and repeating a similar line of reasoning, we can obtain the bounds for $u^{(3)}(x)$ and $u^{(4)}(x)$ as well. □

4.2 The Fitting Parameter and Discretization

For the domain discretization, we use a uniform mesh as

$$x_0 = 0, \quad x_i = ih, \quad x_N = 1, \quad i = 0, 1, 2, \ldots, N, \quad \text{where } h = \frac{1}{N}.$$

The exponentially fitted difference method will be applied to treat the resulting boundary value problems. We first find the exponential fitting parameter for any linear boundary value problem to use the discretizations.

4.2.1 The Fitting Parameter

To obtain the numerical solution of equation (1.7), we use the technique from the theory of asymptotic methods for solving singularly perturbed boundary value problems (BVPs). Since the boundary layer is on the left side of the domain, from the theory of singular perturbation problems, the zeroth-order asymptotic solution of the singularly perturbed BVP of the form

$$\varepsilon u''(x) + a(x)u'(x) - b(x)u(x) = q(x), \quad x \in (0, l), \tag{4.12}$$

with boundary conditions

$$u(0) = A, \quad u(l) = B, \tag{4.13}$$

is given by

$$u(x) = u_0(x) + (A - u_0(0))\, e^{-\frac{a(0)(1-x)}{\varepsilon}} \tag{4.14}$$

where $u_0(x)$ is the solution of the reduced problem obtained by setting $\varepsilon = 0$.

On the mesh points $x_i = ih$, where $h = \frac{1}{N}$, evaluating the result in equation above at x_i gives:

$$u(ih) = u_0(0) + (A - u_0(0))\, e^{-\frac{a(0)(1-ih)}{\varepsilon}}. \tag{4.15}$$

Let $\rho = \frac{h}{\varepsilon}$. For a mesh function u_i, we introduce the following finite difference operators for a mesh function $\{u_i\}$:

$$\begin{aligned}
D^+ u_i &= \frac{u_{i+1} - u_i}{h}, \quad \text{(Forward difference)} \\
D^- u_i &= \frac{u_i - u_{i-1}}{h}, \quad \text{(Backward difference)} \\
D^0 u_i &= \frac{u_{i+1} - u_{i-1}}{2h}, \quad \text{(Central difference)} \\
D^+ D^- u_i &= \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}, \quad \text{(Second-order central difference)}
\end{aligned}$$

We overcome the effect of the perturbation parameter ε by multiplying an exponential fitting parameter $\sigma(\rho)$ on the diffusion part of the problem. Applying the central finite difference method, the equation becomes:

$$\varepsilon\sigma(\rho) D^+ D^- u(x_i) + a(x_i) D^0 u(x_i) - b(x_i) u(x_i) = q(x_i). \tag{4.16}$$

for convenience, we take $a(x_i) = a_i$, $b(x_i) = b_i$ and $\sigma(\rho) = \sigma$.

$$\left(\frac{\sigma\varepsilon}{h^2} (u_{i+1} - 2u_i + u_{i-1}) + \frac{a_i}{2h} (u_{i+1} - u_{i-1}) \right) = (q(x_i) + b_i u(x_i)) \tag{4.17}$$

Multiplying equation above by h, assuming h is small

$$\lim_{h \to 0} \left(\frac{\sigma\varepsilon}{h} (u_{i+1} - 2u_i + u_{i-1}) + \frac{a_i}{2} (u_{i+1} - u_{i-1}) \right) = \lim_{h \to 0} h\,(q(x_i) + b_i u(x_i)) \tag{4.18}$$

As $h \to 0$, equation (4.18) becomes we get:

$$\frac{\sigma}{\rho} (u_{i-1} - 2u_i + u_{i+1}) + \frac{a(x_i)}{2} (u_{i+1} - u_{i-1}) = 0. \tag{4.19}$$

From (4.16), we have:

$$u_{i\pm1} = u_0(0) + (A - u_0(0))\, e^{-a(0)\frac{(1-(i\pm1)h)}{\varepsilon}} \tag{4.20}$$

Substituting (4.20)into (4.19) :

$$\begin{aligned}
&\frac{\sigma}{\rho}\Big[u_0(0)+(A-u_0(0))\, exp\left(-\frac{a(0)(1-(i+1)h)}{\varepsilon}\right)\\
&\qquad -2u_0(0)-2\,(A-u_0(0))\exp\left(-a(0)\left(\frac{1-ih}{\varepsilon}\right)\right)\\
&\qquad +u_0(0)+(A-u_0(0))\, exp\left(-\frac{a(0)(1-(i-1)h)}{\varepsilon}\right)\Big]\\
&+\frac{a(i)}{\rho}\Big[\,(A-u_0(0))\, exp\left(-\frac{a(0)(1-(i+1)h)}{\varepsilon}\right)\\
&\qquad -(A-u_0(0))\, exp\left(-\frac{a(0)(1-(i-1)h)}{\varepsilon}\right)\Big]=0
\end{aligned} \tag{4.21}$$

After some simplifications, we have:

$$\begin{aligned}
-\frac{\sigma}{\rho}(A-u_0(0))exp\left(-\frac{a(0)(1-(i+1)h)}{\varepsilon}\right)+\frac{2\sigma}{\rho}(A-u_0(0))\exp\left(-a(0)\left(\frac{1-ih}{\varepsilon}\right)\right)\\
+\frac{a(i)}{2}\left((A-u_0(0))exp\left(-\frac{a(0)(1-(1+i)h}{\varepsilon}\right)-[A-u_0(0)]\, exp\left(-\frac{a(0)(1-(i-1)h)}{\varepsilon}\right)\right)\\
-\frac{\sigma}{\rho}(A-u_0(0))exp\left(-\frac{a(0)(1-(i-1)h)}{\varepsilon}\right)=0
\end{aligned} \tag{4.22}$$

Simplifying the above, we obtain:

$$-\frac{\sigma}{\rho}\left[exp\left(-\frac{a(0)(1-(1+i)h)}{\varepsilon}\right)-2\exp\left(-a(0)\left(\frac{1}{\varepsilon}-i\rho\right)\right)+exp\left(-\frac{a(0)(1-(i-1)h)}{\varepsilon}\right)\right]$$

$$+\frac{a_i}{2}\left[exp\left(-\frac{a(0)(1-(1+i)h)}{\varepsilon}\right)-exp\left(-\frac{a(0)(1-(i-1)h)}{\varepsilon}\right)\right]=0$$

Taking the common factor and further simplifying, we obtain:

$$\begin{aligned}
-\frac{\sigma}{\rho}\left[\exp\left(-a(0)\left(\frac{1}{\varepsilon}-\rho\right)\right)-2\exp\left(\frac{-a(0)}{\varepsilon}\right)+\exp\left(-a(0)\left(\frac{1}{\varepsilon}+\rho\right)\right)\right]\\
+\frac{a(0)}{2}\left[\exp\left(-a(0)\left(\frac{1}{\varepsilon}-\rho\right)\right)-\exp\left(-a(0)\left(\frac{1}{\varepsilon}+\rho\right)\right)\right]=0.
\end{aligned}$$

Finally, we have:

$$\frac{\sigma}{\rho}\left[\exp\left(-a(0)\left(\frac{1}{\varepsilon}-\rho\right)\right)-2\exp\left(\frac{-a(0)}{\varepsilon}\right)+\exp\left(-a(0)\left(\frac{1}{\varepsilon}+\rho\right)\right)\right]$$
$$=\frac{a(0)}{2}\left[\exp\left(-a(0)\left(\frac{1}{\varepsilon}-\rho\right)\right)-\exp\left(-a(0)\left(\frac{1}{\varepsilon}+\rho\right)\right)\right]. \tag{4.23}$$

From the above, we obtain the fitting factor of the left boundary layer problem as:

$$\sigma_l=\frac{\rho a(0)}{2}\coth\left(\frac{\rho a(0)}{2}\right) \tag{4.24}$$

Similarly, the fitting factor of the right boundary layer (after simplification) is given as:

$$\sigma_r=\frac{\rho a(l)}{2}\coth\left(\frac{\rho a(l)}{2}\right) \tag{4.25}$$

Finally, the fitting factor for the boundary layer type problems will be:

$$\sigma=\frac{\rho a(x_i)}{2}\coth\left(\frac{\rho a(x_i)}{2}\right), \tag{4.26}$$

where

$$\coth(\frac{\rho a(x_i)}{2})=\frac{\cosh(\frac{\rho a(x_i)}{2})}{\sinh(\frac{\rho a(x_i)}{2})}$$

is the hyperbolic cotangent function. This ensures the scheme is stable and accurate even in the presence of boundary layers.

4.2.2 Discretization of the Differential Terms

We discretize the interval $[0,l]$ using N points with uniform step size $h=\frac{l}{N}$, so that:

$$x_i=ih,\quad i=0,1,2,\ldots,N.$$

Finite Difference Approximation

$$u(x_{i+1})=u(x_i)+hu'(x_i)+\frac{h^2}{2}u''(x_i)+\frac{h^3}{6}u^{(3)}(x_i)+\frac{h^4}{24}u^{(4)}(x_i)+\mathcal{O}(h^5),$$
$$u(x_{i-1})=u(x_i)-hu'(x_i)+\frac{h^2}{2}u''(x_i)-\frac{h^3}{6}u^{(3)}(x_i)+\frac{h^4}{24}u^{(4)}(x_i)+\mathcal{O}(h^5).$$

Second derivative (Central Difference Formula):

$$u(x_{i+1})-2u(x_i)+u(x_{i-1})=h^2u''(x_i)+\frac{h^4}{12}u^{(4)}(x_i)+\mathcal{O}(h^5)$$

$$u''(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

The truncation error of the central difference for the second derivative is well known:

$$\text{Truncation Error} = O(h^2),$$

meaning the discretization of $u''(x)$ is second-order accurate.

First derivative (Central Difference Formula): The central difference formula for the first derivative is:

$$\frac{u(x_{i+1}) - u(x_{i-1})}{2h} = u'(x_i) + \frac{h^2}{6} u^{(3)}(x_i) + \mathcal{O}(h^4)$$

$$u'(x_i) \approx \frac{u_{i+1} - u_{i-1}}{2h},$$

with a discretization

$$\text{Truncation Error} = O(h^2),$$

which means it is second-order accurate as well. Thus, the differential operator Lu at x_i is approximated as:

$$\sigma\varepsilon \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + a_i \frac{u_{i+1} - u_{i-1}}{2h} - b_i u_i - \lambda I_i = f_i,$$

where I_i is the discretization of the integral.

4.2.3 Composite Trapezoidal Rule for Integral Approximation

The integral term:

$$\int_0^{x_i} K(x_i, s) u(s)\, ds$$

is approximated using the composite trapezoidal rule:

$$I_i \approx \frac{h}{2} \left[K(x_i, x_0) u_0 + 2 \sum_{j=1}^{i-1} K(x_i, x_j) u_j + K(x_i, x_i) u_i \right].$$

Constructing the System of Equations

For $i = 1, 2, \ldots, N-1$, substituting I_i into the finite difference equation gives:

$$
\begin{aligned}
&\sigma\varepsilon\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}+a_i\frac{u_{i+1}-u_{i-1}}{2h}-b_iu_i \\
-\frac{\lambda h}{2}&\left[K(x_i,x_0)u_0+2\sum_{j=1}^{i-1}K(x_i,x_j)u_j+K(x_i,x_i)u_i\right]=f_i. \qquad (4.27)
\end{aligned}
$$

with the boundary conditions

$$u_0=A, \quad u_N=B. \qquad (4.28)$$

where $\varepsilon\in(0,1]$ is a perturbation parameter, λ is a real parameter, and l is some fixed number.

Manual Computation for Each i

For $i=1$:

$$\sigma\varepsilon\frac{u_2-2u_1+u_0}{h^2}+a_1\frac{u_2-u_0}{2h}-b_1u_1-\frac{\lambda h}{2}\left[K(x_1,x_0)u_0+K(x_1,x_1)u_1\right]=f_1.$$

Rewriting :

$$\left(\frac{\varepsilon\sigma}{h^2}-\frac{a_1}{2h}\right)u_0+\left(\frac{-2\varepsilon\sigma}{h^2}-b_1-\frac{\lambda h}{2}K(x_1,x_1)\right)u_1+\left(\frac{\varepsilon\sigma}{h^2}+\frac{a_1}{2h}\right)u_2=f_1+\frac{\lambda h}{2}K(x_1,x_0)u_0.$$

For $i=2$:

$$\sigma\varepsilon\frac{u_3-2u_2+u_1}{h^2}+a_2\frac{u_3-u_1}{2h}-b_2u_2-\frac{\lambda h}{2}\left[K(x_2,x_0)u_0+2K(x_2,x_1)u_1+K(x_2,x_2)u_2\right]=f_2.$$

Rewriting:

$$\left(\frac{\varepsilon\sigma}{h^2}-\frac{a_2}{2h}\right)u_1+\left(\frac{-2\varepsilon\sigma}{h^2}-b_2-\frac{\lambda h}{2}K(x_2,x_2)\right)u_2+\left(\frac{\varepsilon\sigma}{h^2}+\frac{a_2}{2h}\right)u_3$$

$$=f_2+\frac{\lambda h}{2}\left[K(x_2,x_0)u_0+2K(x_2,x_1)u_1\right].$$

For $i=3$:

$$
\begin{aligned}
&\sigma\varepsilon\frac{u_4-2u_3+u_2}{h^2}+a_3\frac{u_4-u_2}{2h}-b_3u_3 \\
&-\frac{\lambda h}{2}\left[K(x_3,x_0)u_0+2K(x_3,x_1)u_1+2K(x_3,x_2)u_2+K(x_3,x_3)u_3\right]=f_3.
\end{aligned}
$$

Rewriting:

$$\left(\frac{\varepsilon\sigma}{h^2} - \frac{a_3}{2h}\right) u_2 + \left(\frac{-2\varepsilon\sigma}{h^2} - b_3 - \frac{\lambda h}{2} K(x_3, x_3)\right) u_3 + \left(\frac{\varepsilon\sigma}{h^2} + \frac{a_3}{2h}\right) u_4$$

$$= f_3 + \frac{\lambda h}{2} \left[K(x_3, x_0)u_0 + 2K(x_3, x_1)u_1 + 2K(x_3, x_2)u_2 \right].$$

For $i = 4$:

$$\sigma\varepsilon \frac{u_5 - 2u_4 + u_3}{h^2} + a_4 \frac{u_5 - u_3}{2h} - b_4 u_4 - \frac{\lambda h}{2} [K(x_4, x_0)u_0 + 2K(x_4, x_1)u_1$$
$$+2K(x_4, x_2)u_2 + 2K(x_4, x_3)u_3 + K(x_4, x_4)u_4] = f_4.$$

Rearranging into the standard tridiagonal form:

$$\left(\frac{\varepsilon\sigma}{h^2} - \frac{a_4}{2h}\right) u_3 + \left(\frac{-2\varepsilon\sigma}{h^2} - b_4 - \frac{\lambda h}{2} K(x_4, x_4)\right) u_4 + \left(\frac{\varepsilon\sigma}{h^2} + \frac{a_4}{2h}\right) u_5$$

$$= f_4 + \frac{\lambda h}{2} \left[K(x_4, x_0)u_0 + 2K(x_4, x_1)u_1 + 2K(x_4, x_2)u_2 + 2K(x_4, x_3)u_3 \right].$$

For $i = 5$:

$$\sigma\varepsilon \frac{u_6 - 2u_5 + u_4}{h^2} + a_5 \frac{u_6 - u_4}{2h} - b_5 u_5 - \frac{\lambda h}{2} [K(x_5, x_0)u_0 + 2K(x_5, x_1)u_1$$
$$+2K(x_5, x_2 u_2 + 2K(x_5, x_3)u_3 + 2K(x_5, x_4)u_4 + K(x_5, x_5)u_5] = f_5.$$

Rearranging into tridiagonal form:

$$\left(\frac{\varepsilon\sigma}{h^2} - \frac{a_5}{2h}\right) u4_+ \left(\frac{-2\varepsilon\sigma}{h^2} - b_5 - \frac{\lambda h}{2} K(x_5, x_5)\right) u_5 + \left(\frac{\varepsilon\sigma}{h^2} + \frac{a_5}{2h}\right) u_6 = f_5 +$$
$$\frac{\lambda h}{2} \left[K(x_5, x_0)u_0 + 2K(x_5, x_1)u_1 + 2K(x_5, x_2)u_2 + 2K(x_5, x_3)u_3 + 2K(x_5, x_4)u_4 \right]$$

General Form

For $i = N - 1$, the equation is:

$$\left(\frac{\varepsilon\sigma}{h^2} - \frac{a_{N-1}}{2h}\right) u_{N-2} + \left(\frac{-2\varepsilon\sigma}{h^2} - b_{N-1} - \frac{\lambda h}{2} K(x_{N-1}, x_{N-1})\right) u_{N-1} + \left(\frac{\varepsilon\sigma}{h^2} + \frac{a_{N-1}}{2h}\right) u_N =$$

$$f_{N-1} + \frac{\lambda h}{2} \left[K(x_{N-1}, x_0)u_0 + \sum_{j=1}^{N-2} 2K(x_{N-1}, x_j)u_j \right]$$

.

In Matrix Representation

We express the given system of equations in matrix form as:

$$A\mathbf{u} = f + g\mathbf{u}, \tag{4.29}$$

where:

- A is the tridiagonal matrix derived from the finite difference and integral discretization,
- $\mathbf{u} = [u_0, u_1, \ldots, u_N]^T$ is the vector of unknowns,
- f is the right-hand side vector,
- g represents lower triangular matrix contributions from the integral discretization.

The tridiagonal matrix A is given by:

$$A_{i,i-1} = \frac{\varepsilon\sigma}{h^2} - \frac{a_i}{2h}, \quad A_{i,i} = -\frac{2\varepsilon\sigma}{h^2} - b_i - \frac{\lambda h}{2}K(x_i, x_i), \quad A_{i,i+1} = \frac{\varepsilon\sigma}{h^2} + \frac{a_i}{2h} \tag{4.30}$$

for $i = 1 : N - 1$.
The integral term contributes:

$$g_i\mathbf{u} = \frac{\lambda h}{2}\left[K(x_i, x_0)u_0 + \sum_{j=1}^{N-2} 2K(x_i, x_j)u_j\right] \tag{4.31}$$

So, we can now write the system in this new form :

$$\begin{bmatrix} A_{0,0} & A_{0,1} & 0 & \ldots & 0 \\ A_{1,0} & A_{1,1} & A_{1,2} & \ldots & 0 \\ 0 & A_{2,1} & A_{2,2} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & A_{N-1,N} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix} + \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{N-1} \end{bmatrix}. \tag{4.32}$$

Matrix Representation of g

The integral contribution vector g is structured as:

$$
g\mathbf{u} = \frac{\lambda h}{2} \begin{bmatrix} K(x_1, x_0) & 0 & 0 & \dots & 0 \\ K(x_2, x_0) & 2K(x_2, x_1) & 0 & \dots & 0 \\ K(x_3, x_0) & 2K(x_3, x_1) & 2K(x_3, x_2) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ K(x_{N-1}, x_0) & 2K(x_{N-1}, x_1) & 2K(x_{N-1}, x_2) & \dots & 2K(x_{N-1}, x_{N-2}) \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-2} \end{bmatrix}. \tag{4.33}
$$

Final Matrix $(A - g)$

Now, we subtract g from A to form $A - g$, ensuring a single unknown vector u: for $i = 1, 2, \dots, N - 1$.and $j = 0, 1, 2, \dots, N$.

$$
(A-g)_{i,j} = \begin{cases} A_{i,i-1} - \lambda h K(x_i, x_{i-1}), & j = i - 1 \quad , j > 0 \quad \text{(subdiagonal)} \\ A_{i,i}, & j = i \quad \text{(diagonal)} \\ A_{i,i+1}, & j = i + 1 \quad \text{(superdiagonal)} \\ -\lambda h K(x_i, x_j) & j < i - 1, j > 0 \quad \text{(integral contribution)} \\ -\frac{\lambda h}{2} K(x_i, x_0) & j = 0, j \neq i \\ 0, & j > i + 1. \end{cases}
$$

$$
(A - g)\mathbf{u} = f
$$

$$
\begin{bmatrix} A_{0,0} & A_{0,1} & 0 & 0 & \cdots & 0 \\ A_{1,0} - \frac{\lambda h}{2} K(x_1, x_0) & A_{1,1} & A_{1,2} & 0 & \cdots & 0 \\ -\frac{\lambda h}{2} K(x_2, x_0) & A_{2,1} - \lambda h K(x_2, x_1) & A_{2,2} & A_{2,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{\lambda h}{2} K(x_{N-1}, x_0) & -\lambda h K(x_{N-1}, x_1) & \cdots & A_{N-1,N-2} - \lambda h K(x_{N-1}, x_{N-2}) & A_{N-1,N-1} & A_{N-1,N} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix}
$$

4.3 Stability analysis of the scheme

In this part, we'll walk through a few important lemmas that play a key role in analyzing the numerical method we discussed earlier.

Lemma 4.2. *Let's consider the following differential part*

$$
l_h y_i = \sigma \varepsilon \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + a_i \frac{u_{i+1} - u_{i-1}}{2h} - b_i u_i \tag{4.34}
$$

rearranged into a tri-diagonal form like this:

$$l_h y_i = A_i U_{i-1} - B_i U_i + C_i U_{i+1} \tag{4.35}$$

where:

$$A_i = \frac{\sigma\varepsilon}{h^2} - \frac{a_i}{2h}, \quad B_i = \frac{2\sigma\varepsilon}{h^2} + b_i, \quad C_i = \frac{\sigma\varepsilon}{h^2} + \frac{a_i}{2h}. \tag{4.36}$$

Now, here are a few key results related to the above operator.
(I) Discrete maximum principle: If for every index i from 1 to N-1, we have

$$l_h y_i \geq 0, \tag{4.37}$$

and the boundary value satisfies

$$y_0 \geq 0, \quad \textit{and} \quad y_N \geq 0$$

implies that

$$y_i \geq 0, \quad \textit{for all} \quad 0 \leq i \leq N, \tag{4.38}$$

So basically, if the ends of the interval and the operator stay non-negative, the whole solution stays above zero too.

(II) Bound on the solution

$$l_h y_i = P_i, \quad for \quad 1 \leq i \leq N-1,$$

with boundary conditions:

$$y_0 = A, \quad y_N = B$$

Then the solution satisfies the following upper bound:

$$\|y_i\|_\infty \leq \max\{|A|,\ |B|\} + \frac{1}{\alpha} \max_{0 \leq i \leq N} |P_i|. \tag{4.39}$$

This tells us that the size of the solution stays under control-limited by the size of boundary values and the source term.
(III) Monotonic Source Term case
If $P_i \geq 0$ is non decreasing with i and also if $B_i - A_i - C_i \geq \alpha > 0$, then we have another

bound

$$|y_i| \leq \max\{|A|,\ |B|\} + \frac{P_i}{\alpha}, for \quad 1 \leq i \leq N-1, \tag{4.40}$$

so, in this case, the solution's size is affected predictably by the boundary value and the growing source function P_i.

Proof. (I) Let j be such that $\Psi_j = \min_{0 \leq i \leq N} y_i$, and suppose that $\Psi_j < 0$.
Evidently, $j \in \{0, N\}$, since $\Psi_0 \geq 0$ and $\Psi_N \geq 0$ by assumption.
Then it is clear that:

$$\Psi_j \leq \Psi_{j+1} \quad \text{and} \quad \Psi_j \leq \Psi_{j-1}.$$

We evaluate the discrete operator $l\Psi_j$ at the minimum point j:

$$\begin{aligned} l\Psi_j &= A_j\Psi_{j-1} - B_j\Psi_j + C_j\Psi_{j+1} \\ &= A_j(\Psi_{j-1} - \Psi_j) - (B_j - A_j - C_j)\Psi_j + C_j(\Psi_{j+1} - \Psi_j). \end{aligned}$$

Since $\Psi_{j-1} - \Psi_j \geq 0, \quad \Psi_{j+1} - \Psi_j \geq 0, \quad \text{and} \quad \Psi_j < 0$, we find:

$$l\Psi_j \leq -(B_j - A_j - C_j)\Psi_j < 0,$$

because $B_j - A_j - C_j < 0$ for an (4.42) and $\Psi_j < 0$.

This contradicts our assumption that $l\Psi_j \geq 0$. Therefore, our assumption that $\Psi_j < 0$ must be false, and we conclude:

$$\Psi_j \geq 0 \quad \text{for all } 0 \leq j \leq N.$$

Hence,

$$y_i \geq 0, \quad 0 \leq i \leq N.$$

□

Proof. (II) Let's define two mesh functions as follows:

$$\Psi_i = \max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0 \leq i \leq N}|P_i| \pm y_i. \tag{4.41}$$

for boundary value $i = 0$ and $i = N$,

$$\Psi_0 = \max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0 \leq i \leq N}|P_0| \pm y_0 \geq 0,$$

$$\Psi_N = \max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0 \leq i \leq N}|P_N| \pm y_N \geq 0,$$

and for $1 \leq i \leq N-1$,

$$
\begin{aligned}
l\Psi_i &= A_i\left(\max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0\leq i\leq N}|P_i| \pm y_{i-1}\right) \\
&\quad - B_i\left(\max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0\leq i\leq N}|P_i| \pm y_{i}\right) \\
&\quad + C_i\left(\max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0\leq i\leq N}|P_i| \pm y_{i+1}\right) \\
&= (-B_i + A_i + C_i)\max\{|A|,\ |B|\} + \frac{1}{\alpha}(A_i - B_i + C_i)\max_{0\leq i\leq N}|P_i| \\
&\quad \pm (A_i y_{i-1} - B_i y_i + C_i y_{i+1}) \\
&= (-B_i + A_i + C_i)\left(\max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0\leq i\leq N}|P_i|\right) \pm l_h y_i \\
&= (-B_i + A_i + C_i)\left(\max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0\leq i\leq N}|P_i|\right) \pm P_i.
\end{aligned}
$$

Now, assuming that

$$\frac{-B_i + A_i + C_i}{\alpha} \geq 1,$$

we get

$$l\Psi_i \geq 0.$$

By Equation(4.37) and (4.38), it follows that $\Psi_i \geq 0$ for all i. Therefore,

$$\pm y_i \leq \max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0\leq i\leq N}|P_i|,$$

which gives the desired result:

$$\|y_i\|_\infty \leq \max\{|A|,\ |B|\} + \frac{1}{\alpha}\max_{0\leq i\leq N}|P_i|.$$

□

Proof. (III) let us define the mesh:

$$\Psi_i = \max\{|A|,\ |B|\} + \frac{1}{\alpha}P_i \pm y_i.$$

For the boundary points $i = 0$ and $i = N$, we have:

$$\Psi_0 = \max\{|A|,\ |B|\} + \frac{1}{\alpha}P_0 \pm y_0 \geq 0,$$

$$\Psi_N = \max\{|A|,\ |B|\} + \frac{1}{\alpha}P_N \pm y_N \geq 0.$$

Now, we apply the difference operator for $1 \le i \le N-1$:

$$\begin{aligned} l\Psi_i &= A_i\Psi_{i-1} - B_i\Psi_i + C_i\Psi_{i+1} \\ &= (A_i - B_i + C_i)\max\{|A|,\ |B|\} \\ &\quad + \frac{1}{\alpha}(A_iP_{i-1} - B_iP_i + C_iP_{i+1}) \pm l_h y_i \\ &= (A_i - B_i + C_i)\max\{|A|,\ |B|\} + \frac{1}{\alpha}(A_iP_{i-1} - B_iP_i + C_iP_{i+1}) \pm P_i. \end{aligned}$$

Since P_i is nondecreasing, we know $P_{i-1} \le P_i \le P_{i+1}$. Therefore:

$$A_iP_{i-1} - B_iP_i + C_iP_{i+1} \le (A_i - B_i + C_i)P_i.$$

Thus,

$$l\Psi_i \ge (A_i - B_i + C_i)\max\{|A|,\ |B|\} + \frac{1}{\alpha}(A_i - B_i + C_i)P_i \pm P_i.$$

Using the assumption $A_i - B_i + C_i \ge \alpha > 0$, we obtain:

$$l\Psi_i \ge 0.$$

By Equation (4.37) and (4.38), we have $\Psi_i \ge 0$, which implies:

$$\pm y_i \le \max\{|A|,\ |B|\} + \frac{1}{\alpha}P_i,$$

and hence:

$$|y_i| \le \max\{|A|,\ |B|\} + \alpha^{-1}P_i.$$

□

The uniqueness of the discrete solution is ensured by the discrete maximum principle. Since the discrete system is linear, the existence of a solution follows from its uniqueness. Moreover, the discrete maximum principle also allows us to establish the boundedness of the numerical solution (B. C. Iragi & Munyakazi, 2020).

Lemma 4.3. *Assume that for all $i = 1 : N$, the following condition holds:*

$$|-b_i| \ge \alpha^* > 0, \tag{4.42}$$

for the difference operator (4.34) we have

$$\|u\|_\infty \le \max\{|u_0|, |u_1|\} + \frac{1}{\alpha^*}\max_{1\le i\le N}|ly_i|.$$

where α^* a uniform lower bound for b_i.

Proof.

$$Iy_i = A_i U_{i-1} - B_i U_i + C_i U_{i+1}$$

where:

$$\begin{aligned} A_i &= \frac{\sigma\varepsilon}{h^2} - \frac{a_i}{2h} > 0, \\ B_i &= \frac{2\sigma\varepsilon}{h^2} + b_i \quad > 0, \\ C_i &= \frac{\sigma\varepsilon}{h^2} + \frac{a_i}{2h} > 0. \end{aligned}$$

Also, observe that

$$A_i - B_i + C_i = -b_i > 0, \quad (b_i < 0),$$

since $\quad -b_i > \alpha^* > 0$, α^* uniform lower bound. ◻

Lemma 4.4. *Let $l_h y_i$ be the difference operator defined in equation (4.34). Then, the discrete problem given by equations (4.27)–(4.28) becomes:*

$$|l_h y_i| \leq \|f\|_\infty + C\lambda h \left(A + B + 2\sum_{j=1}^{i} |y_j| \right), \quad 1 \leq i \leq N-1. \tag{4.43}$$

Proof. From (4.27), we rewrite $l_h y_i$ as:

$$l_h y_i = f_i - \frac{\lambda h}{2} \left[K(x_i, x_0) y_0 + 2\sum_{j=1}^{i-1} K(x_i, x_j) y_j + K(x_i, x_i) y_i \right].$$

Taking absolute values:

$$|l_h y_i| \leq |f_i| + \left| \frac{\lambda h}{2} K(x_i, x_0) y_0 \right| + \left| \lambda h \sum_{j=1}^{i-1} K(x_i, x_j) y_j \right| + \left| \frac{\lambda h}{2} K(x_i, x_i) y_i \right|.$$

Now, assuming $K(x,t)$ is bounded , say $|K(x,t)| \leq M$, and defining $\|y\|_\infty = \max_j |y_j|$, we get:

$$|l_h y_i| \leq \|f\|_\infty + \frac{\lambda h}{2} M |y_0| + h\lambda M \sum_{j=1}^{i-1} |y_j| + \frac{\lambda h}{2} M |y_i|.$$

Grouping terms:

$$|l_h y_i| \leq \|f\|_\infty + M\lambda h \left(\frac{1}{2}|y_0| + \sum_{j=1}^{i-1} |y_j| + \frac{1}{2}|y_i| \right)$$

we get:

$$|l_h y_i| \leq \|f\|_\infty + C\lambda h (A + B + 2\sum_{j=1}^{i} |y_j|)$$

$$|l_h y_i| \leq \|f\|_\infty + C\lambda h(A + B + 2\sum_{j=1}^{i} |y_j|), \quad \text{for } 1 \leq i \leq N-1$$

Hence proved. □

Lemma 4.5. *For solution y_i of the difference scheme described in equation (4.27)-(4.28) the following bound holds:*

$$|y_i| \leq (\frac{\|f\|_\infty}{\alpha} + \max\{|A|,\ |B|\} + \frac{Ch\lambda(A+B)}{\alpha})(\exp(\frac{C_1 x_i}{\alpha})), \quad 1 \leq i \leq N. \tag{4.44}$$

Proof. Let the scheme

$$\sigma\varepsilon\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + a_i\frac{y_{i+1} - y_{i-1}}{2h} - b_i y_i = F$$

where

$$F = f_i - \frac{\lambda h}{2}\left[K(x_i, x_0)y_0 + 2\sum_{j=1}^{i-1} K(x_i, x_j)y_j + K(x_i, x_i)y_i\right]$$

with

$$y_0 = A, \quad and \quad y_N = B$$

define $v_i = \lambda h(A + B + 2\sum_{j=1}^{i-1}|y_j|)$ and $C = \frac{M}{2}$ from (4.43) we obtain

$$|l_h y_i| = |F| \leq Cv_i + \|f\|_\infty,$$

By boundedness of solution we have (4.39) so,

$$|y_i|_\infty \leq \max\{|A|,\ |B|\} + \alpha^{-1}(Cv_i + \|f\|_\infty) \tag{4.45}$$

$$\leq w + \frac{C_1}{\alpha}\sum_{j=1}^{i-1}|y_j| \tag{4.46}$$

where

$$w = \max\{|A|,\ |B|\} + \frac{\|f\|_\infty}{\alpha} + \frac{Ch\lambda(A+B)}{\alpha}, \quad and \quad C_1 = 2Ch\lambda$$

using the difference analogue of Gronwall's inequality to (4.46)

$$|y_i| \leq wexp(\alpha^{-1}C_1 x_i)$$

This show that

$$|y_i| \leq (\frac{\|f\|_\infty}{\alpha} + \max\{|A|,\ |B|\} + \frac{Ch\lambda(A+B)}{\alpha})(\exp(\frac{C_1 x_i}{\alpha})).$$

□

4.4 Uniform Convergence of the Scheme

In this part, we'll look into how accurately the method we discussed earlier approximates the true solution. To do that, first define the error at each mesh point x_i as $z_i = y_i - u_i$, where y_i is the numerical solution obtained from the discrete scheme (4.27)-(4.28) and u_i is the exact solution of the original problem.(1.7)-(1.8) evaluated at x_i.

This error term z_i, evaluated for $i = 0, 1, ...N$, follows its own discrete relation. By studying this relation, we can gain insights into how the numerical error is distributed and how it evolves throughout the domain.

$$l_h z_i = \varepsilon \left(z''(x_i) - \sigma \frac{z_{i+1} - 2z_i + z_{i-1}}{h^2} \right) + a_i \left(z'(x_i) - \frac{z_{i+1} - z_{i-1}}{2h} \right) - b_i z_i$$

$$-\lambda \left(\int_0^{x_i} K(x_i, s) u(s)\, ds \right)$$

$$+\lambda \frac{h}{2} \left[K(x_i, x_0) u_0 + 2 \sum_{j=1}^{i-1} K(x_i, x_j) u_j + K(x_i, x_i) u_i \right] = R_i, \quad 1 \leq i \leq N. \tag{4.47}$$

$$z_0 = 0, \quad z_N = 0. \tag{4.48}$$

Theorem 4.1. *Let $a, f \in C^2[0, I], K \in C^2(I \times I)$, for $I > 0$. Then, the remainder term R_i arise from the numerical scheme satisfies the bound:*

$$\|R_i\|_\infty \leq C h^n, \quad n = 1, 2 \tag{4.49}$$

where C is a constant independent of the mesh size h and ε.

The error is $O(h)$ when $\varepsilon \ll h$, and $O(h^2)$ when $h \ll \varepsilon$.

Proof. The truncation error of

$$L^N(u(x_i) - u_i) = f_i - L^N u_i = (L - L^N) u_i$$

$$= (\varepsilon u_i'' + a(x_i) u_i' - b(x_i) u_i) - f(x) - \lambda \int_0^x k(x_i, s) u(s) ds \tag{4.50}$$

$$- \left(\varepsilon \sigma D^+ D^- u_i + a_i D^0 u_i - b_i u_i - f_i - \lambda \int_0^x k(x_i, s) u(s) ds \right)$$

$$= (\varepsilon u_i'' + a u_i') - \left(\varepsilon \sigma D^+ D^- + a_i D^0 u_i \right) + R_1$$

$$= R_1 + R_2 \tag{4.51}$$

$$R_2 = \varepsilon\left(\frac{d^2}{dx^2} - \sigma D^+ D^-\right)u(x) + a_i\left(\frac{d}{dx} - D^0\right)u(x)$$
$$R_2 \le \left|\varepsilon\left[\frac{\rho a(x)}{2}\coth\left(\frac{\rho a(x)}{2}\right) - 1\right] D^+D^-u(x)\right| + \left|\varepsilon\left(\frac{d^2}{dx^2} - D^+D^-u(x)\right)\right|$$
$$+\left|a(x)\left(\frac{d}{dx} - D^0\right)u(x)\right| \tag{4.52}$$

where
$$\rho = \frac{h}{\varepsilon} \quad \text{and} \quad \sigma = \frac{\rho a(x)}{2}\coth\left(\frac{\rho a(x)}{2}\right).$$

For two constants c_1 and c_2, we have

$$|\rho\coth(\rho) - 1| \le c_1\rho^2, \quad \text{for } \rho \le 1.$$

Since $\coth(\frac{h}{\varepsilon}) \to 1$ as $\frac{h}{\varepsilon} \to 0$, and $\frac{h}{\varepsilon}\coth(\frac{h}{\varepsilon}) \to 1$ as $\frac{h}{\varepsilon} \to \infty$, we also have

$$\left|\frac{h}{\varepsilon}\coth(\frac{h}{\varepsilon}) - 1\right| \le c_2\rho^2, \quad \text{for all } \rho \ge 0.$$

Hence, we can write for all $\rho \ge 0$.

$$c_1\frac{\rho^2}{\rho+1} \le \rho\coth(\rho) - 1 \le c_2\frac{\rho^2}{\rho+1} \tag{4.53}$$

Using (4.52), we can write

$$\varepsilon\left[\frac{a(x)\rho}{2}\coth\left(\frac{a(x)\rho}{2}\right) - 1\right] \le \varepsilon\frac{\rho^2}{\rho+1} = \frac{\hat{h}^2}{h+\varepsilon} \tag{4.54}$$

Using the Taylor series expansion, we have the bound

$$\left|D^+D^-u(x)\right| \le C\|u'(\zeta)\|$$
$$\left|\left(\frac{d}{dx} - D^0\right)u(x)\right| \le Ch^2\|u'(\zeta)\|$$
$$\left|\left(\frac{d^2}{dx^2} - D^+D^-\right)u(x)\right| \le Ch^2\|u^{(4)}(\zeta)\| \tag{4.55}$$

Substituting (4.55) and (4.54) in (4.52), the truncation error becomes

$$|R_2| \le C\left(\frac{h^2}{h+\varepsilon}\right)\|u'(\zeta)\| + \varepsilon Ch^2\|u^{(4)}(\zeta)\| + Ch^2\|u'(\zeta)\| \tag{4.56}$$
$$R_2 \le \left(\frac{Ch^2}{h+\varepsilon}\right)\|u'(\zeta)\| + \varepsilon Ch^2\|u^{(4)}(\zeta)\| + Ch^2\|u'(\zeta)\| \tag{4.57}$$

From (4.50)

$$R_1 \approx -\frac{h^2}{12}\left[f''(\zeta)\right], f''(s) = \frac{d^2}{ds^2}\left[K(x_i, s)u(s)\right]$$

and from (4.57)

$$R_2 \leq \left(\frac{Ch^2}{h+\varepsilon}\right)\|u'(\zeta)\| + \varepsilon Ch^2\|u^{(4)}(\zeta)\| + Ch^2\|u'(\zeta)\|$$

Since the remainder term is $R = R_1 + R_2$, we obtain

$$|R| = |R_1 + R_2|$$

$$|R| \leq \left|-\frac{h^2}{12}\left[f''(\zeta)\right]\right| + \left(\frac{Ch^2}{h+\varepsilon}\right)\|u'(\zeta)\| + \varepsilon Ch^2\|u^{(4)}(\zeta)\| + Ch^2\|u'(\zeta)\|$$

Simplifying the above expression, we get

$$\|R\|_\infty \leq \frac{h^2}{12}\|f''\|_\infty + Ch^2\left(\frac{1}{h+\varepsilon}+1\right)\|u'\|_\infty + \varepsilon Ch^2\|u^{(4)}\|_\infty$$

- Term 1: $\frac{h^2}{12}\|f''\|_\infty = \mathcal{O}(h^2)$
- Term 2:

$$Ch^2\left(\frac{1}{h+\varepsilon}+1\right) \sim \begin{cases}\mathcal{O}(h), & \varepsilon \ll h\\ \mathcal{O}(h^2), & h \ll \varepsilon\end{cases}$$

- Term 3: $\varepsilon Ch^2\|u^{(4)}\|_\infty = \mathcal{O}(\varepsilon h^2)$

Overall, the error is $\mathcal{O}(h)$ when $\varepsilon \ll h$, and $\mathcal{O}(h^2)$ when $h \ll \varepsilon$.

□

Lemma 4.6. *Under the condition that the difference operator satisfies the discrete maximum principle and is stable (i.e, $A_i - B_i + C_i \geq \alpha > 0$), the solution z_i of the problem*

$$L_h z_i = R_i, \quad 1 \leq i \leq N-1, \quad z_0 = z_N = 0$$

satisfies the estimate

$$\|z\|_\infty \leq C_3 \max_{1\leq i\leq N-1} |R_i|,$$

where $\|z\|_\infty = \max_{0\leq i\leq N}|z_i|$, and C_3 is a constant independent of h and ε.

Proof. Given the estimate from (4.44)

$$|y_i| \leq (\frac{\|f\|_\infty}{\alpha} + \max\{|A|,\ |B|\} + \frac{Ch\lambda(A+B)}{\alpha})(\exp(\frac{C_1 x_i}{\alpha})), \quad 1 \leq i \leq N.$$

set $z_0 = A = 0$, $z_N = B = 0$. Then the estimate becomes

$$|y_i| \leq \alpha^{-1}\|f\|_\infty \exp(\alpha^{-1}C_1 x_i).$$

Let $f_i = R_i$, and $y_i = z_i$. Then

$$|z_i| \leq \alpha^{-1}\|R\|_\infty \exp(\alpha^{-1}C_1 x_i).$$

Since $x_i \in [0,1]$, we have

$$\exp(\alpha^{-1}C_1 x_i) \leq \exp(\alpha^{-1}C_1) =: C_2.$$

Therefore,

$$|z_i| \leq \alpha^{-1}C_2\|R\|_\infty =: C_3 \max_{1\leq i\leq N-1} |R_i|.$$

Hence,

$$\|z\|_\infty = \max_{0\leq i\leq N} |z_i| \leq C_3 \max_{1\leq i\leq N-1} |R_i|.$$

□

Theorem 4.2. *Assume the conditions outlined in Theorem 4.1 hold. Then, when the difference scheme* (4.27)-(4.28) *is applied to the continuous problem* (1.7)-(1.8) *on a uniform mesh, it achieves first-order convergence with respect to the discrete maximum norm, uniformly in ε. In other words, for C constant independent of both h and ε.*

$$\|y - u\|_\infty \leq Ch.$$

CHAPTER FIVE

Result and Discussions

In this section, we explore how well the proposed scheme performs by working through a couple of examples that illustrate its stability and accuracy. To evaluate the method, we use two key indicators: the maximum error e_N and the observed rate of convergence r_k.

Because the exact solutions are known for these examples, we can directly compute the maximum error across all mesh points using the formula

$$e_N = \max_{0 \le j \le 1} |u_j - y_j|, \tag{5.1}$$

where u_j represents the exact solution and y_j the numerical approximation.

To get a sense of how the accuracy improves with mesh refinement, we calculate the convergence rate with the expression

$$r_k := \log_2 \left(\frac{e_{N_k}}{e_{2N_k}} \right), \quad k = 1, 2, 3, \ldots \tag{5.2}$$

This rate helps us understand how quickly the error decreases as we increase the number of mesh points.

Example 5.1. Consider the following linear second-order singularly perturbed Volterra integro-differential equation (Panda & Mohapatra, 2023):

$$\varepsilon u''(x) + u'(x) - u(x) - \int_0^x x u(s)\, ds = f(x), \quad 0 < x \le 1 \tag{5.3}$$

with boundary conditions:

$$u(0) = 1, \quad u(1) = e^{-1/\varepsilon} \tag{5.4}$$

The exact solution is given by: $u(x) = e^{-x/\varepsilon}$. To match the exact solution, the right-hand side $f(x)$ is chosen as: $f(x) = -e^{-x/\varepsilon} - \varepsilon x \left(1 - e^{-x/\varepsilon}\right)$.

Example 5.2. Consider the following linear second-order singularly perturbed Volterra integro-differential equation:

$$\varepsilon u''(x) + (1 + x)\, u'(x) + (2 - x)\, u(x) - \int_0^x x\, u(s)\, ds = f(x), \quad 0 < x \le 1,$$

with boundary conditions:

$$u(0) = 1, \quad u(1) = e^{-1/\varepsilon},$$

and the exact solution is $u(x) = e^{-x/\varepsilon}$. To match the exact solution, the right-hand side $f(x)$ is chosen as: $f(x) = \left(\frac{-x}{\varepsilon} + 2 - x\right) e^{-x/\varepsilon} - x\varepsilon \left(1 - e^{-x/\varepsilon}\right)$.

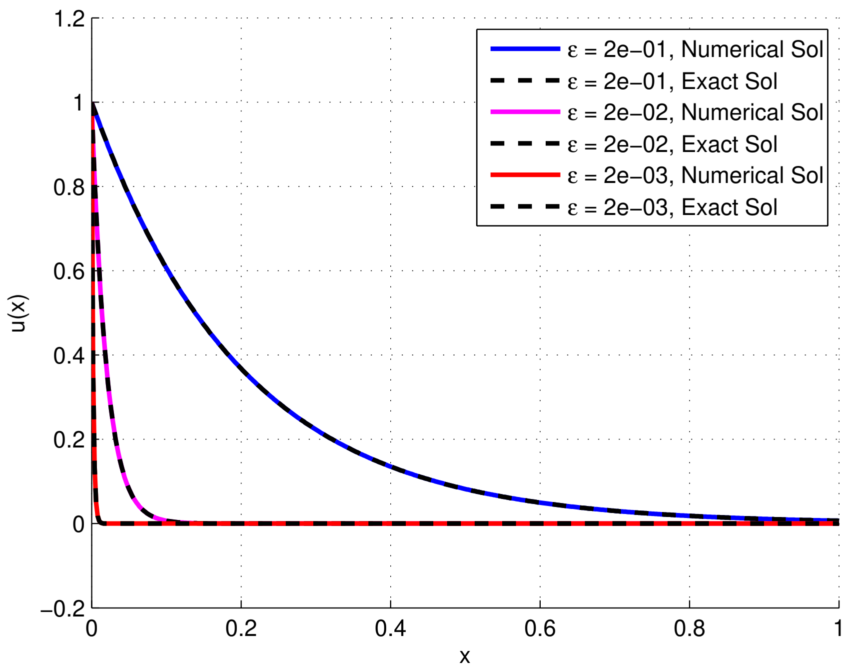

Figure 5.1: Solution behavior of Example 5.1 with the boundary layer effects as ε diminishes, with N=500.

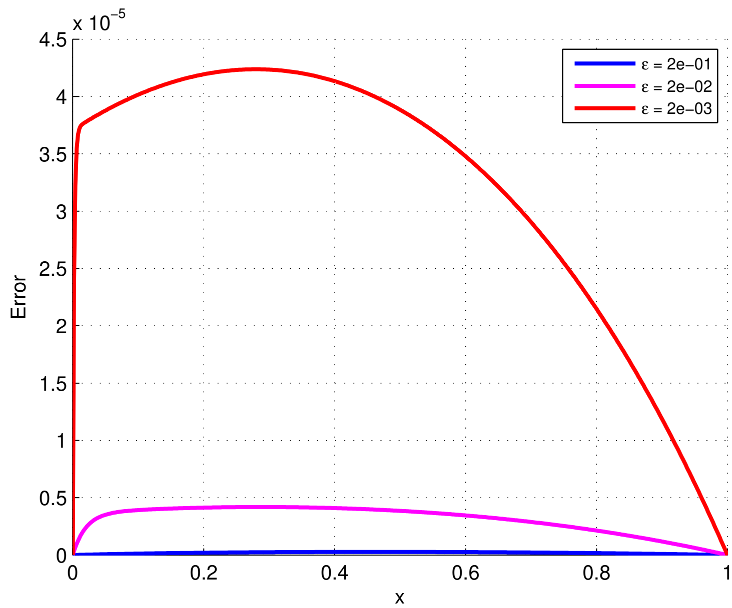

Figure 5.2: Point-wise absolute error for Example 5.1 with a fixed mesh size of $N = 500$ as ε decreases.

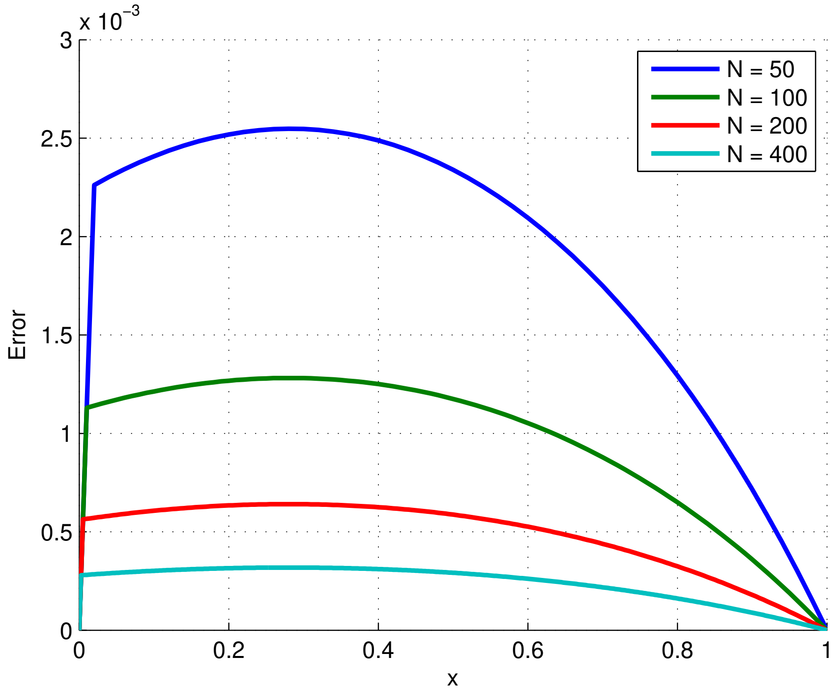

Figure 5.3: Point-wise absolute error using the proposed method for Example 5.1 with increasing mesh size N, and fixed $\varepsilon = 2^{-05}$.

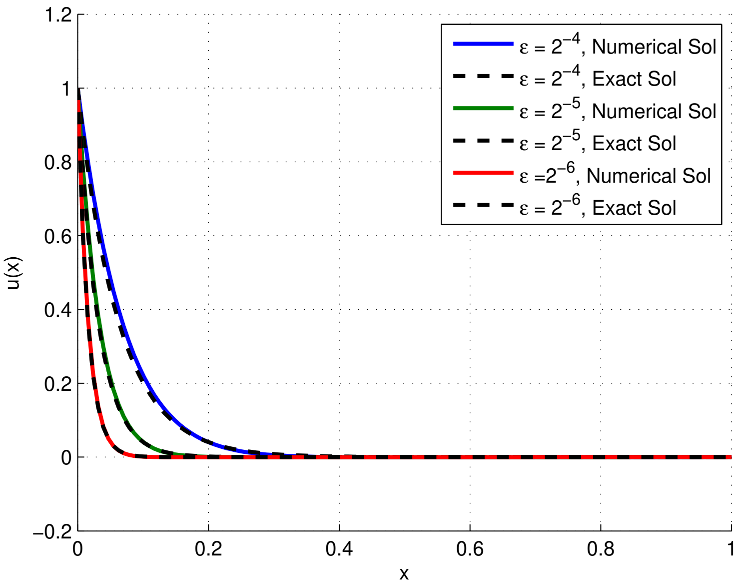

Figure 5.4: Solution behavior for Example 5.2 with the boundary layer effects as ε diminishes, with N=128.

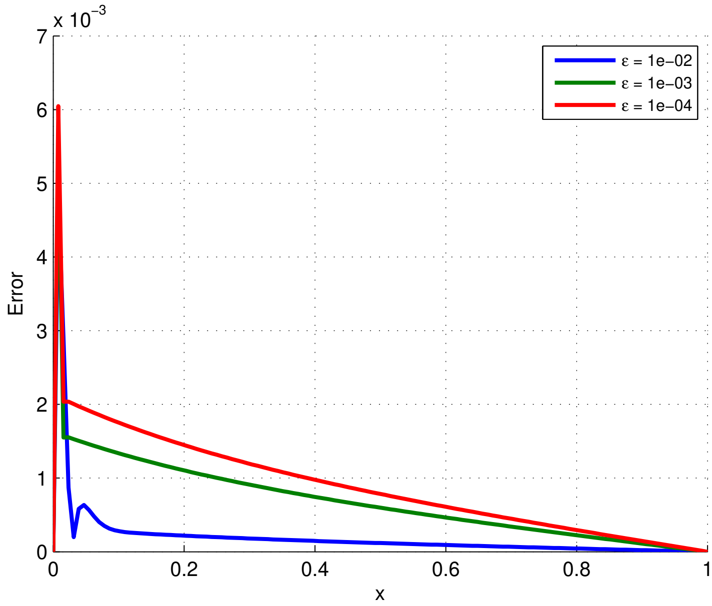

Figure 5.5: Point-wise absolute error using the proposed method for Example 5.2 with a fixed mesh size of $N = 128$ as ε decreases.

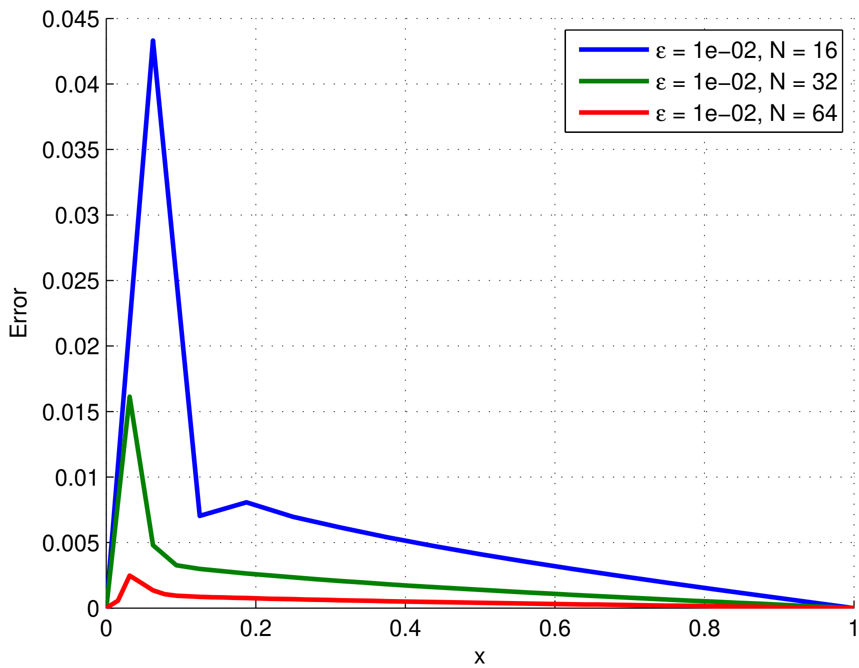

Figure 5.6: Point-wise absolute error using the proposed method for Example 5.2 with increasing mesh size N, and fixed $\varepsilon = 10^{-02}$.

Table 5.1: Maximum absolute error of the exponentially finite difference method for Example 5.1.

	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
$\varepsilon = 10^{-2}$	1.28E−04	3.24E−05	8.11E−06	2.03E−06	5.07E−07
$\varepsilon = 10^{-3}$	7.50E−04	2.67E−04	7.75E−05	2.03E−05	5.12E−06
$\varepsilon = 10^{-4}$	9.81E−04	4.79E−04	2.27E−04	1.01E−04	3.83E−05
$\varepsilon = 10^{-5}$	1.00E−03	5.02E−04	2.50E−04	1.24E−04	6.07E−05

Table 5.2: Convergence rate of the exponentially finite difference method applied to Example 5.1.

	$r = 1$	$r = 2$	$r = 3$	$r = 4$
$\varepsilon = 10^{-2}$	1.99	2.00	2.00	2.00
$\varepsilon = 10^{-3}$	1.49	1.79	1.94	1.98
$\varepsilon = 10^{-4}$	1.03	1.08	1.17	1.39
$\varepsilon = 10^{-5}$	1.00	1.01	1.01	1.03

Table 5.3: Maximum absolute error of the exponentially finite difference method for Example 5.2.

	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
$\varepsilon = 10^{-2}$	8.49E-04	2.06E-04	5.10E-05	1.27E-05	3.18E-06
$\varepsilon = 10^{-3}$	5.52E-03	2.25E-03	6.29E-04	1.32E-04	3.24E-05
$\varepsilon = 10^{-4}$	6.05E-03	2.99E-03	1.47E-03	7.06E-04	3.10E-04
$\varepsilon = 10^{-5}$	6.10E-03	3.04E-03	1.52E-03	7.57E-04	3.76E-04

Table 5.4: Convergence rate of the exponentially finite difference method for Example 5.2.

	$r = 1$	$r = 2$	$r = 3$	$r = 4$
$\varepsilon = 10^{-2}$	2.05	2.01	2.00	2.00
$\varepsilon = 10^{-3}$	1.30	1.84	2.25	2.03
$\varepsilon = 10^{-4}$	1.01	1.03	1.06	1.19
$\varepsilon = 10^{-5}$	1.00	1.00	1.01	1.01

Table 5.5: Comparison of maximum point-wise absolute errors using the proposed method and the method without the fitting parameter for Example 5.1

ε	Method	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
10^{-2}	Without fitting parameter	1.94E−02	4.62E−03	1.16E−03	2.88E−04	7.20E−05
	With fitting parameter	1.28E−04	3.24E−05	8.11E−06	2.03E−06	5.07E−07
10^{-3}	Without fitting parameter	5.88E−01	3.42E−01	1.30E−01	3.27E−02	7.49E−03
	With fitting parameter	7.50E−04	2.67E−04	7.75E−05	2.03E−05	5.12E−06
10^{-4}	Without fitting parameter	9.43E−01	8.99E−01	8.13E−01	6.59E−01	4.26E−01
	With fitting parameter	9.81E−04	4.79E−04	2.27E−04	1.01E−04	3.83E−05
10^{-5}	Without fitting parameter	1.11E+00	1.00E+00	9.78E−01	9.59E−01	9.21E−01
	With fitting parameter	1.00E−03	5.02E−04	2.50E−04	1.24E−04	6.07E−05

Table 5.6: Comparison of rates of convergence using the proposed method and the method without the fitting parameter for Example 5.1

ε	Method	$r = 1$	$r = 2$	$r = 3$	$r = 4$
10^{-2}	Without fitting parameter	2.07	2.00	2.00	2.00
	With fitting parameter	1.99	2.00	2.00	2.00
10^{-3}	Without fitting parameter	0.78	1.40	1.99	2.13
	With fitting parameter	1.49	1.79	1.94	1.98
10^{-4}	Without fitting parameter	0.07	0.15	0.30	0.63
	With fitting parameter	1.03	1.08	1.17	1.39
10^{-5}	Without fitting parameter	0.14	0.03	0.03	0.06
	With fitting parameter	1.00	1.01	1.01	1.03

Table 5.7: Comparison of maximum point-wise absolute errors using the proposed method and the method without the fitting parameter for Example 5.2.

ε	Method	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
10^{-2}	Without fitting parameter	2.05E−02	4.89E−03	1.22E−03	3.05E−04	7.61E−05
	With fitting parameter	8.49E−04	2.06E−04	5.10E−05	1.27E−05	3.18E−06
10^{-3}	Without fitting parameter	6.04E−01	3.47E−01	1.31E−01	3.29E−02	7.54E−03
	With fitting parameter	5.52E−03	2.25E−03	6.29E−04	1.32E−04	3.24E−05
10^{-4}	Without fitting parameter	1.11E+00	9.10E−01	8.18E−01	6.62E−01	4.27E−01
	With fitting parameter	6.05E−03	2.99E−03	1.47E−03	7.06E−04	3.10E−04
10^{-5}	Without fitting parameter	1.60E+00	2.45E+01	9.93E−01	9.62E−01	9.22E−01
	With fitting parameter	6.10E−03	3.04E−03	1.52E−03	7.57E−04	3.76E−04

Table 5.8: Comparison of rates of convergence using the proposed method and the method without the fitting parameter for Example 5.2.

ε	Method	$r = 1$	$r = 2$	$r = 3$	$r = 4$
10^{-2}	Without fitting parameter	2.07	2.00	2.00	2.00
	With fitting parameter	2.05	2.01	2.00	2.00
10^{-3}	Without fitting parameter	0.80	1.41	1.99	2.13
	With fitting parameter	1.30	1.84	2.25	2.03
10^{-4}	Without fitting parameter	0.29	0.15	0.31	0.63
	With fitting parameter	1.01	1.03	1.06	1.19
10^{-5}	Without fitting parameter	-3.94	4.63	0.05	0.06
	With fitting parameter	1.00	1.00	1.01	1.01

Table 5.9: Comparison of maximum absolute errors between the Hybrid scheme and EFDM for Example 5.1.

ε	Method	$N = 128$	$N = 256$	$N = 512$	$N = 1024$
10^{-2}	Hybrid in (Panda & Mohapatra, 2023)	2.90E−3	9.54E−4	3.05E−4	9.71E−5
	EFDM	1.28E−4	3.24E−5	8.11E−6	2.03E−6
10^{-3}	Hybrid in (Panda & Mohapatra, 2023)	2.85E−3	9.32E−4	2.95E−4	9.14E−5
	EFDM	7.50E−4	2.67E−4	7.75E−5	2.03E−5
10^{-5}	Hybrid in (Panda & Mohapatra, 2023)	2.83E−3	9.22E−4	2.91E−4	8.99E−5
	EFDM	1.00E−3	5.02E−4	2.50E−4	1.24E−4

Table 5.10: Comparison of convergence rates between the Hybrid scheme in and EFDM for Example 5.1.

ε	Method	$r = 1$	$r = 2$	$r = 3$
10^{-2}	Hybrid in (Panda & Mohapatra, 2023)	1.60	1.64	1.65
	EFDM	2.00	2.00	2.00
10^{-3}	Hybrid in (Panda & Mohapatra, 2023)	1.61	1.65	1.69
	EFDM	1.79	1.94	1.98
10^{-5}	Hybrid in (Panda & Mohapatra, 2023)	1.62	1.66	1.69
	EFDM	1.01	1.01	1.03

Discussion

To validate the performance of the proposed exponentially fitted Difference Method (EFDM), we conducted a series of numerical experiments based on two benchmark SP-VIDEs. The objective was to assess the method's ability to produce accurate results and maintain uniform convergence under varying mesh sizes and perturbation parameters ε.

Figures 5.1 and 5.4 display the numerical solution behavior for Examples 5.1 and 5.2, respectively. As the parameter ε decreases, a sharp boundary layer forms near the left boundary, specifically around $x = 0$, and becomes more pronounced with smaller ε values. Beyond this region, the solution remains smooth and nearly flat throughout the rest of the domain. These figures clearly demonstrate that the proposed method accurately captures the steep gradients within this layer, effectively addressing one of the main challenges in solving singularly perturbed problems.

To further evaluate the method's precision, we plotted pointwise absolute errors in Figures 5.2 and 5.5 show that as ε decreases, the error increases due to sharper boundary layers but remains uniformly bounded, confirming the method's robustness for fixed N. Figures 5.3 and 5.6 extend this by showing error trends for fixed ε as N increases. The downward error trajectory supports the method's convergence as N grows.

Tables 5.1 and 5.3 present the maximum absolute errors for both examples under multiple ε values and increasing mesh sizes. These results reveal a consistent decline in error with denser discretization, highlighting the method's mesh sensitivity and reliability.

Tables 5.2 and 5.4 examine convergence rates. For larger ε, the method achieves nearly second-order convergence (rates around 2.00). However, as ε reduces, convergence rates slightly decrease but still remain within a satisfactory range (above 1.00 for very small ε), which aligns well with theoretical expectations for singular perturbation problems.

Crucially, Tables 5.5 through 5.8 provide comparative results between the proposed method and one without fitting. It is evident that EFDM consistently outperforms the unfitted version. For example, Table 5.7 shows that, at $\varepsilon = 10^{-5}$ and $N = 2048$, the maximum pointwise error drops from 9.22×10^{-1} (without fitting) to 3.76×10^{-4} (with fitting). Similarly, Table 5.8 confirms that convergence rates remain stable and accurate even for the smallest perturbations when fitting is employed, whereas the unfitted version exhibits erratic or divergent behavior.

Tables 5.9 and 5.10 present a comparative analysis of the Hybrid scheme (Panda & Mohapatra, 2023) and EFDM in terms of maximum absolute error and convergence rate for Example 5.1. The results show that EFDM consistently produces smaller errors and achieves higher convergence rates across different values of ε, indicating its superior accuracy and efficiency in solving singularly perturbed problems.

Conclusion

This thesis focuses on designing a finite difference method with an exponential fitting parameter for second-order linear singularly perturbed Volterra integro-differential equations. The differential part was approximated using a standard central difference scheme. For the integral part, we used the composite Trapezoidal rule on a uniform partition. To address the sharp behavior due to the small perturbation parameter, we fitted an exponential parameter based on the solution's expected form. We investigated the stability and convergence of the method, as well as how the perturbation parameter influences the solution through plots. The provided analysis demonstrates that the method with special treatment achieves uniform convergence with first-order accuracy. To support this, a couple of numerical examples were presented, confirming the stability and reliability of the scheme. All computations were performed using MATLAB. These results confirm that the exponentially fitted difference method (EFDM) is both accurate and robust, effectively addressing the numerical challenges posed by singularly perturbed Volterra integro-differential equations (SPVIDEs). The method delivers stable solutions even under highly stiff conditions and maintains consistent performance across varying mesh resolutions and perturbation parameters. This establishes EFDM as a reliable and practical tool for solving a broad class of singularly perturbed integro-differential problems.

Recommendation

In this thesis, we focused on developing an exponentially fitted difference technique for solving second-order SPVIDEs. The technique was both reliable and efficient, yielding stable and precise results. From these results, we would like to propose for further work the consideration of nonlinear problems of these types, perhaps via the finite element method, which could be more adaptable in complex situations.

Research Fund Acknowledgment

This research project is funded by Adama Science and Technology University under the grand number ASTU/SM - R /1160/25, Adama, Ethiopia.

References

Aigo, M. (2013). On the numerical approximation of volterra integral equations of the second kind using quadrature rules. *International Journal of Advanced Scientific and Technical Research*, *1*(3), 558–564.

Amiraliyev, G., & Şevgin, S. (2006). Uniform difference method for singularly perturbed volterra integro-differential equations. *Applied mathematics and computation*, *179*(2), 731–741.

Angell, J., & Olmstead, W. (1987). Singularly perturbed volterra integral equations ii. *SIAM Journal on Applied Mathematics*, *47*(6), 1150–1162.

Angell, J., & Olmstead, W. E. (1985). Singular perturbation analysis of an integrodifferential equation modelling filament stretching. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, *36*(3), 487–490.

Babu, G., Prithvi, M., Sharma, K. K., & Ramesh, V. (2024). A uniformly convergent numerical algorithm on harmonic (h ()) mesh for parabolic singularly perturbed convection-diffusion problems with boundary layer. *Differential Equations and Dynamical Systems*, *32*(2), 551–564.

Bijura, A. M. (2006). Initial-layer theory and model equations of volterra type. *IMA journal of applied mathematics*, *71*(3), 315–331.

Burton, T. A. (2005). *Volterra integral and differential equations* (Vol. 202). Elsevier.

Cakir, M., & Gunes, B. (2022). A fitted operator finite difference approximation for singularly perturbed volterra–fredholm integro-differential equations. *Mathematics*, *10*(19), 3560.

Ebrahimi, N., & Rashidinia, J. (2014). Spline collocation for fredholm and volterra integro-differential equations. *International Journal of Mathematical Modelling & Computations*, *4*(3 (SUMMER)), 289–298.

Erdoğan, F. (2023). A second order numerical method for singularly perturbed volterra integro-differential equations with delay. *International Journal of Mathematics and Computer in Engineering.* Retrieved from https://doi.org/10.2478/ijmce-2024-0007 doi: 10.2478/ijmce-2024-0007

Erdoğan, F., & Amiraliyev, G. M. (2012). Fitted finite difference method for singularly perturbed delay differential equations. *Numerical Algorithms*, *59*(1), 131–145. Retrieved from https://doi.org/10.1007/s11075-011-9480-7 doi: 10.1007/s11075-011-9480-7

Hadi, A. A. (2023). Numerical solution of a volterra integral equation. *Formosa Journal of Applied Sciences*, *2*(5). Retrieved from https://doi.org/10.55927/fjas.v2i5.4038 doi: 10.55927/fjas.v2i5.4038

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning.

Iragi, B. (2017). *On the numerical integration of singularly perturbed volterra integro-differential equations* (Unpublished doctoral dissertation). Department of Mathematics and Applied Mathematics, University of the Western ....

Iragi, B. C., & Munyakazi, J. B. (2020). A uniformly convergent numerical method for a singularly perturbed volterra integro-differential equation. *International Journal of Computer Mathematics*, *97*(4), 759–771.

Kadalbajoo, M. K., & Gupta, V. (2010). A brief survey on numerical methods for solving singularly perturbed problems. *Applied mathematics and computation*, *217*(8), 3641–3716.

Kadalbajoo, M. K., Patidar, K. C., & Sharma, K. K. (2006). ε-uniformly convergent fitted methods for the numerical solution of the problems arising from singularly perturbed general ddes. *Applied Mathematics and Computation*, *182*(1), 119–139.

Koto, T. (2002). Stability of runge–kutta methods for delay integro-differential equations. *Journal of Computational and Applied Mathematics*, *145*(2), 483–492.

Kudu, M., Amirali, I., & Amiraliyev, G. M. (2016). A finite-difference method for a singularly perturbed delay integro-differential equation. *Journal of Computational and Applied Mathematics*, *308*, 379–390.

Kumar, S., Kumar, S., & Das, P. (2024). Second-order a priori and a posteriori error estimations for integral boundary value problems of nonlinear singularly perturbed parameterized form. *Numerical Algorithms*, 1–28.

Liu, L.-B., Ye, L., Bao, X., & Zhang, Y. (2024). A second order numerical method for a volterra integro-differential equation with a weakly singular kernel. *Networks and Heterogeneous Media*, *19*(2), 740–752.

Lodge, A., McLeod, J., & Nohel, J. (1978). A nonlinear singularly perturbed volterra integrodifferential equation occurring in polymer rheology. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, *80*(1-2), 99–137.

Manebo, W. S., Woldaregay, M. M., Dinka, T. G., & Duressa, G. F. (2024). An equidistributed grid-based second-order scheme for a singularly perturbed fredholm integro-differential equation with an interior layer. *Applied Mathematics and Computation*, *464*, 128398. Retrieved from https://www.sciencedirect.com/science/article/pii/S0096300323005672 doi: https://doi.org/10.1016/j.amc.2023.128398

Messina, E., & Vecchio, A. (2020). Long-time behaviour of the approximate solution to quasi-convolution volterra equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, *54*(1), 129–143.

Miranker, A. (2001). *Numerical methods for stiff equations and singular perturbation problems: and singular perturbation problems* (Vol. 5). Springer Science & Business Media.

Moore, R. E. (1979). *Methods and applications of interval analysis.* SIAM.

Panda, A., & Mohapatra, J. (2023). On the convergence analysis of efficient numerical schemes for singularly perturbed second order volterra integro-differential equations. *Journal of Applied Mathematics and Computing*, *69*(4), 3509–3532.

Panda, A., Mohapatra, J., & Amirali, I. (2021). A second-order post-processing technique for singularly perturbed volterra integro-differential equations. *Mediterranean Journal of Mathematics*, *18*, 25. Retrieved from https://doi.org/10.1007/s00009-021-01873-8 doi: 10.1007/s00009-021-01873-8

Prandtl, L. (1905). Uber flussigkeitsbewegung bei sehr kleiner reibung. *Verhandl. 3rd Int. Math. Kongr. Heidelberg (1904), Leipzig.*

Priyadarshana, S., Padhan, A., & Mohapatra, J. (2025). Adaptive grid based moving mesh algorithms for singularly perturbed second-order volterra integro-differential equations. *Quaestiones Mathematicae.* (Published online: 06 Mar 2025) doi: 10.2989/16073606.2025.2462939

Roos, H.-G. (2008). *Robust numerical methods for singularly perturbed differential equations.* Springer.

Salama, A., & Evans, D. J. (2001). Fourth order scheme of exponential type for singularly perturbed volterra integro-differential equations. *International journal of computer mathematics*, *77*(1), 153–164.

Şevgin, S. (2014). Numerical solution of a singularly perturbed volterra integro-differential equation. *Advances in Difference Equations*, *2014*, 1–15.

Sladek, V., Sladek, J., & Zhang, C. (2005). Local integro-differential equations with domain elements for the numerical solution of partial differential equations with variable coefficients. *Journal of engineering mathematics*, *51*, 261–282.

Wazwaz, A.-M. (2015). *First course in integral equations, a.* World Scientific Publishing Company.

Woldaregay, M. M., & Duressa, G. F. (2019). Parameter uniform numerical method for singularly perturbed parabolic differential difference equations. *Journal of the Nigerian Mathematical Society*, *38*(2), 223–245.

Wu, S., & Gan, S. (2009). Errors of linear multistep methods for singularly perturbed volterra delay-integro-differential equations. *Mathematics and Computers in Simulation*, *79*(10), 3148–3159.

Zhang, W., Zhang, X., & Gan, J. (2024). Integrated decision of production scheduling and condition-based maintenance planning for multi-unit systems with variable replacement thresholds. *Journal of Manufacturing Systems*, *74*, 647–664.

Zhongdi, C., & Lifeng, X. (2006). A parameter robust numerical method for a singularly perturbed volterra equation in security technologies. *matrix*, *1*, 20–22.

Zimmerman, W. (2004). Coupling variables revisited: Inverse problems, line integrals, integral equations, and integro-differential equations. *Process Modelling and Simulation with Finite Element Methods*, *1*, 245.

Çimen, E., & Yatar, S. (2020). A numerical method for solving linear volterra delay integro-differential equations using fitted difference scheme. *Advances in Difference Equations*, *2020*(1), 1–16. Retrieved from https://doi.org/10.1186/s13662-020-02580-w doi: 10.1186/s13662-020-02580-w