UNIVERSITY OF GRANADA

FACULTY OF SCIENCES

A Study of Additive Linear Multistep Methods for Solving Advection-Diffusion-Reaction Equations

by

Raed Ali Mara'Beh

A dissertation presented in fulfillment of the requirements for the Doctor's degree from the University of Granada.

Granada, Spain

Supervised By

1. Prof. Pedro González Rodelas (University of Granada, Professor of Applied Mathematics).
2. Prof. José Miguel Mantas Ruiz (University of Granada, Professor of Computer Science).

This thesis entitled:

A Study of Additive Linear Multistep Methods for Solving Advection-Diffusion-Reaction Equations

written by Raed Ali Mara'Beh

has been approved for the Department of Applied Mathematics

Pedro González Rodelas

José Miguel Mantas Ruiz

Date: _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work.

Editor: Universidad de Granada. Tesis Doctorales
Autor: Raed Ali Mara'Beh
ISBN: 978-84-1195-380-1
URI: https://hdl.handle.net/10481/92973

Abstract

This thesis investigates the efficacy of advanced numerical methods for solving advection-diffusion-reaction (ADR) problems, essential for describing transport mechanisms in fluid or solid mediums. It encompasses a detailed study of variable stepsize, semi-implicit, backward differentiation formula (VSSBDF) methods up to fourth order and introduces novel 3-additive linear multistep methods, emphasizing the treatment of ADR models that are typically formulated as partial differential equations. By discretizing these into systems of ordinary differential equations for numerical analysis, this work addresses the challenges posed by stiff terms in ADR models.

The first part of the thesis investigates the performance of VSSBDF methods of up to fourth order for solving ADR models employing two different implicit-explicit (IMEX) splitting approaches: a *physics-based* splitting and a splitting based on a dynamic linearization of the resulting system of ODEs , which is referred to as *Jacobian splitting*. We develop an adaptive time-stepping and error control algorithm for VSSBDF methods up to fourth order based on a step-doubling refinement technique using estimates of the local truncation errors. Through a systematic comparison between physics-based and Jacobian splitting across seven ADR test models, we evaluate the performance based on CPU times and corresponding accuracy. Our findings demonstrate the general superiority of Jacobian splitting in several experiments.

The second part of the thesis gives a comprehensive analysis of 3-additive linear multistep methods, designed to separately treat diffusion, reaction, and advection components of differential equations that model problems in science and engineering. This approach addresses the limitations of conventional IMEX methods, which typically combine the three terms with only two numerical methods. The stability and order of convergence of these new methods are investigated, and their performance is compared with popular IMEX methods. The findings reveal that the new 3-additive methods generally offer larger stability regions and superior computational efficiency compared to the tested IMEX methods in certain cases.

Together, these studies contribute to the field of numerical analysis by offering enhanced methods for the efficient and accurate solution of differential equations in science and engineering, reflecting significant advancements in the treatment of advection-diffusion-reaction

systems.

Acknowledgements

First of all, I want to thank my supervisors, Professor Pedro González Rodelas and Professor José Miguel Mantas Ruiz for their continuous support and deep knowledge throughout my academic journey. Their guidance has been invaluable, and this thesis owes much to their wisdom.

I also wish to express my sincere appreciation to Professor Raymond J. Spiteri, for his guidance and contribution to this work. His insights have been crucial to the success of the work presented in this thesis.

My gratitude extends to the Department of Mathematics department at the University of Granada for providing support and resources.

A special thanks goes out to my colleagues and friends across Spain, Canada, Qatar, and Palestine. Their support and advice have not only contributed to this work but have also enriched my personal and professional development.

This thesis is dedicated with all my love to my beloved parents, Ali and Ayda; my darling wife, Raheeq; and my dear daughter, Sara. To my brothers: Mohammad, Saed, Baha', and Ala', and to all my family back home.

My gratitude for you is boundless, and my words cannot fully express the depth of my appreciation

Contents

List of Tables

List of Figures

List of Abbreviations

BLAS	Basic Linear Algebra Subprograms Library
DRA	Diffusion-reaction-advection
JS	Jacobian Splitting
IEE	Implicit-Explicit-Explicit
IIE	Implicit-Implicit-Explicit
IMEX	Implicit-Explicit
IVP	Initial value problem
LIS	Library of Iterative Solvers
LMM	Linear Multisteps methods
LTE	Local Truncation Errors
ODE	Ordinary differential equation
PDE	Partial differential equation
PS	Physical Splitting
RHS	Right hand side
SBDF	semi-implicit backward differentiation formula
VS	variable stepsize
VSSBDF	variable stepsize, semi-implicit, backward differentiation formula
VSIMEX	variable stepsize Implicit-Explicit

Chapter 1

Introduction

1.1 An overview

Many important systems modeled by partial differential equations (PDEs) have natural additive splittings according to physical contributions, e.g., advection-diffusion-reaction (ADR) PDEs. ADR PDEs describe the movement of substances from an area of high concentration to an area of low concentration (diffusion), the chemical transformation of a set of substances into others (reaction), and the movement of substances from one region to another (advection). For instance, ARD PDEs are used to model air pollution [44, 66], combustion [11], the Heston model of financial mathematics [30], tumour angiogenesis and chemo-taxis [35], heat conduction [9, 10, 45], hemodynamics [22, 56], blood coagulation [15, 48], cardiac arrhythmias [19, 54], and atherosclerosis [25, 31].

ADR problems can be mathematically modeled using partial differential equations (PDEs), which, after spatial discretization, can be represented as initial value problems (IVPs) for systems of ordinary differential equations (ODE) in the form

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}(t) = \mathbf{F}(t, \mathbf{y}), \quad t \in [t_0, T], \tag{1.1a}$$

$$\mathbf{y}(t_0) = \mathbf{y}_0, \tag{1.1b}$$

where $t \in \mathbb{R}$ is time and the function $\mathbf{F} : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ represents the right hand side (RHS) of the ODE.

The concepts of *existence* and *uniqueness* are crucial in the analysis of IVPs [55]. There are instances where IVPs may have no solutions, and if there is a solution it may not be unique. In this work, we assume all IVPs feature a RHS that is continuous in t and exhibits

Lipschitz continuity in $\mathbf{y}$; meaning, $\forall t \in \mathbb{R}, \mathbf{y}_1 \in \mathbb{R}^m, \mathbf{y}_2 \in \mathbb{R}^m, \exists L \in \mathbb{R}$ such that $||\mathbf{f}(t, \mathbf{y}_1) - \mathbf{f}(t, \mathbf{y}_2)|| \leq L||\mathbf{y}_1 - \mathbf{y}_2||$ [6], which guarantees the uniqueness of solutions to IVPs. Additionally, the thesis assumes that the solutions to IVPs are stable, i.e., they are not sensitive to small changes in the model parameters or initial condition. All these assumptions define a *well-posed* problem [55].

Many ADR equations include nonlinear terms, and this leads to difficulties in obtaining analytic solutions. Therefore, approximating the solution via numerical methods is often the only feasible option. In general, popular numerical methods for solving ODEs are applied to the large systems of ODEs that result from the spatial discretization of ADR PDEs by, e.g., finite difference methods, finite element methods, finite volume methods, or pseudospectral methods [35].

A popular and effective approach to obtain a numerical solution to ADR PDEs is through the application of an implicit-explicit (IMEX) method. In a typical version of this approach, the diffusion and reaction terms are treated implicitly in time, and the advection terms are treated explicitly. The idea of IMEX splitting, which was proposed as a partitioned method, dates back to the late 1970s in [23, 32]. It was proposed as a multistep method in 1980 [20]. After that, many other classes of IMEX methods were studied. For instance, IMEX linear multistep methods (LMMs) were derived in [7, 51] and were investigated in many studies, e.g., [5, 21, 34, 35, 51, 58]. The stability properties of IMEX methods are often studied on the basis of stability regions defined by using complex scalar test equations; see [2, 3, 4, 7, 26, 51]. Higher-order methods and those with other enhanced numerical properties such accuracy, error analysis, monotonicity, and boundedness were studied in [35, 38, 47]. Classes of general linear IMEX methods were developed in [16, 17, 64, 65].

When splitting the RHS of IVPs according to their physical properties, this splitting approach is known as Physics-Based Splitting (PS). Alternatively, the right-hand side of IVPs can also be divided based on the mathematical characteristics of the problem. In this work, we introduce Jacobian Splitting (JS) as an alternative to the physics-based approach. This method is established by linearizing the IVP, allowing the linear and nonlinear terms to be addressed using different numerical methods [57]. A systematic comparison between these splitting approaches on a set of ADR models using IMEX Runge–Kutta methods was performed in [50].

The aim of employing numerical analysis in solving IVPs is to obtain a desirable accuracy level within a feasible time period. In any numerical method, the chosen stepsize is crucial in striking a balance between the time needed to produce a numerical solution and its error. A stepsize that is excessively large can lead to significant errors introduced by the numerical method. Conversely, a stepsize that is too small may result in unnecessary computations, rendering the process inefficient and costly. To address these issues, implementing a variable stepsize can be more effective than using a constant stepsize. In 2008, Wang and Ruuth proposed variable stepsize implicit-explicit linear multistep methods (VSIMEX-LMMs) for time-dependent PDEs [59]. The authors showed the performance of the methods using steps of variable size but not as part of an error-control algorithm. In this work, we use a family of variable stepsize semi-implicit backward differentiation formulae (VSSBDF) up to fourth order to solve a set of ADR problems using physics-based and Jacobian IMEX splitting. We also introduce an algorithm for adaptive time-stepping and error control [28], in which the local truncation error formula for the VSSBDF methods are derived as an extension of the one proposed in [62], who only considered the VSSBDF2 method and were more focused on its stability properties of the error controller applied to the Poisson–Nernst–Planck equations with generalized Frumkin–Butler–Volmer boundary conditions in computing steady-state solutions. We develop an adaptive stepping strategy that dynamically adjusts time step sizes in response to detected changes in the system's time scales, with the goal of ensuring computational accuracy to within predefined tolerances. The local truncation errors for VSSBDF methods up to the fourth order are derived, utilizing a step-doubling refinement technique for approximation of these errors.

In this thesis, we compare the performance of Jacobian splitting with the physics-based splitting by applying VSSBDF up to fourth order to the following set of ADR models:

- a one-dimensional advection-diffusion problem with both linear advection and linear diffusion terms[35].
- a one-dimensional nonlinear diffusion-reaction-advection problem[43],
- the one-dimensional CUSP problem, which is a diffusion-reaction equation that models a combination of Zeeman's "cusp catastrophe" and the Van der Pol oscillator with linear diffusion and non-linear reaction[35],
- a one-dimensional theoretical model of a combustion front that consists of linear ad-

vection, linear diffusion, and non-linear reaction [11],

- one- and two-dimensional Brusselator problems, which model autocatalytic reactions with linear both diffusion and non-linear reaction procedure [29],
- a two-dimensional heat transfer model, which model the transfer of heat in a long, fluid-filled, cylindrical pipe [36].

When using a class of 2-additive methods for dealing with ADR PDEs, this constraint necessitates a grouping of three physical processes into two, and could be undesirable depending on the specific problem. For example, if the reaction term is nonlinear and stiff, then treating diffusion and reaction implicitly together may be much less desirable than treating them separately. In the later case, the diffusion is often linear, and reaction term usually leads to a system of uncoupled ODEs; i.e., both of these sub-problems have specialized solvers associated with them that cannot be taken advantage of when combined. Furthermore, the relative importance of the terms may change during a simulation, again making the choice of an IMEX method sub-optimal. In this thesis, we investigate two classes of linear multistep methods that treat diffusion, reaction, and advection separately, i.e., 3-additive methods.

Consider the initial-value problem (IVP) for a system of ODEs of the form

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}(t) = \mathbf{f}^{[1]}(t,\mathbf{y}) + \mathbf{f}^{[2]}(t,\mathbf{y}) + \mathbf{f}^{[3]}(t,\mathbf{y}), \quad t \in [t_0, T], \tag{1.2a}$$

$$\mathbf{y}(t_0) = \mathbf{y}_0, \tag{1.2b}$$

where it is assumed that IVP (2.2) comes from a spatial discretization of an ADR problem and the functions $\mathbf{f}^{[i]} : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$, ($i = 1, 2$, and 3, and dimension $m \geq 1$), are assumed to be sufficiently smooth.

A 3-additive numerical solution of equation (2.2) can be found by writing the exact integration of the right-hand side from t_n to t_{n+1}:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \int_{t_n}^{t_{n+1}} \left(\mathbf{f}^{[1]}(t,\mathbf{y}(t)) + \mathbf{f}^{[2]}(t,\mathbf{y}(t)) + \mathbf{f}^{[3]}(t,\mathbf{y}(t))\right) dt$$

and then using quadrature to approximate the integrals of $\mathbf{f}^{[1]}$, $\mathbf{f}^{[2]}$, and $\mathbf{f}^{[3]}$ by different methods. For example, if the integral of the first term is approximated by backward Euler method, the integral of the second term is approximated by trapezoidal method, and the

integral of the last term is approximated by forward Euler method, one can obtain the following formula:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t\, \mathbf{f}^{[1]}(t_{n+1}, \mathbf{y}_{n+1}) + \frac{\Delta t}{2}\left(\mathbf{f}^{[2]}(t_{n+1}, \mathbf{y}_{n+1}) + \mathbf{f}^{[2]}(t_n, \mathbf{y}_n)\right) + \Delta t\, \mathbf{f}^{[3]}(t_n, \mathbf{y}_n). \quad (1.3)$$

The formula (1.3) is an example of 3-additive numerical method.

The functions $\mathbf{f}^{[1]}$, $\mathbf{f}^{[2]}$, and $\mathbf{f}^{[3]}$ correspond to the spatial discretizations of the diffusion, reaction, and advection terms, respectively. The function $\mathbf{f}^{[1]}$ is assumed to be a stiff term that is to be integrated implicitly, whereas $\mathbf{f}^{[3]}$ is assumed to be a nonstiff term that is to be integrated explicitly. The function $\mathbf{f}^{[2]}$ could be stiff or non-stiff and hence could be treated either implicitly or explicitly, yielding Implicit-Implicit-Explicit (IIE) or Implicit-Explicit-Explicit (IEE) methods, respectively.

In this part, we compare IIE-LMMs and IEE-LMMs with IMEX methods, in terms of stability and performance. We apply IMEX methods to the ODE of the form

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{g}(t, \mathbf{y}) + \mathbf{f}(t, \mathbf{y}). \quad (1.4)$$

In chapter 3, where IIE-LMMs are compared with IMEX methods, the diffusion and reaction terms are treated implicitly, and their discretizations are grouped together as part of $\mathbf{g}(t, \mathbf{y})$; the advection term is treated explicitly, and its discretization corresponds to $\mathbf{f}(t, \mathbf{y})$. In other words, $\mathbf{g}(t, \mathbf{y}) = \mathbf{f}^{[1]}(t, \mathbf{y}) + \mathbf{f}^{[2]}(t, \mathbf{y})$ and $\mathbf{f}(t, \mathbf{y}) = \mathbf{f}^{[3]}(t, \mathbf{y})$.

In chapter 4, where IEE-LMMs are compared with IMEX methods, the diffusion term is treated implicitly, and its discretization corresponds to $\mathbf{g}(t, \mathbf{y})$; the advection and reaction terms are treated explicitly, and their discretizations are grouped together as part of $\mathbf{f}(t, \mathbf{y})$. In other words, $\mathbf{g}(t, \mathbf{y}) = \mathbf{f}^{[1]}(t, \mathbf{y})$ and $\mathbf{f}(t, \mathbf{y}) = \mathbf{f}^{[2]}(t, \mathbf{y}) + \mathbf{f}^{[3]}(t, \mathbf{y})$.

1.2 Outline of the thesis

The outline of this thesis is organized as follows:

- Chapter 1 presents a briefer introduction to additive numerical splitting methods, and outlines the objectives and contributions of the thesis.
- Chapter 2 presents efficient VSSBDF up to the fourth order, along with error control

algorithms. It then utilizes these methods to compare the performance of two splitting approaches: Physics-based Splitting and Jacobian Splitting, across a set of tested ADR models.

- Chapter 3 provides the class of k-step IIE linear multistep methods, and study the stability analysis of IIE methods for a linear ADR PDE. The chapter also derives several IIE methods up to fourth order, and presents some numerical experiments that verify the convergence order and illustrate the performance of 3-additive splitting methods compared to popular IMEX methods.
- Chapter 4 provides the class of k-step IEE linear multistep methods, and study the stability analysis of IEE methods for a linear ADR PDE. The chapter also derives several IEE methods up to fourth order, and presents some numerical experiments that verify the convergence order and illustrate the performance of 3-additive splitting methods compared to popular IMEX methods.
- Chapter 5 generalizes the IIE and IEE approaches to N-additive methods.
- Chapter 6 contains a summary of this work and proposed future work.

Chapter 2

Performance Comparison of Variable-Stepsize IMEX SBDF Methods On Advection-Diffusion-Reaction Models

Consider the initial-value problems (IVPs)

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}(t) = \mathbf{F}(t, \mathbf{y}), \quad t \in [t_0, T], \tag{2.1a}$$

$$\mathbf{y}(t_0) = \mathbf{y}_0, \tag{2.1b}$$

where $t \in \mathbb{R}$ is time and the function $\mathbf{F} : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ represents the right-hand side (RHS) of the ODE.

In 2-additive splitting methods, the RHS of the ODE, $\mathbf{F}$, can be split in the form

$$\mathbf{F}(t, \mathbf{y}) = \mathbf{f}(t, \mathbf{y}) + \mathbf{g}(t, \mathbf{y}), \quad t \in [t_0, T], \tag{2.2a}$$

$$\mathbf{y}(t_0) = \mathbf{y}_0, \tag{2.2b}$$

where the functions $\mathbf{f}, \mathbf{g} : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$, with dimension $m \geq 1$, are considered to be sufficiently smooth. Here f represents the non-stiff (or mildly stiff) term of the equation, while g denotes the stiff term, which requires implicit integration. To take advantage of this structure, 2-additive numerical splitting methods were proposed (e.g., [51, 7]). Specifically, IMEX methods that treat g implicitly and f explicitly have proven to be of interest in many studies [51, 7, 37, 57]. In particular, we focus our study on IMEX versions of LMMS based on backward differentiation formulae (BDF) [51, 7, 8, 42]. These methods have proven to be highly effective for solving differential equations [60, 59, 62, 63, 61, 33].

A common form of 2-additive splitting of the RHS of an IVP (i.e., to determine functions f and g) is physics-based splitting where the terms are split based on their physical properties, e.g., advection, diffusion, or reaction.

It is also possible to establish an alternative splitting of the form (2.2) that does not depend on the physical properties of the problem but rather reflects the numerical properties of the solution. In this work, we study this approach, hereafter called Jacobian splitting, as an alternative to physics-based splitting for IVPs. Jacobian splitting is established by linearizing an IVP so that linear and non-linear terms can be integrated by different numerical techniques [57]. The Jacobian splitting for the problem (2.1a) is written as

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{J_F}\mathbf{y}(t) + [\mathbf{F}(t, \mathbf{y}(t)) - \mathbf{J_f}\mathbf{y}(t)], \tag{2.3}$$

where $\mathbf{J_F}$ is the Jacobian of $\mathbf{F}$ that is evaluated at a specific t and $\mathbf{y}(t)$, usually the beginning of each step t_n. Using the notation of (2.2), $\mathbf{g}(t, \mathbf{y}(t))$ corresponds to $\mathbf{J_F}\mathbf{y}(t)$, and $\mathbf{f}(t, \mathbf{y}(t))$ corresponds to $\mathbf{F}(t, \mathbf{y}(t)) - \mathbf{J_f}\mathbf{y}(t)$.

In this chapter, we compare the performance of physics-based splitting and Jacobian splitting on a set of seven test ADR models, using 2-additive IMEX-SBDF methods up to fourth order with a variable stepsize. The performance of the two splitting approaches is evaluated by comparing the CPU times for given accuracies of the SBDF methods for each model. We also extend a previous analysis of the second-order VSSBDF scheme to develop an adaptive stepping and error control strategy for numerical simulations. Our systematic approach allows for the dynamic adjustment of time step sizes in response to detected changes in time scales, ensuring the maintenance of computational accuracy within predefined tolerances. A key contribution of this work is the derivation of local truncation errors for VSSBDF up to the fourth order using a coarse-fine refinement strategy. We include details on the software framework developed for the implementation of these numerical methods and IVPs .

2.1 General variable stepsize IMEX-LMM

Let $\mathbf{y}_n$ represent the approximate solution at t_n, and let the stepsize $\Delta t_{n+j} = t_{n+j} - t_{n+j-1}$, $j = -k+2, \ldots, 1$. The general k-step VSIMEX-LMM for ODE (2.2) takes the form

$$\frac{1}{\Delta t_{n+1}} \sum_{j=-1}^{k-1} \alpha_j \mathbf{y}_{n-j} = \sum_{j=0}^{k-1} \beta_j \mathbf{f}(t_{n-j}\mathbf{y}_{n-j}) + \sum_{j=-1}^{k-1} \gamma_j \mathbf{g}(t_{n-j}\mathbf{y}_{n-j}), \tag{2.4}$$

where $\alpha_{-1}, \gamma_{-1} \neq 0$, and $k \geq 2$ The variable coefficients of $\alpha_j, \beta j$, and γ_j are functions of stepsize ratios $\omega_i = \frac{\Delta t_i}{\Delta t_{i-1}}$ for $i = n-k+3, \ldots, n+1$. These coefficients are assumed to be bounded and satisfy the order conditions derived in [59]. Because we are interested in the versions of SBDF methods with variable stepsize, we present the form for these methods up to fourth order.

VSSBDF methods

The one-step, first-order IMEX VSSBDF method can be given as

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t_{n+1} \left(\mathbf{f}(\mathbf{y}_n) + \mathbf{g}(\mathbf{y}_{n+1}) \right). \tag{2.5}$$

We refer to this method to as VSSBDF1.

The two-step, second-order IMEX VSSBDF method can be given as

$$\alpha_{-1}\mathbf{y}_{n+1} = \alpha_0 \mathbf{y}_n + \alpha_1 \mathbf{y}_{n-1} + \Delta t_{n+1} \left(\beta_0 \mathbf{f}(\mathbf{y}_n) + \beta_1 \mathbf{f}(\mathbf{y}_{n-1}) + \mathbf{g}(\mathbf{y}_{n+1}) \right),$$

where

$$\alpha_{-1} = \frac{1+2\omega_{n+1}}{1+\omega_{n+1}}, \quad \alpha_0 = 1+\omega_{n+1}, \quad \alpha_1 = -\frac{\omega_{n+1}^2}{1+\omega_{n+1}},$$
$$\beta_0 = 1+\omega_{n+1}, \qquad \beta_1 = -\omega_{n+1}.$$

We refer to this method to as VSSBDF2.

The three-step, third-order IMEX VSSBDF method can be given as

$$\alpha_{-1}\mathbf{y}_{n+1} = \alpha_0\mathbf{y}_n + \alpha_1\mathbf{y}_{n-1} + \alpha_2\mathbf{y}_{n-2} + \Delta t_{n+1}\left(\beta_0\mathbf{f}(\mathbf{y}_n) + \beta_1\mathbf{f}(\mathbf{y}_{n-1}) + \beta_2\mathbf{f}(\mathbf{y}_{n-2}) + \mathbf{g}(\mathbf{y}_{n+1})\right),$$

where

$$\begin{aligned}
\alpha_{-1} &= 1 + \frac{\omega_{n+1}}{1+\omega_{n+1}} + \frac{\omega_n\omega_{n+1}}{1+\omega_n\left(1+\omega_{n+1}\right)},\\
\alpha_0 &= 1 + \omega_{n+1} + \frac{\omega_n\omega_{n+1}\left(1+\omega_{n+1}\right)}{1+\omega_n},\\
\alpha_1 &= -\omega_{n+1}^2\left(\omega_n + \frac{1}{1+\omega_{n+1}}\right),\\
\alpha_2 &= \frac{\omega_n^3\omega_{n+1}^2\left(1+\omega_{n+1}\right)}{\left(1+\omega_n\right)\left(1+\omega_n+\omega_n\omega_{n+1}\right)},\\
\beta_0 &= \frac{\left(1+\omega_{n+1}\right)\left[1+\omega_n\left(1+\omega_{n+1}\right)\right]}{1+\omega_n},\\
\beta_1 &= -\omega_{n+1}\left[1+\omega_n\left(1+\omega_{n+1}\right)\right],\\
\beta_2 &= \frac{\omega_n^2\omega_{n+1}\left(1+\omega_{n+1}\right)}{1+\omega_n}.
\end{aligned}$$

We refer to this method to as VSSBDF3.

The four-step, fourth-order IMEX VSSBDF method can be given as

$$\begin{aligned}
\alpha_{-1}\mathbf{y}_{n+1} = \alpha_0\mathbf{y}_n + \alpha_1\mathbf{y}_{n-1} + \alpha_2\mathbf{y}_{n-2} + \alpha_3\mathbf{y}_{n-3} +\\
\Delta t_{n+1}\left(\beta_0\mathbf{f}(\mathbf{y}_n) + \beta_1\mathbf{f}(\mathbf{y}_{n-1}) + \beta_2\mathbf{f}(\mathbf{y}_{n-2}) + \beta_3\mathbf{f}(\mathbf{y}_{n-3})\right) + \Delta t_{n+1}\mathbf{g}(\mathbf{y}_{n+1}),
\end{aligned}$$

where

$$\alpha_{-1} = 1 + \frac{\omega_{n+1}}{1+\omega_{n+1}} + \frac{\omega_n\omega_{n+1}}{A_2} + \frac{\omega_{n-1}\omega_n\omega_{n+1}}{A_3},$$
$$\alpha_0 = 1 + \omega_{n+1}\left[1 + \frac{\omega_n\left(1+\omega_{n+1}\right)}{1+\omega_n}\left(1 + \frac{\omega_{n-1}A_2}{A_1}\right)\right],$$
$$\alpha_1 = -\omega_{n+1}\left[\frac{\omega_{n+1}}{1+\omega_{n+1}} + \omega_n\omega_{n+1}\frac{A_3+\omega_{n-1}}{1+\omega_{n-1}}\right],$$
$$\alpha_2 = \omega_n^3\omega_{n+1}^2\frac{1+\omega_{n+1}}{1+\omega_n}\frac{A_3}{A_2},$$
$$\alpha_3 = -\frac{1+\omega_{n+1}}{1+\omega_{n-1}}\frac{A_2}{A_1}\frac{\omega_{n-1}^4\omega_n^3\omega_{n+1}^2}{A_3},$$
$$\beta_0 = \frac{\omega_n\left(1+\omega_{n+1}\right)}{1+\omega_n}\frac{\left(1+\omega_{n+1}\right)\left(A_3+\omega_{n-1}\right) + \frac{1+\omega_{n-1}}{\omega_n}}{A_1},$$
$$\beta_1 = -A_2A_3\frac{\omega_{n+1}}{1+\omega_{n-1}},$$
$$\beta_2 = \omega_n^2\omega_{n+1}\frac{1+\omega_{n+1}}{1+\omega_n}A_3,$$
$$\beta_3 = -\omega_{n-1}^3\omega_n^2\omega_{n+1}\frac{1+\omega_{n+1}}{1+\omega_{n-1}}\frac{A_2}{A_1},$$
$$A_1 = 1+\omega_{n-1}\left(1+\omega_n\right), \quad A_2 = 1+\omega_n\left(1+\omega_{n+1}\right), \quad A_3 = 1+\omega_{n-1}A_2.$$

We refer to this method to as VSSBDF4.

2.2 Adaptive Time Stepping and Error Control

Unlike the constant stepsize strategy, the variable stepsize approach determines the time stepsize of the next iteration based on the previously calculated solutions and time stepsizes. In particular, it determines the next stepsize Δt_{n+1} in (2.4) by approximating the local truncation error (LTE) ϵ_{n+1}^c. The target is to choose a Δt_{n+1} such that $\epsilon_{n+1}^c \in (tol - range, tol + range)$, where tol is the tolerance giving a maximal allowable numerical error and $range = tol/5$.

In this work, we used a step-doubling refinement technique to approximate the local truncation error [62, 28]. This technique involves initially solving the problem on a coarse grid, which provides a rough approximation of the solution. We then refine the stepsize by a factor of two to obtain a more accurate solution on a finer grid. Specifically, the LTE is approximated by employing a single "coarse" time step, denoted as y_{n+1}^c, and two "fine" half time steps, resulting in y_{n+1}^f; see Figures 2.1–2.2. The formulae to approximate LTEs for

VSSBDF up to fourth order are derived in A and summarized in Table 2.1.

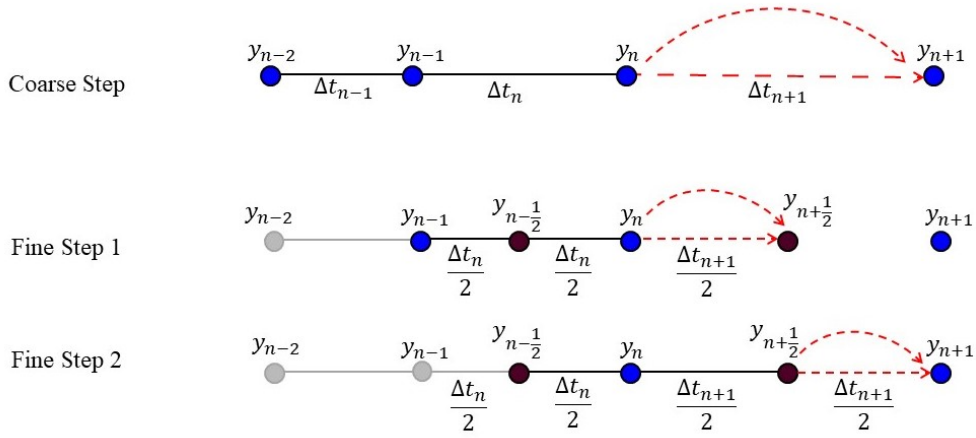


Figure 2.1: Step-doubling time-stepping method for the VSSBDF3 method. First, a coarse time-step uses y_{n-2}, y_{n-1}, and y_n to compute y^c_{n+1}. After that, a fine time-step uses $y^f_{n-1/2}$ (with y_{n-1} and y_n) to create $y^f_{n+1/2}$, and another fine time-step uses $y^f_{n+1/2}$ (with y_n and $y_{n-1/2}$) to calculate y^f_{n+1}.

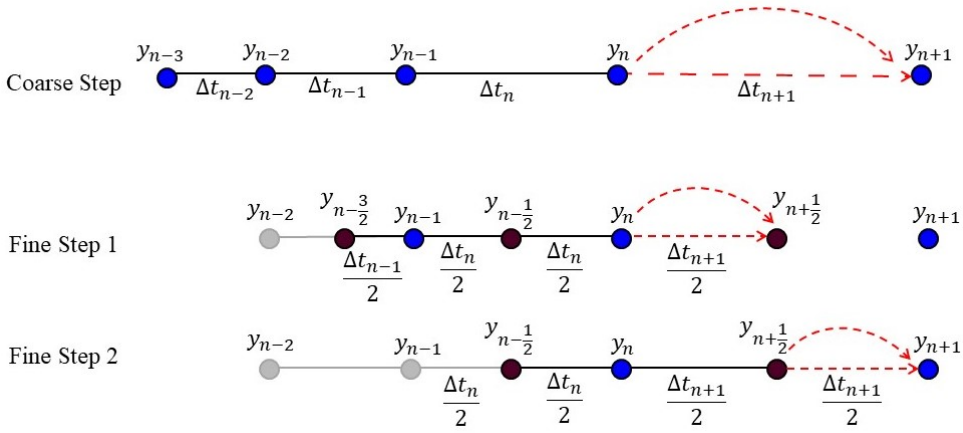


Figure 2.2: Step-doubling time-stepping method for the VSSBDF4 method. First, a coarse time-step uses $y_{n-3}, \mathbf{y}_{n-2} \ldots, y_n$ to compute y^c_{n+1}. After that, a fine time-step uses $y^f_{n-1/2}$ (with $y_{n-3/2}$, y_{n-1} and y_n) to create $y^f_{n+1/2}$, and another fine time-step uses $y^f_{n+1/2}$ (with y_{n-1}, $y_{n-1/2}$, and y_n) to calculate y^f_{n+1}.

We use this approximation of ϵ^c_{n+1} to test whether to admit the Δt_{n+1} or to coarsen or refine it. If Δt_{n+1} is accepted, then we can use y^c_{n+1} to give better approximation of y_{n+1}, which is $y(t_{n+1}) = y^c_{n+1} - \epsilon^c_{n+1}$.

The algorithm of an adaptive time-stepping k-step VSSBDF method is given in Algorithm

H

Table 2.1: The Local Truncation Errors for VSSBDF methods

Method	LTE Approximation
VSSBDF1	$2(y^c - y^f)$
VSSBDF2	$\frac{8(\Delta t_{n+1}+\Delta t_n)}{5\Delta t_{n+1}+7\Delta t_n}\left(y^c - y^f\right)$
VSSBDF3	$\frac{16(\Delta t_n)(\Delta t_{n+1}+\Delta t_n+\Delta t_{n-1})(y^c-y^f)}{\left(15\,\Delta t_n^2+14\,\Delta t_n\Delta t_{n-1}+30\,\Delta t_n\Delta t_{n+1}+10\,\Delta t_{n-1}\Delta t_{n+1}+11\,\Delta t_{n+1}^2\right)}$
VSSBDF4	$\frac{32(\Delta t_{n+1}+\Delta t_n)(\Delta t_{n+1}+\Delta t_n+\Delta t_{n-1})(\Delta t_{n+1}+\Delta t_n+\Delta t_{n-1}+\Delta t_{n-2})(y^c-y^f)}{C_1+C_2}$ †

† C_1 and C_2 are given in A.16 and A.17, respectively.

1. The new time stepsize is updated using the formula given in [28]:

$$\Delta t_{n+1} \leftarrow \min\left(\max\left(\alpha\left(\frac{tol}{\epsilon_i^c}\right)^{1/(p+1)}, \eta_{\min}\right), \eta_{\max}\right)\Delta t_{\mathrm{n+1}}, \tag{2.6}$$

where p represents the order of the method and $\alpha \in (0, 1]$ represents a safety factor. In this work, we choose $\alpha = 0.8$, $range = tol/5$, $\eta_{\max} = 1.3$, and $\eta_{\min} = 0.5$. These values are used throughout this work unless specified otherwise.

Algorithm 1 Adaptive time-stepping method for a single time step of k-step VSSBDF method

$\Delta t_{n+1} \leftarrow$ next Δt ▷ the initial Δt_{n+1} is next Δt, computed in the previous time step
$accepted \leftarrow false$
while not *accepted* **do** ▷ Loop until the error is acceptable
  $y_{n+1}^c \leftarrow \text{TimeStep}(\Delta t_{n+1}, \Delta t_n, \ldots, \Delta t_{n-k+2})$ ▷ Coarse step: TimeStep()
  $y_{n+1}^f \leftarrow \text{TimeStep}(\frac{\Delta t_{n+1}}{2}, \ldots, \frac{\Delta t_{n+1-\lfloor k/2 \rfloor}}{2})$ ▷ Fine time step
  $\epsilon_i^c \leftarrow \text{Error}(\Delta t_{n+1}, \Delta t_n, \ldots, \Delta t_{n-k+2}, y_{n+1}^c, y_{n+1}^f)$ ▷ Error() from Table (2.1)
  $accepted \leftarrow (\epsilon_i^c - tol) \leq range$ ▷ Check if Δt_{n+1} is accepted
  $factor \leftarrow \min\left(\max\left(\alpha\left(\frac{\text{tol}}{\epsilon_i^c}\right)^{1/(p+1)}, \eta_{\min}\right), \eta_{\max}\right)$
  if (*accepted* and $|\epsilon_i^c - tol| > range$) **then** ▷ Update next Δt only if $|\epsilon_i^c - tol| > range$
    next $\Delta t \leftarrow \Delta t_{n+1} \cdot factor$ ▷ Update next Δt for the next time step
  end if
  if (not *accepted*) **then** ▷ Δt_{n+1} is updated if error is not acceptable
    $\Delta t_{n+1} \leftarrow \Delta t_{n+1} \cdot factor$
  end if
end while
$y_{n+1} \leftarrow y_{n+1}^c - \epsilon^c$

2.3 Description of the software

All the numerical methods and IVPs have been implemented in software framework using the C++ programming language. The software framework includes several subclasses to represent the IVP models, a class to implement the VSSBDF solvers, and a driver program to perform all numerical experiments.

The driver program instantiates the different classes and calls the methods of the corresponding objects to run the experiments associated with a given IVP model and generate the numerical errors for both splitting approaches, the execution times, and the solutions at the final simulation times. In the driver program, the initialization of the intermediate pre-steps (from the initial conditions), which are necessary to start a particular VSSDF method, are performed by using the fourth-order explicit Runge–Kutta method with a small stepsize.

The underlying code will be publicly accessible as a GitHub repository with support that allows its easy and efficient installation on platforms running Linux.

2.3.1 Implementation of the IVP models

The IVPs representing the different mathematical models are implemented using a class hierarchy. In order to avoid redundancy, each IVP is defined as a subclass of a C++ abstract class called IVP_ODE (see figure 2.3). Class IVP_ODE defines the virtual functions to be implemented in the subclasses as well as functions which are common to all the IVPs (to compute $\mathbf{F}(t, \mathbf{y})$ in (2.2a) from $\mathbf{f}(t, \mathbf{y})$ and $\mathbf{g}(t, \mathbf{y})$, to get the number of ODEs, to initialize the sparse matrices, etc.).

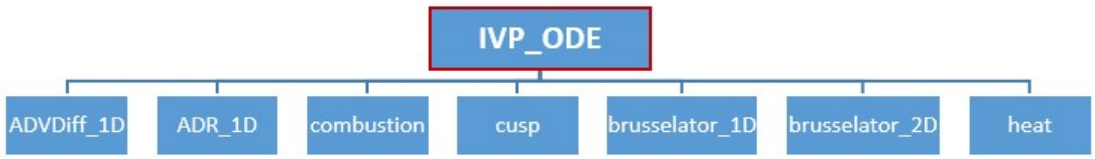


Figure 2.3: Class hierarchy for the IVPs

Each subclass defines the functions that are specific of each particular IVP model:

- A function to compute the initial conditions $\mathbf{y}(t_0)$.
- The functions required to compute the functions $\mathbf{f}(t, \mathbf{y})$ and $\mathbf{g}(t, \mathbf{y})$ (see 2.2a).
- The functions to compute the Jacobian matrix of $\mathbf{g}$ and $\mathbf{F}$ (see 2.2a).

In the current implementation, the sparse Jacobian matrices that are needed for each splitting approach are computed analytically by defining a specific function that depends on the particular IVP. However, we are working on defining an alternative generic function to approximate numerically the sparse Jacobian matrix without the need to explicitly program the function for each IVP model. This generic function would analyze the sparsity pattern of the Jacobian as a preliminary step in the computation.

2.3.2 Implementation of the VSSBDF solvers

We have defined a specific C++ class to implement the VSSBDF solvers with support for orders 1 to 4. The solvers apply adaptive time-stepping techniques to optimize the computational process where the accuracy is ensured by error checking (see section 2.2). The following public functions are highlighted in this class:

- **Time_Step** function describes the computations necessary to perform a time step of a kth-order SBDF method. This function has an argument which is a pointer to a IVP_ODE object and uses a specific argument to choose between physics-based and Jacobian-based splitting approaches.

  This function uses functions of LIS (Library of Iterative Solvers) [46, 39] to perform certain operations with sparse matrices and iteratively solve linear systems. The function also calls functions of an implementation of the Basic Linear Algebra Subprograms (BLAS) library [13] called *openblas* to perform various vector operations. A modified Newton iteration method is used to solve the nonlinear system (see (2.4)) at each time step. To solve the sparse linear systems that arise, we have used the Flexible GMRES (FGMRES) method [52] using the ILUT preconditioner [53] because we have obtained good performance with these choices for the LIS iterative solver function.
- **Variable_Time_Step** function implements one coarse time step and two fine time steps followed by the approximation of the local truncation error (see Algorithm 1).
- **Adaptive_dt_SBDF_Integrate** implements the entire Algorithm 1.

2.4 Test suite of problems

In this section, we present seven ADR models that are widely used in the fields of physics and engineering.

2.4.1 One-Dimensional Advection-Diffusion

The one-dimensional advection-diffusion model is used to describe the transport of a quantity inside a physical system due to both advection and diffusion processes. Here, we consider a one-dimensional advection-diffusion equation with constant coefficients. The advection-diffusion equation for a quantity y is written

$$\begin{aligned} \frac{\partial y}{\partial t} + a\frac{\partial y}{\partial x} &= d\frac{\partial^2 y}{\partial x^2}, \\ y(0,x) &= \sin(2\pi x), \\ y(t,0) &= y(t,1), \end{aligned} \tag{2.7}$$

where $x \in [0,1]$, and $t \in [0,1]$. $a \in \mathbb{R}$ and $d \in \mathbb{R}$ are constant scalars representing the strength of advection and diffusion, respectively. For the simulations presented here we used $a = \{1, 10\}$ and $d = \{1, 10\}$. The spatial domain is uniformly divided into N sub-intervals with a uniform spacing mesh $\Delta x = 1/N$; using a second-order centered finite difference method, the system of ordinary differential equations (ODEs) can be written as follows:

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = d\left(\frac{y_{i+1} - 2y_i + y_{i-1}}{(\Delta x)^2}\right) - a\left(\frac{y_{i+1} - y_{i-1}}{\Delta x}\right), \tag{2.8}$$

where $i = 1, 2, \ldots, N$. Since the boundary conditions are periodic, we impose $y_0(t) = y_N(t)$ and $y_{N+1}(t) = y_1(t)$.

2.4.2 One-Dimensional Advection-Diffusion-Reaction Model

We consider the nonlinear ADR model [43]

$$\begin{aligned} u_t + a\,uu_x &= d\,u_{xx} + u + f(x,t), \\ u(x,0) &= \sin(2\pi x), \\ u(0,t) &= u(1,t), \end{aligned} \tag{2.9}$$

where

$$f(x,t) = \cos(2\pi x + t) + 2a\pi \sin(2\pi x + t)\cos(2\pi x + t) + 4d\pi^2 \sin(2\pi x + t) - \sin(2\pi x + t)$$

has been chosen such that (3.24) has the exact solution $u(x,t) = \sin(2\pi x + t)$. The scalar coefficients a and d represent the strength of advection and diffusion terms, respectively. We study this model in two cases: $a = 10, d = 1$ and $a = 100, d = 10$, $t \in [0,1]$ and $x \in [0,1]$. One can rewrite equation (3.24) as $u_t + (\frac{u^2}{2})_x = u_{xx} + u + f(x,t)$. The spatial domain is uniformly discretized into N points with spacing $\Delta x = 1/N$; a second-order centered finite discretization for advection and diffusion terms provides a system of ODEs written as

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\left(\frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x}\right) + \left(\frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}\right) + u_i + f(i\Delta x, t), \tag{2.10}$$

where $i = 1, 2, \ldots, N$.

2.4.3 Combustion models

Combustion models are examples of advection-diffusion-reaction equations that model the conservation of energy, mass, and momentum [24].

In this work, we consider a simple form for combustion model [12], where a combustion reaction is represented by a single PDE. The simplified model can be given as

$$\frac{\partial c}{\partial t} = \left[1 + U_0 \sin\left(\frac{\pi x}{L}\right)\right]\left(\gamma \frac{\partial^2 c}{\partial x^2} - \frac{\partial c}{\partial x}\right) + f(c), \tag{2.11}$$

where $x \in [10, 50]$, c represents the mass fraction of the reacted products, $f(c)$ is the reaction model, the parameter $U_0 \in [0,1]$ is utilized to regulate changes in the density and velocity

within the fluid domain, where L denotes the length scale of the domain. The parameters γ and β represent the average diffusivity. Building on the numerical experiments in [12], the model has been solved for $U_0 \in \{0, 0.75, 0.99\}$ and for the Fisher–Kolmogorov–Petrovskii–Piskunov (FKPP), ignition, and general order-m Fisher reaction models. The three reaction terms are given, respectively, as

$$\begin{aligned} f_{\text{FKPP}}(c) &= \alpha_1 c(1-c), \\ f_{\text{ignition}}(c) &= \alpha_2 (1-c)(c-c_s)^+, \\ f_{\text{Fisher}}(c) &= \alpha_3 c^m (1-c), \end{aligned}$$

where α_1, α_2, and α_3 represent the rates of reaction, c_s is a reaction threshold, $(\cdot)^+ = \max(0, \cdot)$, and m represents the general Fisher non-linearity [12]. The relation $\max(f(c)) = \frac{\alpha_1}{4}$, given by [12], is used to calculate the values of α_2 and α_3, so that each of the three reaction types has a roughly equivalent magnitude. The calculations are written as follows:

$$\begin{aligned} f'_{\text{ignition}}(c) = 0 \qquad &\implies c = \frac{c_s + 1}{2} \qquad &&\implies \alpha_2 = \frac{\alpha_1}{(1-c_s)^2}; \\ f'_{\text{Fisher}}(c) = 0 \qquad &\implies c = \frac{m}{m+1} \qquad &&\implies \alpha_3 = \frac{\alpha_1}{4(\frac{m}{m+1})^m (1 - \frac{m}{m+1})}. \end{aligned}$$

Boundary conditions are periodic. The initial condition of the system is given as

$$c(x, 0) = \exp\left[-\frac{(x - x_0)^2}{\sigma_0} \right].$$

The spatial domain of this model is uniformly discretized into N subintervals, yielding the ODEs

$$\begin{aligned} \frac{dc_i}{dt} = &\left[1 + U_0 \sin\left(\frac{i\pi\Delta x}{L} \right) \right] \left[\frac{\gamma}{(\Delta x)^2} (c_{i-1} + c_i + c_{i+1}) + \frac{1}{2\Delta x} (c_{i+1} - c_{i-1}) \right] \\ &+ f(c_i), \end{aligned}$$

where $i = 1, 2, \ldots, N$, the boundary conditions are applied as $c_0(t) = c_N(t)$ and $c_{N+1}(t) = c_1(t)$, and initial condition are applied as $c_i(0) = \exp[-(x - x_i)^2/\sigma_0]$. This experiment uses parameters $L = 1$, $\alpha_1 = 0.1$, $\gamma = 0.1$, $c_s = 0.6$, $m = 10$, and $\sigma_0 = 10$.

2.4.4 The CUSP Model

The CUSP model is a combination of Zeeman's CUSP catastrophe model and the van der Pol oscillator model [28], resulting three coupled diffusion-reaction equations PDEs with linear (non-coupled) diffusion and non-linear reactions. The CUSP model is defined over the domain $x \in [0, 1]$ and is defined as

$$\begin{aligned}
\frac{\partial y}{\partial t} &= \sigma \frac{\partial^2 y}{\partial x^2} - \frac{1}{\epsilon}(y^3 + ay + b), \\
\frac{\partial a}{\partial t} &= \sigma \frac{\partial^2 a}{\partial x^2} + b + \frac{0.07(y - 0.7)(y - 1.3)}{(y - 0.7)(y - 1.3) + 0.1}, \\
\frac{\partial b}{\partial t} &= \sigma \frac{\partial^2 b}{\partial x^2} + (1 - a^2)b - a - 0.4y + \frac{0.035(y - 0.7)(y - 1.3)}{(y - 0.7)(y - 1.3) + 0.1},
\end{aligned} \tag{2.12}$$

where σ represents the diffusivity and the parameter ϵ is used to control the stiffness of the reaction. It can be noted that the reaction term in y becomes more stiff as ϵ goes to zero. These PDEs are uniformly discretized into N grid points using second-order centred differences. They can be represented by the following system of ODEs:

$$\begin{aligned}
\frac{\mathrm{d}y_i}{\mathrm{d}t} &= \sigma \left(\frac{y_{i-1} - 2y_i + y_{i+1}}{(\Delta x)^2} \right) - \frac{1}{\epsilon}(y_i^3 + a_i y_i + b_i), \\
\frac{\mathrm{d}a_i}{\mathrm{d}t} &= \sigma \left(\frac{a_{i-1} - 2a_i + a_{i+1}}{(\Delta x)^2} \right) + b_i + \frac{0.07(y_i - 0.7)(y_i - 1.3)}{(y_i - 0.7)(y_i - 1.3) + 0.1}, \\
\frac{\mathrm{d}b_i}{\mathrm{d}t} &= \sigma \left(\frac{b_{i-1} - 2b_i + b_{i+1}}{(\Delta x)^2} \right) + (1 - a_i^2)b_i - a_i - 0.4y_i + \frac{0.035(y_i - 0.7)(y_i - 1.3)}{(y_i - 0.7)(y_i - 1.3) + 0.1},
\end{aligned}$$

where $i = 1, 2, \ldots, N$ and $\Delta x = 1/N^2$. The boundary conditions are periodic and given as

$$\begin{aligned}
y_0(t) &= y_N(t), & a_0(t) &= a_N(t), & b_{N+1}(t) &= b_1(t), \\
y_{N+1}(t) &= y_1(t), & a_{N+1}(t) &= a_1(t), & b_{N+1}(t) &= b_1(t).
\end{aligned}$$

Initial conditions are given as

$$y_i(0) = 0, \qquad a_i(0) = -2\cos\left(\frac{2i\pi}{N}\right), \qquad b_i(0) = 2\sin\left(\frac{2i\pi}{N}\right).$$

The CUSP problem has been run for 32 spatial grid points, producing a system of ODEs with 96 unknowns. In this experiment, $\sigma = 1/144$, $\epsilon = 10^{-4}$, and the final simulation time

$t_\mathrm{f} = 1.1$.

2.4.5 1D The Brusselator

The Brusselator system, a type of reaction-diffusion equation, was developed by the Brussels School [40]. This system is specifically designed to analyze the behaviors of chemical systems that exhibit non-linear oscillation, as detailed in [35]. This model is characterized by two coupled diffusion-reaction equations, featuring linear diffusion coupled with non-linear reaction dynamics. They are written as

$$\begin{aligned} \frac{\partial u}{\partial t} &= \alpha \nabla^2 u + A + u^2 v - (B+1)u, \\ \frac{\partial v}{\partial t} &= \alpha \nabla^2 v + Bu - u^2 v, \end{aligned} \tag{2.13}$$

where A and B are parameters controlling the reaction terms and α represents the diffusivity. In this example, we consider the one-dimensional Brusselator model. If $B > 1 + A^2$, the solution is oscillatory; otherwise, it approaches an equilibrium over time.

The spatial domain is given by $x \in [0, 1]$ and ∇ simplifies to $\frac{\partial}{\partial x}$. Prescribed initial and Dirichlet boundary conditions are written as

$$\begin{aligned} u(0,t) &= u(1,t) = 1, & v(0,t) &= v(1,t) = 3, \\ u(x,0) &= 1 + \sin(2\pi x), & v(x,0) &= 3. \end{aligned}$$

Applying a uniform second-order centered differences with N unknown grid points, the one-dimensional Brusselator equation takes the form

$$\begin{aligned} \frac{\mathrm{d}u_i}{\mathrm{d}t} &= \alpha(N+1)^2(u_{i-1} - 2u_i + u_{i+1}) + A + u_i^2 v_i - (B+1)u_i, \\ \frac{\mathrm{d}v_i}{\mathrm{d}t} &= \alpha(N+1)^2(v_{i-1} - 2v_i + v_{i+1}) + Bu_i - u_i^2 v_i, \end{aligned}$$

where $i = 1, 2, \ldots, N$. The initial and boundary conditions are written as

$$\begin{aligned} u_0(t) &= u_{N+1}(t) = 1, & v_0(t) &= v_{N+1}(t) = 3, \\ u_i(0) &= 1 + \sin\left(\frac{2\pi i}{N+1}\right), & v_i(0) &= 3. \end{aligned}$$

In this experiment, $A = 1$, $B = 3$, and $\alpha \in \{1/50, 1/500\}$,

2.4.6 2D Brusselator

The two-dimensional form of the Brusselator model closely resembles its one-dimensional counterpart, but it includes an extra forcing (reaction) term, denoted as f, which influences u. In the context of the 2D model, the parameters are chosen as follows: $A = 1$, $B = 3.4$, with α varying between $1/50, 1/500$, and both x and y ranging from $[0, 1]$. The initial conditions are defined as

$$u(x, y, 0) = 22y(1 - y)^{3/2}, \qquad v(x, y, 0) = 27x(1 - x)^{3/2},$$

and the boundary conditions are considered to be periodic. The Brusselator model is discretized uniformly into a grid of $N \times N$ points, employing second-order centered differences. The model with $N \times N$ unknown grid points is described accordingly.

$$\begin{aligned}
\frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} &= \alpha N^2(u_{i,j-1} + u_{i-1,i} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}) + f_{i,j}(t) + A + u_{i,j}^2 v_{i,j} - (B + 1)u_{i,j}, \\
\frac{\mathrm{d}v_{i,j}}{\mathrm{d}t} &= \alpha N^2(v_{i,j-1} + v_{i-1,i} - 4v_{i,j} + v_{i+1,j} + v_{i,j+1}) + Bu_{i,j} - u_{i,j}^2 v_{i,j},
\end{aligned}$$

where $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N$, and the forcing term $f_{i,j}(t)$ is written

$$f_{i,j}(t) = \begin{cases} 5 & \text{if } (x_i - 0.3)^2 + (y_j - 0.5)^2 \leq 0.1^2 \text{ and } t \geq 1.1, \\ 0 & \text{otherwise,} \end{cases}$$

where $x_i = i\Delta x$ and $y_j = j\Delta y$. This yields to system of ODEs of size $2N^2$. The initial conditions on the discretized grid are given by

$$u_{i,j}(0) = 22y_j(1 - y_j)^{3/2}, \qquad v_{i,j}(0) = 27x_i(1 - x_i)^{3/2}.$$

2.4.7 Two-Dimensional Heat Transfer Model

A heat transfer model in a long, fluid-filled, cylindrical pipe characterized by a predefined velocity distribution is presented in [36]. This model is a two-dimensional linear advection-diffusion PDE. The heat transfer models is commonly used in the application of mechanical

engineering. In this model, heat transfer along the pipe by moving with the fluid (advection) and spreading out (diffusion). Initially, the inside of the pipe is set to a steady, cooler temperature. The heat entering the pipe is higher, and while we know how easily the pipe's walls can transfer heat, the model aims to find out what the temperature will be at the end of the pipe after a certain amount of time.

The pipe is characterized by a length l and a diameter (height) h. It is oriented along the x-direction, setting the domain to $x \in [0, l]$ and $y \in [0, h]$. The PDE for the temperature T, along with its initial and boundary conditions, is defined as follows.

$$\begin{aligned} \frac{\partial T}{\partial t} &= d\,\nabla^2 T - \mathbf{u} \cdot \nabla T, \\ \frac{\partial T}{\partial y}(t, x, 0) &= -\frac{\partial T}{\partial y}(t, x, h) = \sigma, \\ T(t, 0, y) &= T_{\text{in}}, \qquad T(0, x, y) = T_0, \end{aligned} \tag{2.14}$$

where d is the diffusivity of heat, $\mathbf{u}$ represents the flow velocity within the pipe, T_{in} is the input temperature, T_0 represents the pipe's initial temperature, and σ describes the heat transfer through the pipe wall. Note that boundary condition at $x = l$ is considered as part of the solution as no prescribed information is given at that border.

The fluid moves along the length of the pipe, implying that there is no movement in the y-direction and that the movement in the x-direction varies according to the distance to the pipe wall. The fluid's velocity, denoted as $\mathbf{u}$, is determined by the average velocity u_{avg} as

$$\mathbf{u} = \begin{pmatrix} \frac{3}{2} u_{\text{avg}} \left[1 - \left(\frac{y}{h/2} \right)^2 \right] \\ 0 \end{pmatrix},$$

where $u_{\text{avg}} = \frac{4.806 \times 10^{-6} \times h^2}{12 \times 8.9 \times 10^{-4}}$.

Equation (2.14) is discretized uniformly on a two-dimensional grid, where the mesh points being divided into N_x points in the x-direction and N_y points in the y-direction. These points are located at the center-right within each cell. By employing second-order centered differences for the second derivatives and first-order backward differences for the first derivatives, we obtain a discretized system that can be described as follows:

$$\frac{\mathrm{d}T_{ij}}{\mathrm{d}t} = d\left(\frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{(\Delta x)^2} + \frac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{(\Delta y)^2}\right) - \frac{3}{2}u_{\text{avg}}\left[1 - \left(\frac{(i+\frac{1}{2})\Delta y}{h/2}\right)^2\right]\frac{T_{i,j} - T_{i-1,j}}{\Delta x}$$

where $i = 1, 2, \ldots, N_x$, $j = 1, 2, \ldots, N_y$, $\Delta x = \frac{l}{N_x}$, and $\Delta y = \frac{h}{N_y}$. The boundary values are, therefore, written

$$T_{0,j}(t) = T_{\text{in}}(t), \qquad T_{i,0}(t) = T_{i,1}(t) - (\Delta y)\sigma, \qquad T_{i,N+1}(t) = T_{i,N}(t) - (\Delta y)\sigma.$$

The heat transfer problem is evaluated with three sets of spacial grid points: 70×10, 140×20, and 210×30. The problem parameters are chosen such that $d = 1.38 \times 10^{-7}$, $T_0 = 298.15$, $T_{in} = 303.15$, and $\sigma = 0$. The experiment is run with the final simulation time $t_f = 10$ and over the spacial domain $[0, 0.1] \times [0, 0.7]$.

2.5 Verification of the models

This section discusses the verification of the numerical simulations, integral to ensure the accuracy and reliability of computational models. As part of the verification, solution for each of the numerical experiments are presented graphically and compared with the graphs of reference solutions showing that the models are consistent with physical reality.

The solution plots for the one-dimensional advection diffusion problem defined by (2.7) using 200 points is presented in Figure 2.4. It can be evident that the initial condition is set as a harmonic wave. Observe how the advection process shifts the wave, while the diffusion process make it a smoother appearance.

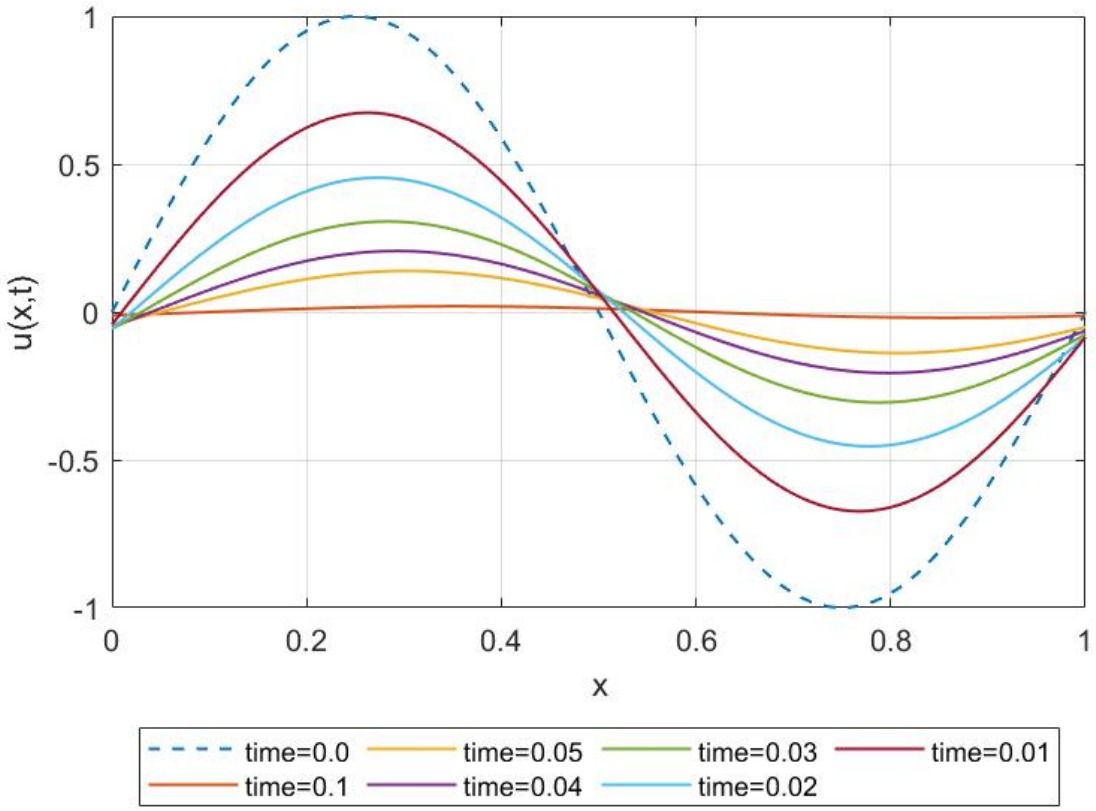


Figure 2.4: Solution plots for the linear advection-diffusion equation defined by (2.7), spatially discretized using 200 points

Solution plots for the one-dimensional advection diffusion reaction problem defined by (3.24) using 200 discretized points is presented in the Figure 2.5. This numerical solution

matches the exact solution $u(x,t) = \sin(2\pi x + t)$. It can be evident that the initial condition is set as a harmonic wave. Observe how the advection process shifts the wave, while the diffusion process make it a smoother appearance.

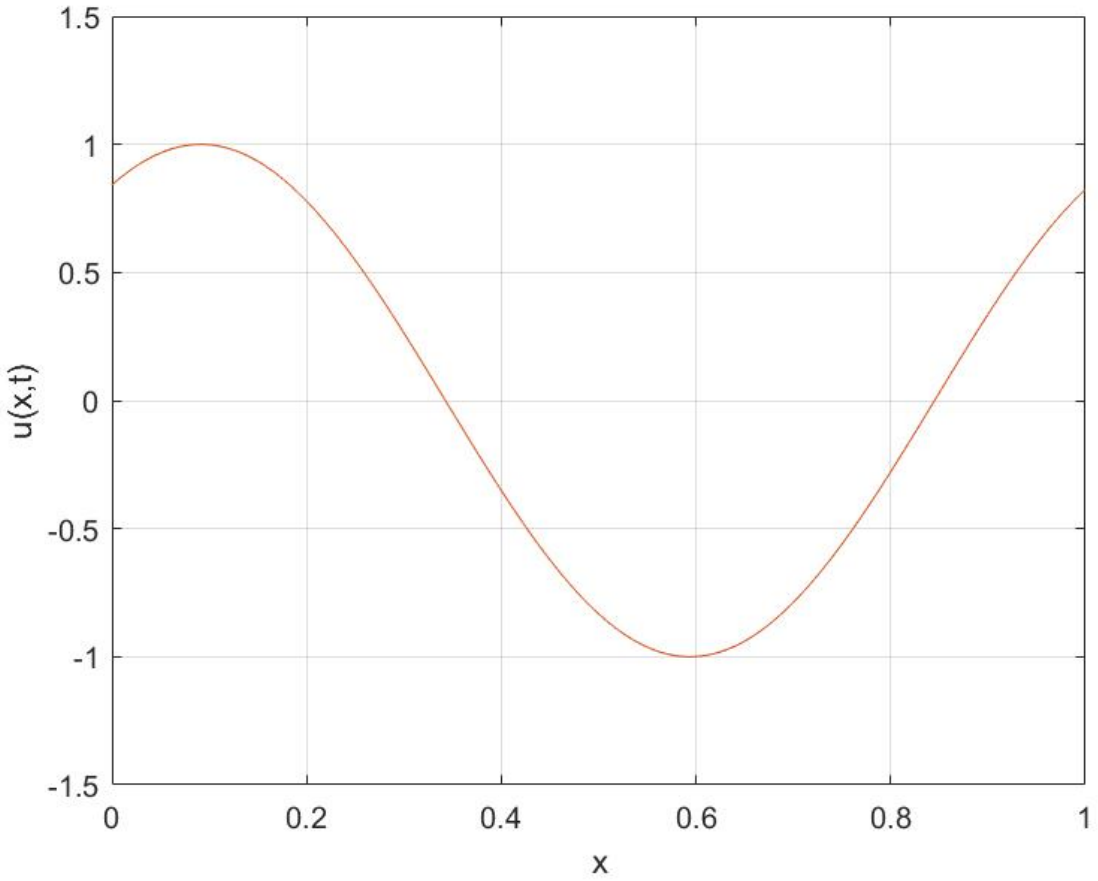


Figure 2.5: Solution plots for the linear advection-diffusion equation defined by 3.24, spatially discretized using 200 points at $t = 1$

The solution plots for the combustion model defined by 2.11 are presented in figure 2.6; The model was solved over a combination of parameters and reaction types. These graphs match the results of the numerical experiment in [11].

The solution plot for the CUSP model defined by 2.12 is presented in Figure 2.7. This plot matches results in [29]

Figure 2.9 shows the solution plots of the two-dimensional Brusselator problem defined by (2.13), which matches results in [49].

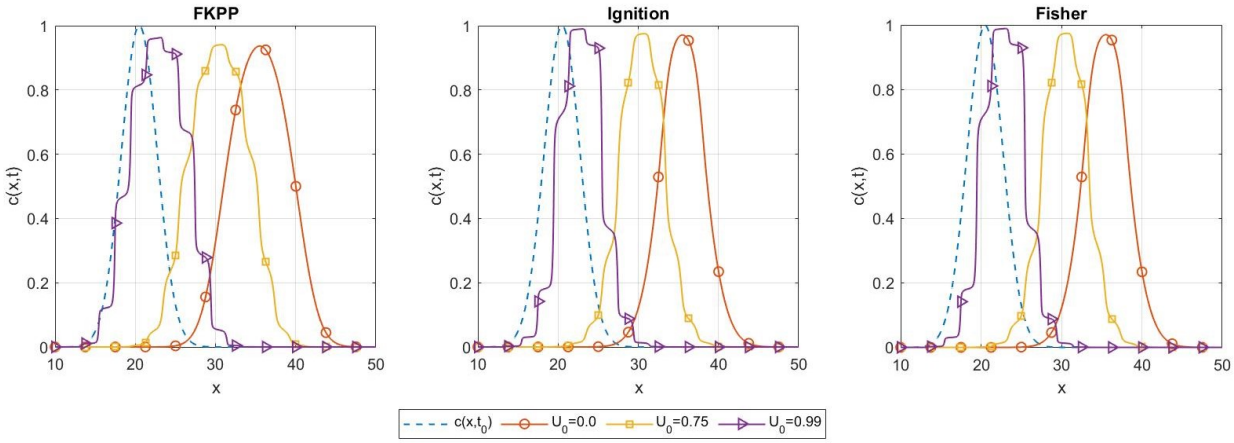


Figure 2.6: Solution plots for the combustion equations defined by (2.11), using 500 discretized points after a transient of $t = 15$.

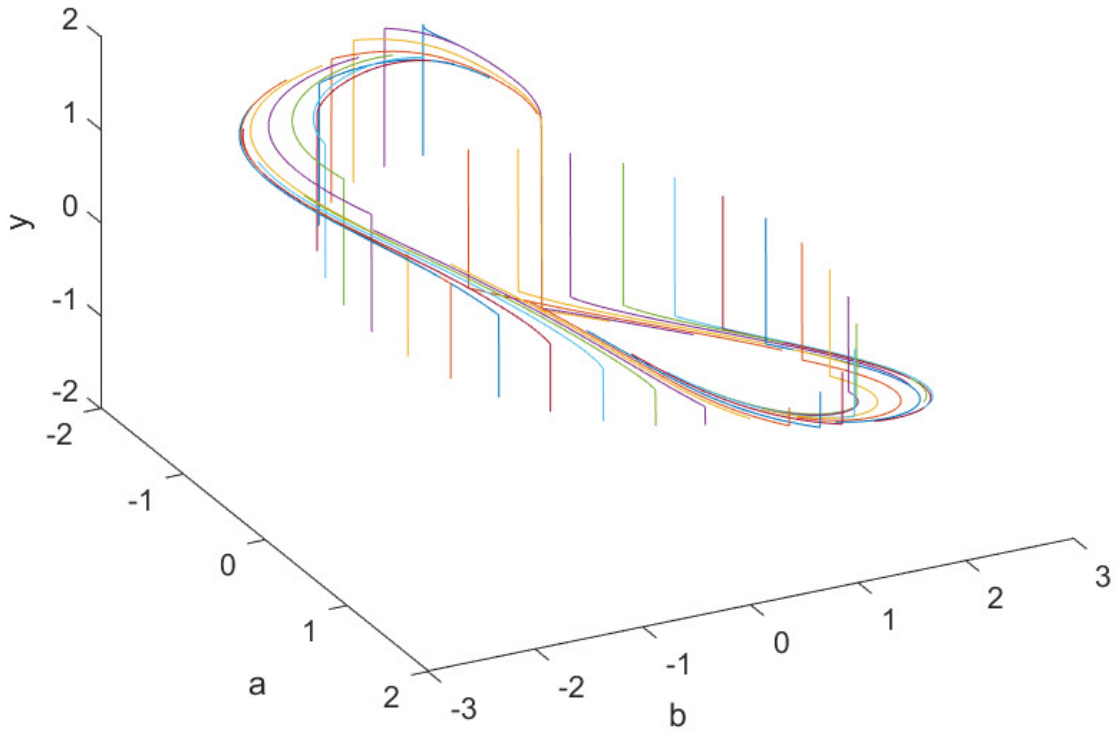


Figure 2.7: Solution plot for the CUSP model defined by 2.12, spatially discretized using 32 points.

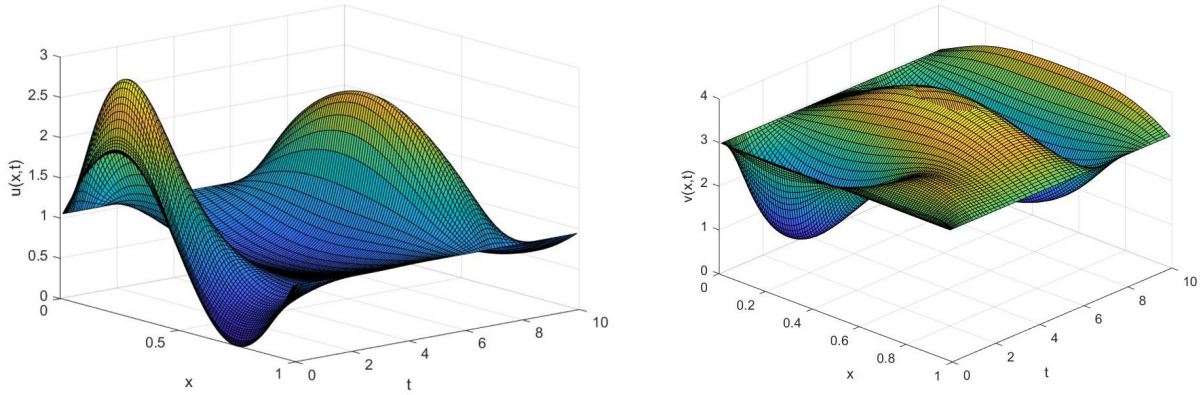


Figure 2.8: Solution plot for the one-dimensional Brusselator model defined by 2.13, using 200 discretized points.

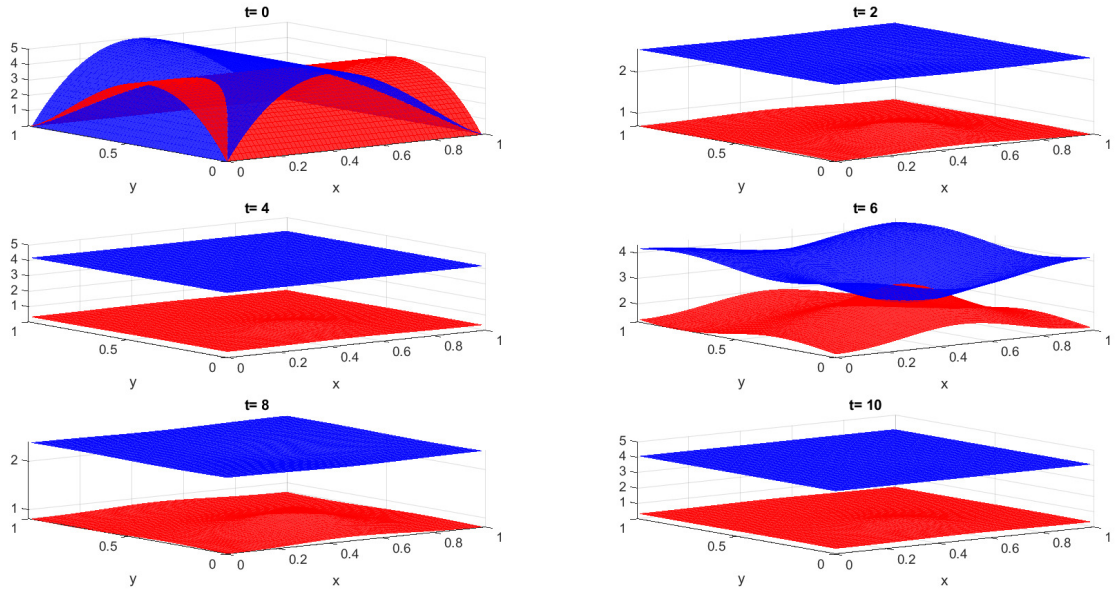


Figure 2.9: Solution plot for the two-dimensional Brusselator model defined by 2.13, using 60×60 discretized points and $\alpha = 0.01$. The function $u(x, y, t)$ is given by the red surface and the function $v(x, y, t)$ is given by the blue surface

The solution plots for the 2D heat transfer model defined by (2.14) is presented Figure 2.10. It can be noted that heat moves along the pipe in the shape of the velocity profile as anticipated. The plots match the results in [49], and they were verified by ODE15S Matlab function.

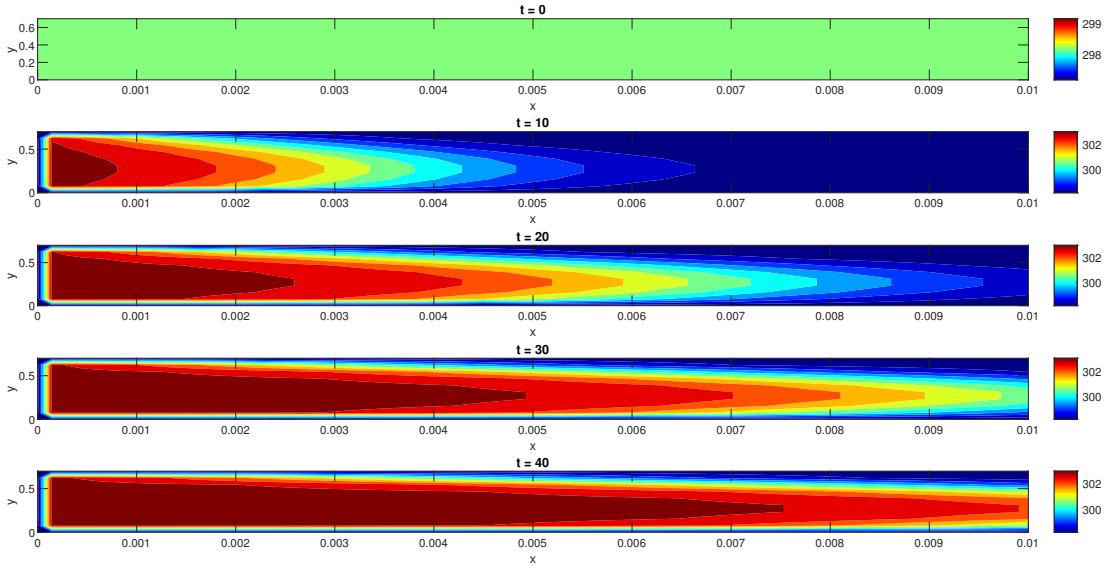


Figure 2.10: Solution plot for the two-dimensional heat model defined by 2.14, using $70x10$ discretized points.

2.6 Performance results and discussion of numerical experiments

This section shows the performance results obtained when the implementations of the IMEX-VSSBDF methods are applied to the seven problems defined in section 2.4. The results are given in terms of work-precision diagrams. In the physics-based splitting (PS) experiments, we treat the diffusion term implicitly and other terms explicitly. The Jacobian-based splitting (JS) is performed according to (2.3). In the numerical experiments, the runtimes reported are the minimum of three runs, and the accuracy data reported are the L_2-norms of the errors computed by comparing the numerical solutions against the reference solutions at the end of the time interval. The reference solutions have been generated using the fourth-order explicit Runge–Kutta Method (RK4) with a constant stepsize of 1e$-$8. The retirements are performed with tolerances range from 10^{-2} to 10^{-7}.

1D advection-diffusion Model

The one-dimensional advection-diffusion model described in Section 2.4 is evaluated for varying levels of diffusivity and strength of the advection term, using $N = 100$ over the time interval $[0, 1]$. In these experiments, the parameters of Algorithm 1 are chosen to be: $\eta_{max} = 5$ and $\eta_{min} = 0.2$. The work-precision plots are shown in Figure 2.11. It can be seen that the Jacobian-based splitting generally have better performance comparing with the physics-based splitting at a given order, especially when the strength of the advection and diffusion is increased.

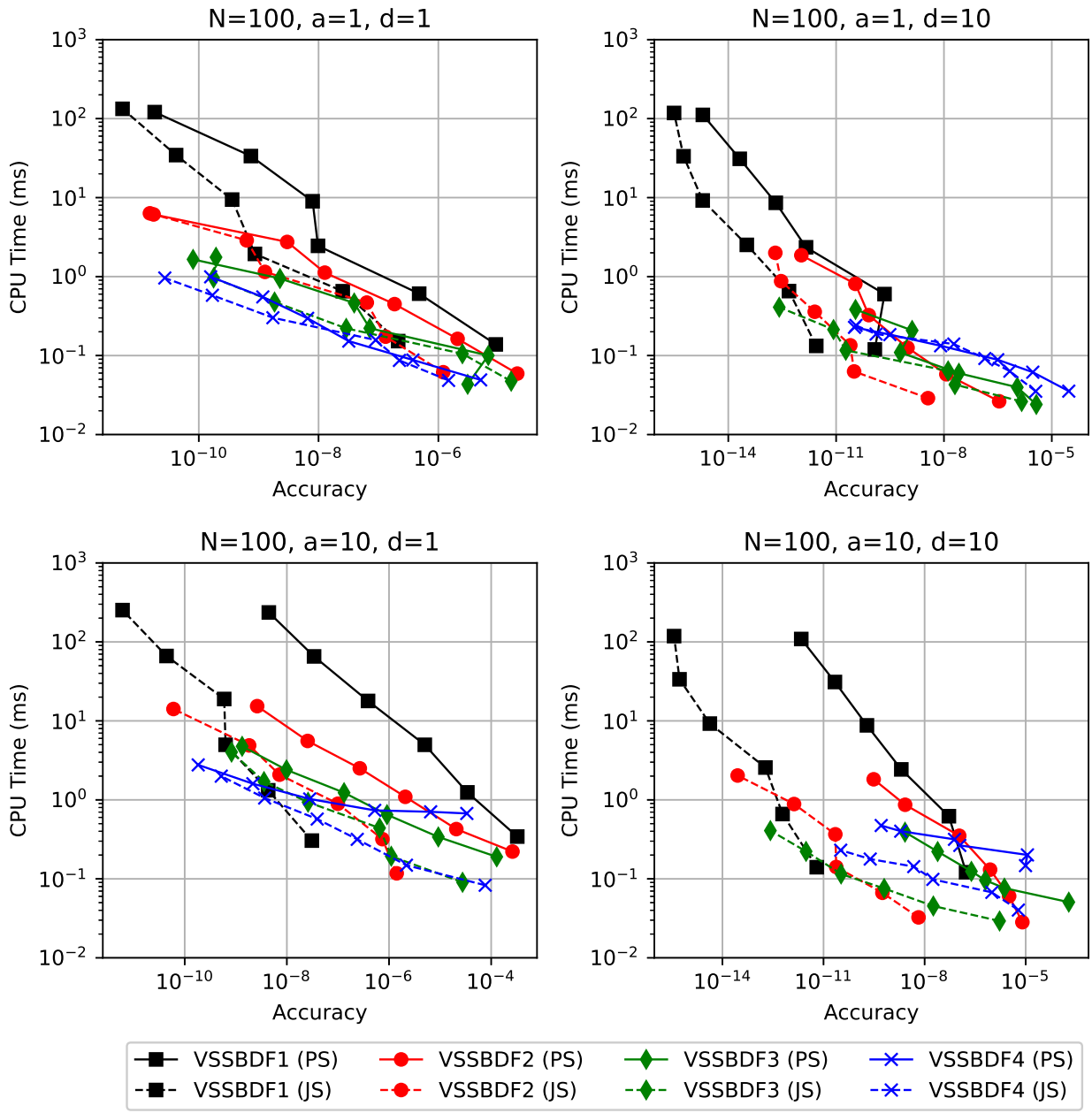


Figure 2.11: CPU time versus accuracy of IMEX-VSSBDF methods applied to 1D advection-diffusion model.

1D Advection-Diffusion-Reaction Model

The one-dimensional advection-diffusion-reaction model described in Section 2.4 is evaluated for different levels of diffusivity and strength of the advection term. The experiments are run over the time interval $[0, 1]$ for $N = 100$. The work-precision plots are shown in Figure 2.12. It can be seen that the Jacobian-based splitting for the VSSBDF4 method generally outperforms all other methods. In addition, the VSSBDF3 solver with the Jacobian-based

splitting generally provides better performance than the rest of solvers, especially for the case of $a = 100$ and $d = 10$.

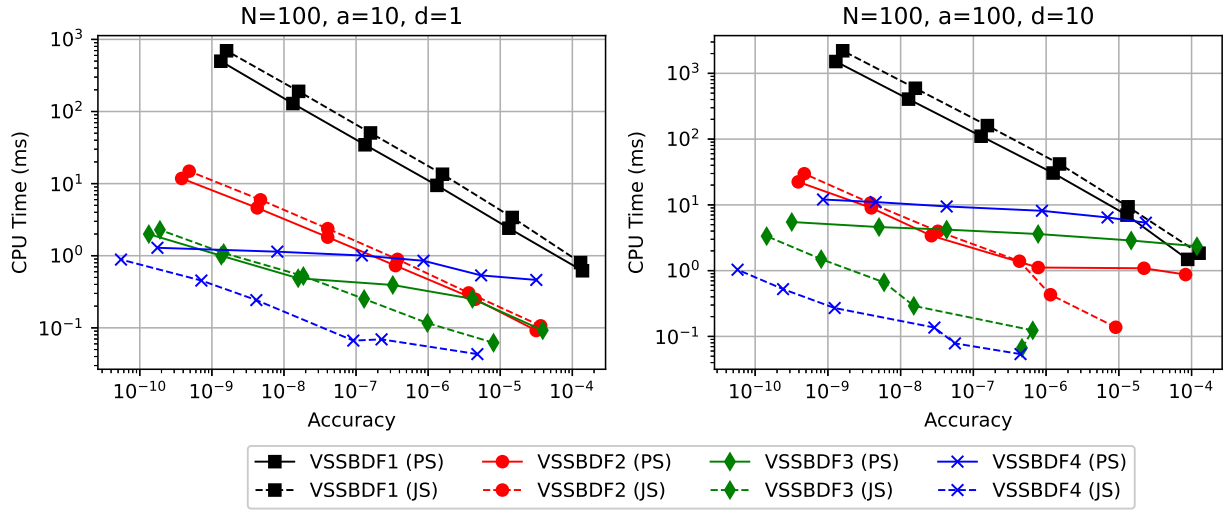


Figure 2.12: CPU time versus accuracy of IMEX-VSSBDF methods applied to 1D advection-diffusion-reaction model.

Combustion Model

The combustion model was simulated for the three types of reaction models, the FKPP, ignition, and Fisher models, for a grid size $N = 500$ and $U_0 \in \{0, 0.75, 0.99\}$. The work-precision plots are given in Figure 2.13. It can be noted that when the combustion model is tested with the FKPP or Fisher reactions, the benefits of the Jacobian splitting are more evident than when the ignition reaction is used. For example, the VSSBDF3 solver using the Jacobian splitting outperforms its counterpart methods using the physics-based splitting, and the VSSBDF2 and VBSDF4 solvers with the Jacobian splitting perform comparably with physics-based splitting for the counterpart methods.

CUSP Model

The CUSP model defined in section 2.4 is run with $N = 32$ and the final simulation time $t_f = 1.1$. In this experiment, the parameter of Algorithm 1 are chosen to be: $\eta_{max} = 5$ and $\eta_{min} = 0.2$. The work-precision plots are presented in Figure 2.14.

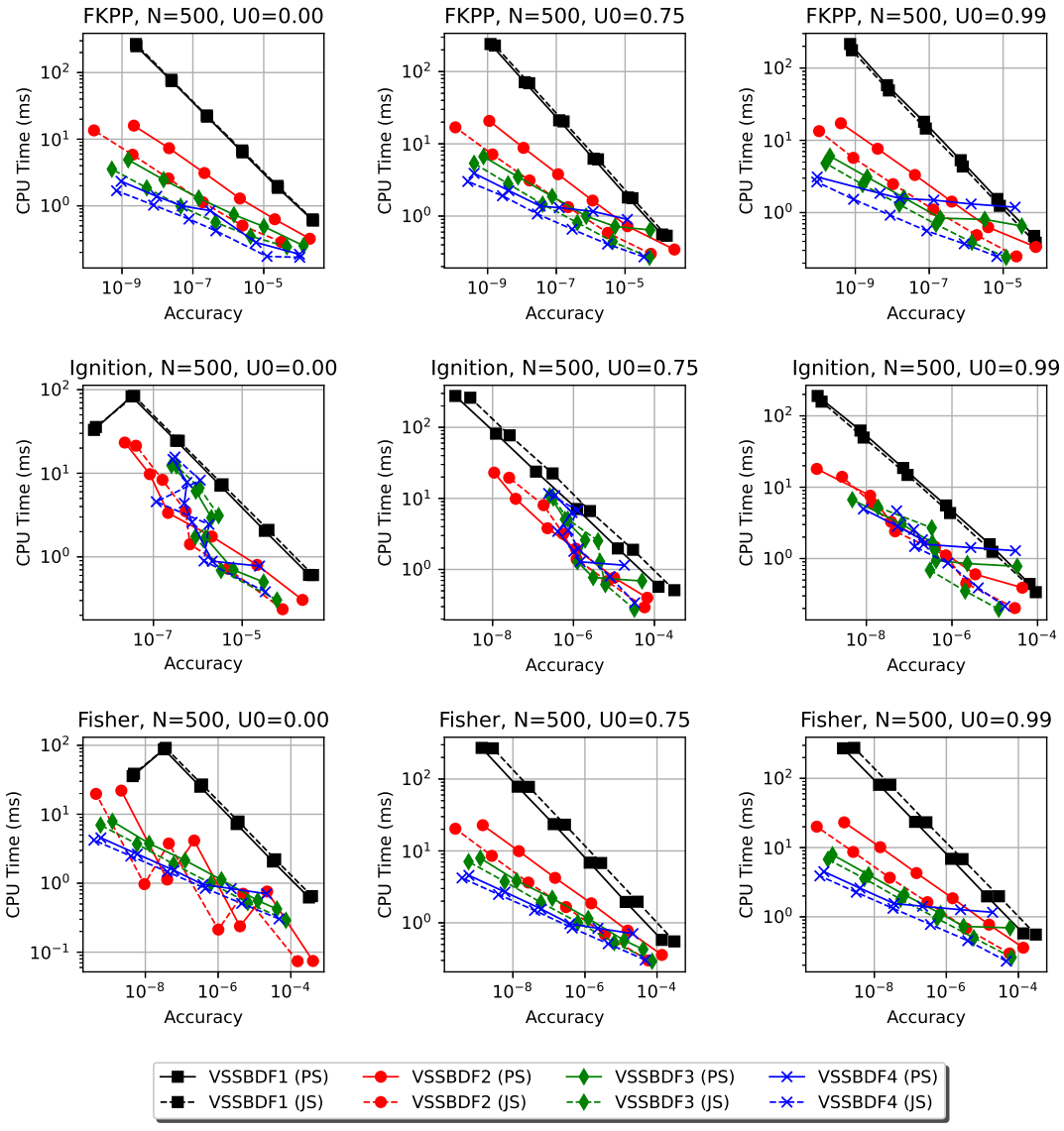


Figure 2.13: CPU time versus accuracy of IMEX-VSSBDF methods applied to the set of one-dimensional combustion problems.

The results show that VSSBDF methods using Jacobian splitting take less CPU time for a given tolerance than physics-based splitting and are thus more efficient.

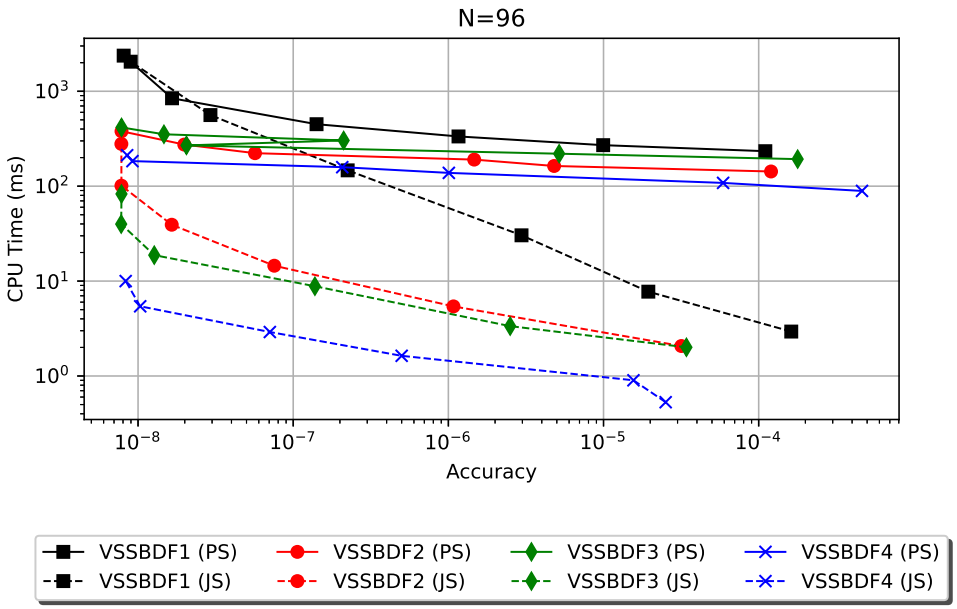


Figure 2.14: CPU time versus accuracy of IMEX-VSSBDF methods applied to CUSP model.

1D Brusselator Model

The additive splitting methods were tested over the one-dimensional Brusselator model presented in section 2.4. The one-dimensional model is run with $N \in \{200, 400\}$ and the final simulation time $t_f = 5$. In these experiments, the variable time step is calculated for $range = tol/3$, $\eta_{\max} = 5$, and $\eta_{\min} = 0.2$. The work-precision plots are shown in Figure 2.15. The results show that there is no noticeable difference between the type of splitting for the second- and third-order methods. The Jacobian splitting for first- and fourth-order methods always takes more CPU time than physics-based splitting to obtain a similar accuracy. The performance improvement when using physics-based splitting is particularly noticeable in the case of first-order methods for the case $\alpha = 1/50$.

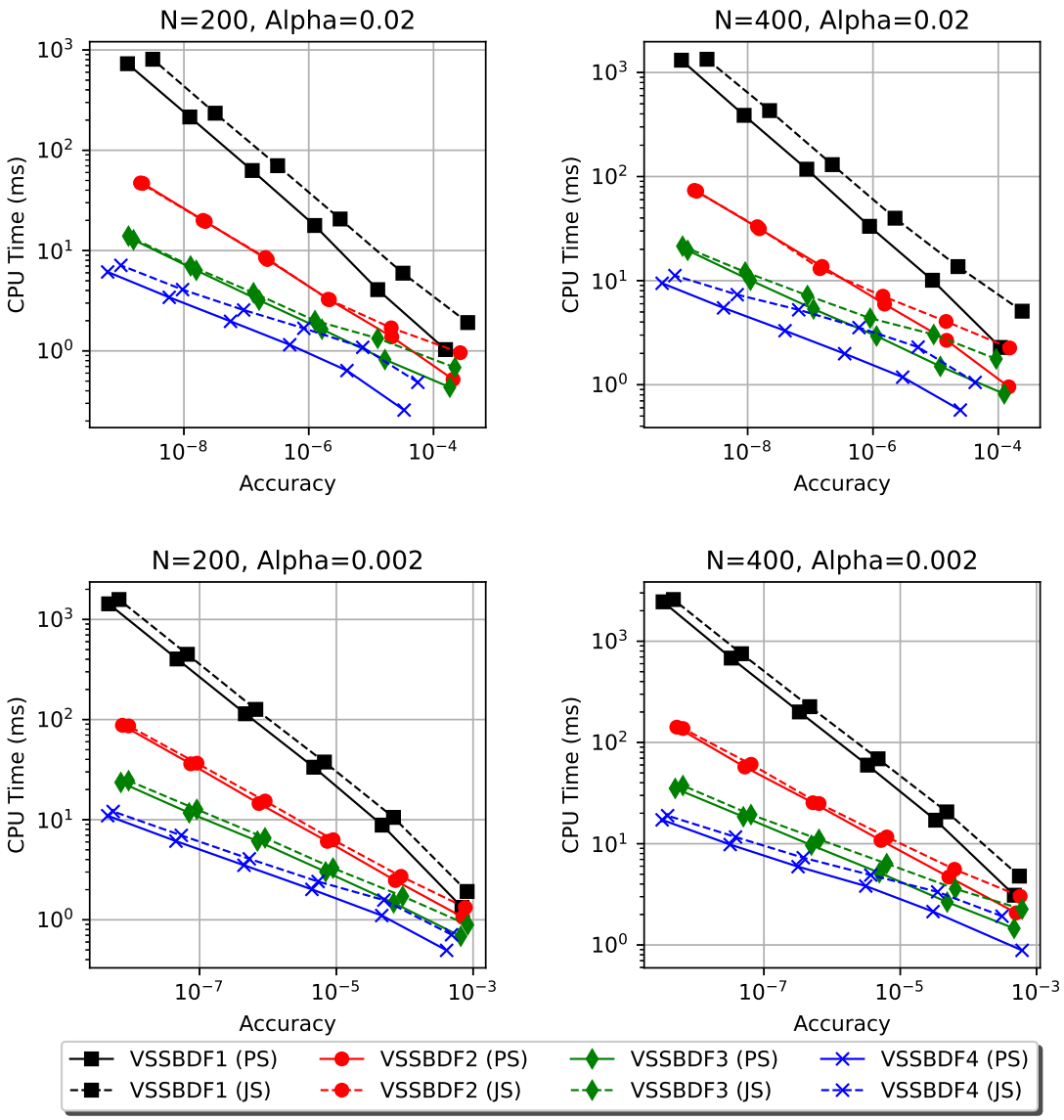


Figure 2.15: CPU time versus accuracy of IMEX-VSSBDF methods applied to one-dimensional Brusselator model.

2D Brusselator Model

The two-dimensional Brusselator model was evaluated using grid sizes of $N = 10$ and $N = 20$ and a final time of $t_f = 1$. The results of these simulations are depicted in work-precision plots in Figure 2.16. Analysis of these results reveals that for the methods of third and fourth order, the choice of splitting technique —whether physics-based or Jacobian— does not markedly impact the performance. Interestingly, for first-order methods, simulations

employing physical splitting techniques outperform those using Jacobian-based splitting.

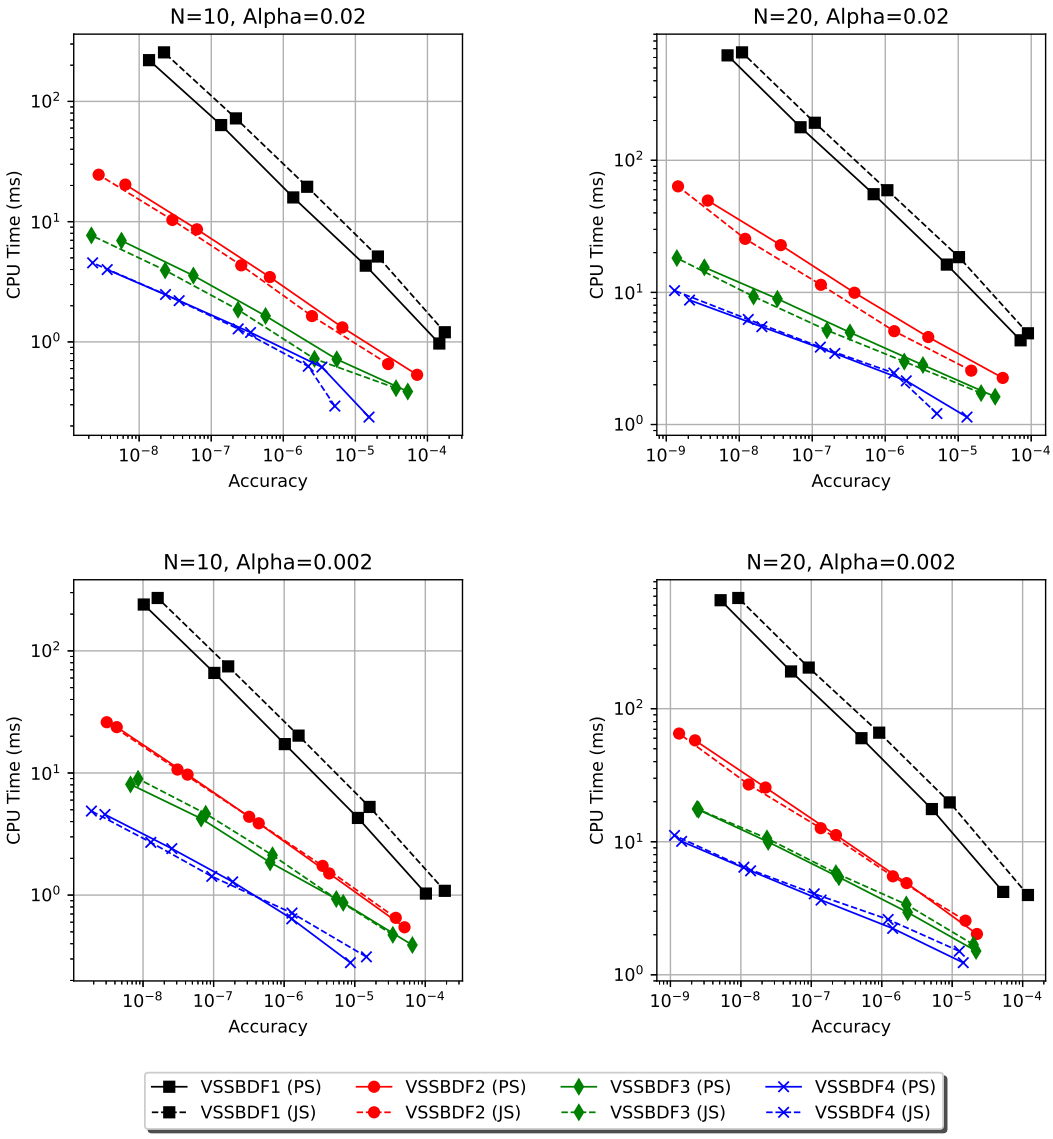


Figure 2.16: CPU time versus accuracy of IMEX-VSSBDF methods applied to two-dimensional Brusselator model.

2D Heat model

The two-dimensional heat transfer model is evaluated on three sets of grids: 70 ×10, 140 × 20, and 210 × 30, and with the final time $t_{final} = 10$. In this experiment, the variable time step is calculated provided $range = tol/3$, $\eta_{\max} = 5$, and $\eta_{\min} = 0.2$. The work-precision plots for the three cases are presented in Figure 2.17. In general, the Jacobian

splitting for the second, third and fourth methods is generally more accurate than the physic-based splitting. Remarkably, for first-order methods experiments, the physics-based splitting approach outshine the one using Jacobian-based splitting in terms of accuracy and run time.

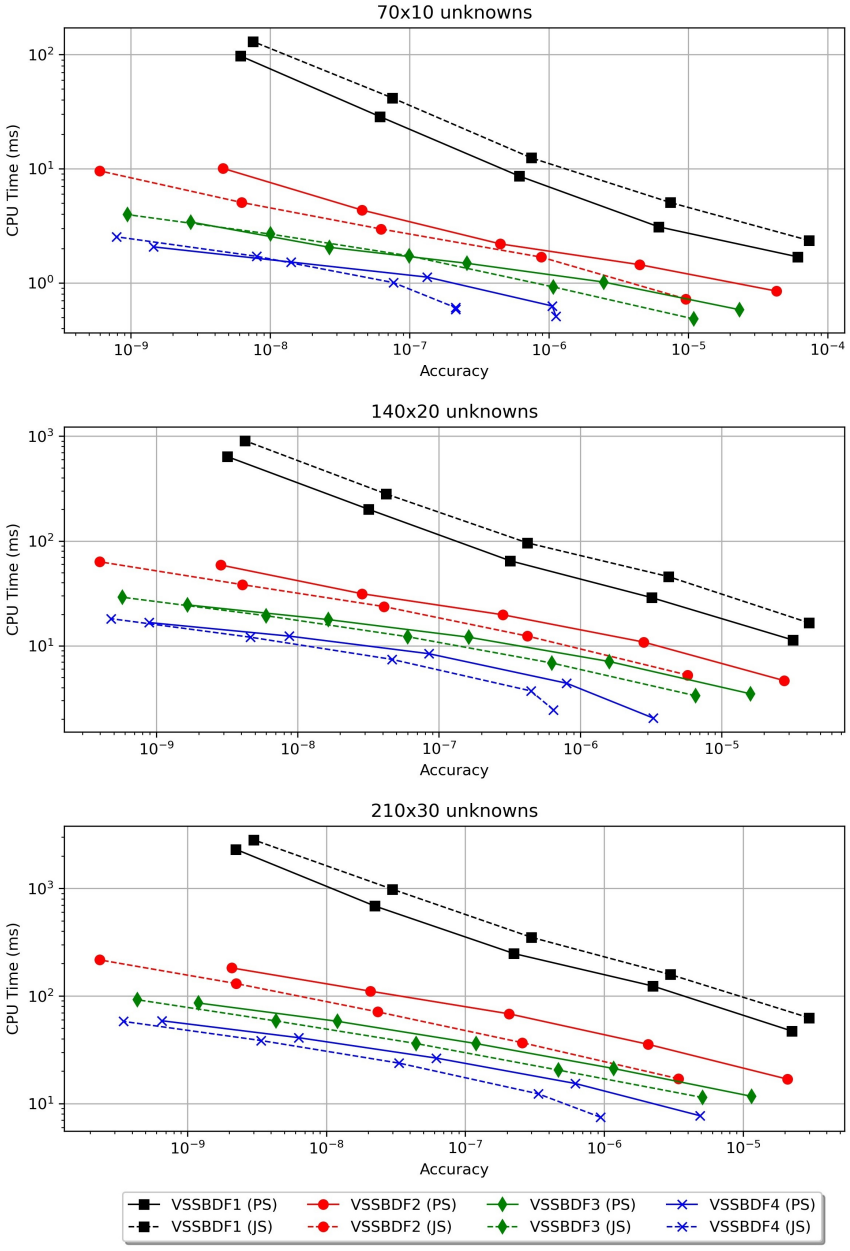


Figure 2.17: CPU time versus accuracy of IMEX-VSSBDF methods applied to two-dimensional heat model.

Chapter 3

k-step IIE linear multistep methods.

In this chapter, we introduce efficient 3-additive splitting method for solving ADR problems. These methods treat the diffusion and reaction terms implicitly, while handling the advection term explicitly. To align more closely with this approach, *we will adjust the abbreviation ADR model to a DRA* (diffusion-reaction-advection) in chapters 3 and 4. Additionally, we present a stability analysis and discuss the numerical performance of the derived methods.

3.1 k-step IIE linear multistep methods

In this section, we derive k-step IIE methods for the IVP (2.2), $k \geq 1$. Let $\mathbf{y}_n$ represent the approximate solution at $t_n = t_0 + n\Delta t$, where Δt is the discretization step size. Then, the IIE linear multistep method with k steps can be defined by

$$\frac{1}{\Delta t}\mathbf{y}_{n+1} + \frac{1}{\Delta t}\sum_{j=0}^{k-1} a_j \mathbf{y}_{n-j} = \sum_{j=-1}^{k-1} b_j^{[1]} \mathbf{f}^{[1]}(t_{n-j}, \mathbf{y}_{n-j}) + \sum_{j=-1}^{k-1} b_j^{[2]} \mathbf{f}^{[2]}(t_{n-j}, \mathbf{y}_{n-j}) + \sum_{j=0}^{k-1} b_j^{[3]} \mathbf{f}^{[3]}(t_{n-j}, \mathbf{y}_{n-j}), \quad (3.1)$$

where we assume that $b_{-1}^{[1]}, b_{-1}^{[2]} \neq 0$. Expanding (3.1) in Taylor series about t_n to obtain the truncation error yields

$$\begin{aligned}
&\frac{1}{\Delta t}\left[1+\sum_{j=0}^{k-1} a_j\right]\mathbf{y}(t_n)+\left[1-\sum_{j=1}^{k-1} j a_j\right]\mathbf{y}^{(1)}(t_n)+\cdots+\frac{(\Delta t)^{p-1}}{(p)!}\left[1+\sum_{j=1}^{k-1}(-j)^p a_j\right]\mathbf{y}^{(p)}(t_n)= \\
&\sum_{j=-1}^{k-1} b_j^{[1]}\mathbf{f}^{[1]}(\mathbf{y}(t_n))+\Delta t\left[b_{-1}^{[1]}-\sum_{j=1}^{k-1} j b_j^{[1]}\right]\left.\frac{\mathrm{d}\mathbf{f}^{[1]}}{\mathrm{d}t}\right|_{t=t_n}+\cdots+\frac{(\Delta t)^{p-1}}{(p-1)!}\left[b_{-1}^{[1]}+\sum_{j=1}^{k-1}(-j)^{p-1} b_j^{[1]}\right]\left.\frac{\mathrm{d}^{p-1}\mathbf{f}^{[1]}}{\mathrm{d}t^{p-1}}\right|_{t=t_n}+ \\
&\sum_{j=-1}^{k-1} b_j^{[2]}\mathbf{f}^{[2]}(\mathbf{y}(t_n))+\Delta t\left[b_{-1}^{[2]}-\sum_{j=1}^{k-1} j b_j^{[2]}\right]\left.\frac{\mathrm{d}\mathbf{f}^{[2]}}{\mathrm{d}t}\right|_{t=t_n}+\cdots+\frac{(\Delta t)^{p-1}}{(p-1)!}\left[b_{-1}^{[2]}+\sum_{j=1}^{k-1}(-j)^{p-1} b_j^{[2]}\right]\left.\frac{\mathrm{d}^{p-1}\mathbf{f}^{[2]}}{\mathrm{d}t^{p-1}}\right|_{t=t_n}+ \\
&\sum_{j=0}^{k-1} b_j^{[3]}\mathbf{f}^{[3]}(\mathbf{y}(t_n))-\Delta t\sum_{j=1}^{k-1} j b_j^{[3]}\left.\frac{\mathrm{d}\mathbf{f}^{[3]}}{\mathrm{d}t}\right|_{t=t_n}+\cdots+\frac{(\Delta t)^{p-1}}{(p-1)!}\sum_{j=1}^{k-1}(-j)^{p-1} b_j^{[3]}\left.\frac{\mathrm{d}^{p-1}\mathbf{f}^{[3]}}{\mathrm{d}t^{p-1}}\right|_{t=t_n}+\mathcal{O}((\Delta t)^p)
\end{aligned} \tag{3.2}$$

Using (2.2) in the truncation error (3.2) leads to the order conditions

$$\begin{aligned}
1+\sum_{j=0}^{k-1} a_j &= 0, \\
1-\sum_{j=1}^{k-1} j a_j &= \sum_{j=-1}^{k-1} b_j^{[1]} = \sum_{j=-1}^{k-1} b_j^{[2]} = \sum_{j=0}^{k-1} b_j^{[3]}, \\
\frac{1}{2}+\sum_{j=1}^{k-1}\frac{j^2}{2} a_j &= b_{-1}^{[1]}-\sum_{j=1}^{k-1} j b_j^{[1]} = b_{-1}^{[2]}-\sum_{j=1}^{k-1} j b_j^{[2]} = -\sum_{j=1}^{k-1} j b_j^{[3]}, \\
&\vdots \\
\frac{1}{p!}+\sum_{j=1}^{k-1}\frac{(-j)^p}{p!} a_j &= \frac{b_{-1}^{[1]}}{(p-1)!}+\sum_{j=1}^{k-1}\frac{(-j)^{p-1} b_j^{[1]}}{(p-1)!} = \frac{b_{-1}^{[2]}}{(p-1)!}+\sum_{j=1}^{k-1}\frac{(-j)^{p-1} b_j^{[2]}}{(p-1)!} = \sum_{j=1}^{k-1}\frac{(-j)^{p-1} b_j^{[3]}}{(p-1)!}.
\end{aligned} \tag{3.3}$$

The system (3.3) gives the conditions for a k-step IIE-LMM to have order p. The following theorem investigates the maximal order for a given k.

Theorem 1. *For the k-step IIE-LMM (3.1), we have:*

I. If $p \leq k$, then the $3p+1$ conditions of the system (3.3) are linearly independent. So there exist k-step IIE-LMMs of order k.

II. The order of accuracy of a k-step IIE-LMM is at most k.

III. The family of k-step IIE-LMMs of order k has $k+1$ free parameters.

Proof. I. Because linear independence for $p = k$ implies linear independence for $p \leq k$,

it is sufficient to prove the case $p = k$ only. The system (3.3) can be represented in matrix form

$$\mathbf{WV} = \mathbf{U}, \tag{3.4}$$

where

$$\mathbf{V} = [a_0, \ldots, a_{k-1}, b_{-1}^{[1]}, b_0^{[1]}, \ldots, b_{k-1}^{[1]}, b_{-1}^{[2]}, b_0^{[2]}, \ldots, b_{k-1}^{[2]}, b_0^{[3]}, b_1^{[3]}, \ldots, b_{k-1}^{[3]}]^T \in \mathbb{R}^{4k+2},$$

$$\mathbf{U} = [-1, \ldots, -1_{k+1}, 0, \ldots, 0_{2k}]^T \in \mathbb{R}^{3k+1},$$

and

$$\mathbf{W} = \left[\begin{array}{c|c|cccc} \mathbf{A} & \mathbf{L} & 0 & \ldots & \ldots & 0 \\ \mathbf{H} & & -\mathbf{DA} & \mathbf{0} & \ldots & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{A} & -\mathbf{1} & -\mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{A} & -\mathbf{A} \end{array}\right]_{(3k+1)\times(4k+2)},$$

with

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 0 & -1 & -2 & \ldots & 1-k \\ 0 & 1 & 4 & \ldots & (1-k)^2 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & (-1)^{k-1} & (-2)^{k-1} & \ldots & (1-k)^{k-1} \end{bmatrix}_{k\times k}, \tag{3.5}$$

$$\mathbf{D} = \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & k \end{bmatrix}_{k\times k}, \tag{3.6}$$

$$\mathbf{H} = \Big[0, (-1)^k, \ldots, (1-k)^k\Big]_{1\times k}, \tag{3.7}$$

$$\mathbf{L} = \Big[0, -1, -2, \ldots, -k\Big]^T_{1\times(k+1)}, \tag{3.8}$$

and $\mathbf{1}$ is a vector of ones with size k. The matrix $\mathbf{A}$ is a Vandermonde matrix for the distinct numbers $\{0, -1, -2, \ldots, 1-k\}$. Because the Vandermonde matrix $\mathbf{A}$ is nonsingular [27], it can be noted that columns $1, k+2, \ldots, 2k+1, 2k+3, \ldots, 4k+2$,

of the matrix $\mathbf{W}$ are linearly independent. Thus, all $(3p+1)$ conditions are linearly independent, and the system (3.3) admits a $(4k-3p+1)$-parameter family of solutions for $p \leq k$. Because columns $(k+1)$ and $(2k+2)$ of matrix $\mathbf{W}$ do not enter into the above proof, we know $b_{-1}^{[1]}$ and $b_{-1}^{[2]}$ can have any nonzero value. This property verifies that the family of methods is IIE-LMM.

II. Assume the IIE-LMM has accuracy $\mathcal{O}((\Delta t)^{k+r})$, $r \geq 1$; then the following conditions should be satisfied:

$$\begin{aligned} \sum_{j=-1}^{k-1} b_j^{[2]} &= \sum_{j=0}^{k-1} b_j^{[3]}, \\ b_{-1}^{[2]} + \sum_{j=1}^{k-1} (-j) b_j^{[2]} &= \sum_{j=1}^{k-1} (-j) b_j^{[3]}, \\ &\vdots \\ b_{-1}^{[2]} + \sum_{j=1}^{k-1} (-j)^k b_j^{[2]} &= \sum_{j=1}^{k-1} (-j)^k b_j^{[3]}. \end{aligned} \tag{3.9}$$

Let $\mu_j = b_j^{[2]} - b_j^{[3]}$; then the system (3.9) can be written as

$$\begin{aligned} b_{-1}^{[2]} + \sum_{j=0}^{k-1} \mu_j &= 0, \\ b_{-1}^{[2]} + \sum_{j=1}^{k-1} (-j) \mu_j &= 0, \\ &\vdots \\ b_{-1}^{[2]} + \sum_{j=1}^{k-1} (-j)^k \mu_j &= 0. \end{aligned} \tag{3.10}$$

Writing the system (3.10) in a matrix form yields

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & -1 & \dots & 1-k \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 0 & (-1)^k & \dots & (1-k)^k \end{bmatrix}_{(k+1)\times(k+1)} \begin{bmatrix} b_{-1}^{[2]} \\ \mu_0 \\ \vdots \\ \mu_{k-1} \end{bmatrix}_{(k+1)\times 1} = \mathbf{0},$$

implying that $b_{-1}^{[2]} = 0$ and $b_j^{[2]} = b_j^{[3]}, j = 1, 2, \dots, k-1$, because the matrix of coefficients is nonsingular. This contradicts the assumption $b_{-1}^{[2]} \neq 0$.

III. Consider a k-step IIE-LMM that achieves the maximal order. Then, as shown above, the family of such method has exactly $(k+1)$ free parameters.

□

3.2 Linear stability analysis for IIE-LMMs

Consider the scalar test problem

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -\mu^2 y + \lambda y + i\nu y, \tag{3.11}$$

where $\mu, \nu \in \mathbb{R}$, $\lambda = \lambda_r + i\lambda_i \in \mathbb{C}$, and $\lambda_r, \lambda_i \in \mathbb{R}$. The terms in the test problem are meant to represent typical eigenvalues of the Jacobians of $\mathbf{f}^{[i]}$, $i = 1, 2, 3$, with respect to the solution $\mathbf{y}$, assuming they are simultaneously diagonalizable. Specifically, $-\mu^2 y$ corresponds to the diffusion term, λy corresponds to the reaction term, and $i\nu y$ corresponds to the advection term.

Applying IIE-LMM (3.1) to (3.11) yields

$$\begin{aligned}\left[1 + \Delta t b_{-1}^{[1]}\mu^2 - \Delta t b_{-1}^{[2]}(\lambda_r + i\lambda_i)\right] y_{n+1} \\ + \sum_{j=0}^{k-1} \left[a_j - i\Delta t(b_j^{[2]}\lambda_i + b_j^{[3]}\nu) + \Delta t\mu^2 b_j^{[1]} - \Delta t b_j^{[2]}\lambda_r\right] y_{n-j} = 0.\end{aligned}$$

Assuming this linear constant-coefficient difference equation has solutions of the form $y_n = \xi^n$, then the characteristic equation is given by

$$\begin{aligned}\Phi(\xi) \equiv \left[1 + \Delta t(b_{-1}^{[1]}\mu^2 - b_{-1}^{[2]}\lambda_r) - i\Delta t b_{-1}^{[2]}\lambda_i\right] \xi^k \\ + \sum_{j=0}^{k-1} \left[a_j - i\Delta t(b_j^{[2]}\lambda_i + b_j^{[3]}\nu) + \Delta t(\mu^2 b_j^{[1]} - b_j^{[2]}\lambda_r)\right] \xi^{k-j-1}.\end{aligned} \tag{3.12}$$

It can be noted that the stability holds if and only if all simple roots of the equation (3.12) satisfy $|\xi_j| \leq 1$ and multiple roots satisfy $|\xi_j| < 1$. The region in complex plane that contains these roots is called the stability region of the method. However, the stability region of 3-additive methods is in $\mathbb{C}^2$ ($\simeq \mathbb{R}^4$). It is not entirely obvious how to characterize the stability behavior of a 3-additive LMM by adjusting the parameters μ, ν, and λ. To overcome

this difficulty, we discuss the stability of IIE-LMMs in three cases $\lambda = i\nu$, $\lambda = -\mu^2$ and $\lambda = -\mu^2 + i\nu$. These cases represent the limits of the reaction term having purely imaginary eigenvalues, purely negative real eigenvalues, and the intermediate case where it has an equal combination of the two.

3.3 Construction and Stability of IIE-LMMs

In this section, we construct IIE-LMMs by using the coefficients of well-known IMEX methods for the $\mathbf{f}^{[1]}$ (diffusion) and $\mathbf{f}^{[2]}$ (reaction) terms, and then derive coefficients of $\mathbf{f}^{[3]}$ (advection) terms that satisfy the relevant order conditions (3.3). The IIE-LMMs up to fourth order are presented in table 3.1. The order conditions for IIE-LMMs methods are summarized in table 3.2. We then also analyze the linear stability of the constructed methods using eq. (3.11) in the three cases described. The linear stability regions for IIE-LMMs up to fourth order are presented in table 3.3.

k	Methods	
$k=1$	$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t[\alpha\mathbf{f}^{[1]}(t_{n+1},\mathbf{y}_{n+1}) + (1-\alpha)\mathbf{f}^{[1]}(t_n,\mathbf{y}_n) + \beta\mathbf{f}^{[2]}(t_{n+1},\mathbf{y}_{n+1}) + (1-\beta)\mathbf{f}^{[2]}(t_n,\mathbf{y}_n) + \mathbf{f}^{[3]}(t_n,\mathbf{y}_n)].$	IIE1
$k=2$	$\mathbf{y}_{n+1} + a_0\mathbf{y}_n + a_1\mathbf{y}_{n-1} = \Delta t\left(b_{-1}^{[1]}\mathbf{f}^{[1]}(t_{n+1},\mathbf{y}_{n+1}) + b_0^{[1]}\mathbf{f}^{[1]}(t_n,\mathbf{y}_n) + b_1^{[1]}\mathbf{f}^{[1]}(t_{n-1},\mathbf{y}_{n-1})\right) + \Delta t\left(b_{-1}^{[2]}\mathbf{f}^{[2]}(t_{n+1},\mathbf{y}_{n+1}) + b_0^{[2]}\mathbf{f}^{[2]}(t_n,\mathbf{y}_n) + b_1^{[2]}\mathbf{f}^{[2]}(t_{n-1},\mathbf{y}_{n-1})\right) + \Delta t\left(b_0^{[3]}\mathbf{f}^{[3]}(t_n,\mathbf{y}_n) + b_1^{[3]}\mathbf{f}^{[3]}(t_{n-1},\mathbf{y}_{n-1})\right).$	IIE2
$k=3$	$\mathbf{y}_{n+1} + a_0\mathbf{y}_n + a_1\mathbf{y}_{n-1} + a_2\mathbf{y}_{n-2} = \Delta t\left(b_{-1}^{[1]}\mathbf{f}^{[1]}(t_{n+1},\mathbf{y}_{n+1}) + b_0^{[1]}\mathbf{f}^{[1]}(t_n,\mathbf{y}_n) + b_1^{[1]}\mathbf{f}^{[1]}(t_{n-1},\mathbf{y}_{n-1}) + b_2^{[1]}\mathbf{f}^{[1]}(t_{n-2},\mathbf{y}_{n-2})\right) + \Delta t\left(b_{-1}^{[2]}\mathbf{f}^{[2]}(t_{n+1},\mathbf{y}_{n+1}) + b_0^{[2]}\mathbf{f}^{[2]}(t_n,\mathbf{y}_n) + b_1^{[2]}\mathbf{f}^{[2]}(t_{n-1},\mathbf{y}_{n-1}) + b_2^{[2]}\mathbf{f}^{[2]}(t_{n-2},\mathbf{y}_{n-2})\right) + \Delta t\left(b_0^{[3]}\mathbf{f}^{[3]}(t_n,\mathbf{y}_n) + b_1^{[3]}\mathbf{f}^{[3]}(t_{n-1},\mathbf{y}_{n-1}) + b_2^{[3]}\mathbf{f}^{[3]}(t_{n-2},\mathbf{y}_{n-2})\right).$	IIE3
$k=4$	$\mathbf{y}_{n+1} + a_0\mathbf{y}_n + a_1\mathbf{y}_{n-1} + a_2\mathbf{y}_{n-2} + a_3\mathbf{y}_{n-3} = \Delta t\left(b_{-1}^{[1]}\mathbf{f}^{[1]}(t_{n+1},\mathbf{y}_{n+1}) + b_0^{[1]}\mathbf{f}^{[1]}(t_n,\mathbf{y}_n) + b_1^{[1]}\mathbf{f}^{[1]}(t_{n-1},\mathbf{y}_{n-1}) + b_2^{[1]}\mathbf{f}^{[1]}(t_{n-2},\mathbf{y}_{n-2}) + b_3^{[1]}\mathbf{f}^{[1]}(t_{n-3},\mathbf{y}_{n-3})\right) + \Delta t\left(b_{-1}^{[2]}\mathbf{f}^{[2]}(t_{n+1},\mathbf{y}_{n+1}) + b_0^{[2]}\mathbf{f}^{[2]}(t_n,\mathbf{y}_n) + b_1^{[2]}\mathbf{f}^{[2]}(t_{n-1},\mathbf{y}_{n-1}) + b_2^{[2]}\mathbf{f}^{[2]}(t_{n-2},\mathbf{y}_{n-2}) + b_3^{[2]}\mathbf{f}^{[2]}(t_{n-3},\mathbf{y}_{n-3})\right) + \Delta t\left(b_0^{[3]}\mathbf{f}^{[3]}(t_n,\mathbf{y}_n) + b_1^{[3]}\mathbf{f}^{[3]}(t_{n-1},\mathbf{y}_{n-1}) + b_2^{[3]}\mathbf{f}^{[3]}(t_{n-2},\mathbf{y}_{n-2}) + b_3^{[3]}\mathbf{f}^{[3]}(t_{n-3},\mathbf{y}_{n-3})\right).$	IIE4

Table 3.1: IIE-LMMs

k	Order Conditions	
$k=1$	$1+a_0 = 0,$ $1 = b_{-1}^{[1]} + b_0^{[1]},$ $1 = b_{-1}^{[2]} + b_0^{[2]},$ $1 = b_0^{[3]}.$	IIE1 order conditions
$k=2$	$1+a_0+a_1 = 0,$ $1-a_1 = b_{-1}^{[1]} + b_0^{[1]} + b_1^{[1]},$ $1-a_1 = b_{-1}^{[2]} + b_0^{[2]} + b_1^{[2]},$ $1-a_1 = b_0^{[3]} + b_1^{[3]},$ $1+a_1 = 2b_{-1}^{[1]} - 2b_1^{[1]},$ $1+a_1 = 2b_{-1}^{[2]} - 2b_1^{[2]},$ $1+a_1 = -2b_1^{[3]}.$	IIE2 order conditions
$k=3$	$1+a_0+a_1+a_2 = 0,$ $1-a_1-2a_2 = b_{-1}^{[1]} + b_0^{[1]} + b_1^{[1]} + b_2^{[1]},$ $1-a_1-2a_2 = b_{-1}^{[2]} + b_0^{[2]} + b_1^{[2]} + b_2^{[2]},$ $1-a_1-2a_2 = b_0^{[3]} + b_1^{[3]} + b_2^{[3]},$ $1+a_1+4a_2 = 2b_{-1}^{[1]} - 2b_1^{[1]} - 4b_2^{[1]},$ $1+a_1+4a_2 = 2b_{-1}^{[2]} - 2b_1^{[2]} - 4b_2^{[2]},$ $1+a_1+4a_2 = -2b_1^{[3]} - 4b_2^{[3]},$ $1-a_1-8a_2 = 3b_{-1}^{[1]} + 3b_1^{[1]} + 12b_2^{[1]},$ $1-a_1-8a_2 = 3b_{-1}^{[2]} + 3b_1^{[2]} + 12b_2^{[2]},$ $1-a_1-8a_2 = 3b_1^{[3]} + 12b_2^{[3]}.$	IIE3 order conditions
$k=4$	$1+a_0+a_1+a_2+a_3 = 0,$ $1-a_1-2a_2-3a_3 = b_{-1}^{[1]} + b_0^{[1]} + b_1^{[1]} + b_2^{[1]} + b_3^{[1]},$ $1-a_1-2a_2-3a_3 = b_{-1}^{[2]} + b_0^{[2]} + b_1^{[2]} + b_2^{[2]} + b_3^{[2]},$ $1-a_1-2a_2-3a_3 = b_0^{[3]} + b_1^{[3]} + b_2^{[3]} + b_3^{[3]},$ $1+a_1+4a_2+9a_3 = 2b_{-1}^{[1]} - 2b_1^{[1]} - 4b_2^{[1]} - 6b_3^{[1]},$ $1+a_1+4a_2+9a_3 = 2b_{-1}^{[2]} - 2b_1^{[2]} - 4b_2^{[2]} - 6b_3^{[2]},$ $1+a_1+4a_2+9a_3 = -2b_1^{[3]} - 4b_2^{[3]} - 6b_3^{[3]},$ $1-a_1-8a_2-27a_3 = 3b_{-1}^{[1]} + 3b_1^{[1]} + 12b_2^{[1]} + 27b_3^{[1]},$ $1-a_1-8a_2-27a_3 = 3b_{-1}^{[2]} + 3b_1^{[2]} + 12b_2^{[2]} + 27b_3^{[2]},$ $1-a_1-8a_2-27a_3 = 3b_1^{[3]} + 12b_2^{[3]} + 27b_3^{[3]},$ $1+a_1+16a_2+81a_3 = 4b_{-1}^{[1]} - 4b_1^{[1]} - 32b_2^{[1]} - 108b_3^{[1]},$ $1+a_1+16a_2+81a_3 = 4b_{-1}^{[2]} - 4b_1^{[2]} - 32b_2^{[2]} - 108b_3^{[2]},$ $1+a_1+16a_2+81a_3 = -4b_1^{[3]} - 32b_2^{[3]} - 108b_3^{[3]}.$	IIE4 order conditions

Table 3.2: IIE-LMMs Order Conditions

First-order IIE methods

The two-parameter family of one-step, first-order IIE-LMMs for (2.2) is presented in table 3.1. The choice $(\alpha, \beta) = (1, \frac{1}{2})$ in IIE1 yields the formula (1.3), which was used to introduce 3-additive splitting methods in chapter 1.

Stability analysis

Applying the method IIE1 to the test equation (3.11) yields the stability region $\{|\xi(\nu, \lambda, \mu)| \leq 1\}$, where

$$\begin{aligned} \xi(\nu, \lambda, \mu) &= 1 + \tfrac{i\Delta t\nu - \Delta t\mu^2 + \Delta t\lambda}{1 + \Delta t(\alpha\mu^2 - \beta\lambda)}, \\ &= 1 + \tfrac{\Delta t(\lambda_r - \mu^2) + i\Delta t(\nu + \lambda_i)}{1 + \Delta t(\alpha\mu^2 - \beta\lambda_r) - i\Delta t\beta\lambda_i}. \end{aligned} \tag{3.13}$$

For simplicity, the stability region can be defined as a region of the parameters $(z_1, z_2, z_3) = (-\mu^2\Delta t, \nu\Delta t, \lambda\Delta t)$. The choice $(\alpha, \beta) = (\frac{1}{2}, \frac{3}{2})$ in IIE1 yields the IIE method

$$\begin{aligned} \mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \Bigg[\frac{1}{2} \left(\mathbf{f}^{[1]}(t_{n+1}, \mathbf{y}_{n+1}) + \mathbf{f}^{[1]}(t_n, \mathbf{y}_n) \right) + \frac{1}{2} \left(3\mathbf{f}^{[2]}(t_{n+1}, \mathbf{y}_{n+1}) + \mathbf{f}^{[2]}(t_n, \mathbf{y}_n) \right) \\ + \mathbf{f}^{[3]}(t_n, \mathbf{y}_n) \Bigg], \end{aligned} \tag{3.14}$$

which turns out to have favorable stability properties. We refer to this method as IIE-1. We discuss the following cases of stability for the method (3.14).

- Case 1 (diffusive reaction term $\lambda = -\mu^2$)

  The stability function (3.13) is given by

  $$\xi(z_1, z_2) = 1 + \frac{iz_2 + 2z_1}{1 - 2z_1}.$$

  The stability region for this case is presented in table 3.3, and it can be shown that the method is A_0-stable.

- Case 2 (non-diffusive reaction term $\lambda = i\nu$)

  The stability function (3.13) is given by

  $$\xi(z_1, z_2) = 1 + \frac{4iz_2 + 2z_1}{2 - z_1 - 3iz_2}.$$

The stability region for this case is presented in table 3.3, and it can be shown that the method is A-stable.

- Case 3: (mixed reaction term $\lambda = -\mu^2 + i\nu$)
  The stability function (3.13) is given by

$$\xi(z_1, z_2) = 1 + \frac{4iz_2 + 4z_1}{2 - 4z_1 - 3iz_2}.$$

The stability region for this case is presented in table 3.3, and it can be shown that the method is A-stable; table 3.3.

In addition, we compare the stability of the first-order IIE1 method (3.14) with the stability of IMEX1 method,

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{\Delta t}{4}\left(\mathbf{g}(t_{n+1}, \mathbf{y}_{n+1}) + 3\mathbf{g}(t_n, \mathbf{y}_n)\right) + \Delta t\mathbf{f}(t_n, \mathbf{y}_n), \tag{3.15}$$

and the semi-implicit BDF (SBDF1),

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t\left(\mathbf{g}(t_{n+1}, \mathbf{y}_{n+1}) + \mathbf{f}(t_n, \mathbf{y}_n)\right). \tag{3.16}$$

It can be noted from table 3.3 that the stability region for IIE1 method (3.14) is significantly larger than the one for SBDF1.

3.3.1 Two-step, second-order IIE-LMMs

Two-step IIE-LMMs for IVP (2.2) is presented in table 3.1. A second-order method is obtained provided the IIE2 order conditions in table 3.2 are satisfied.

Stability analysis for $k = p = 2$ IIE-LMMs

The second-degree characteristic polynomial resulting from (IIE2) applied to the test equation (3.11) is given by

$$\begin{aligned}\Phi(\xi) \equiv \left[1 + b_{-1}^{[1]}\Delta t\mu^2 - \Delta t b_{-1}^{[2]}(\lambda_r + i\lambda_i)\right]\xi^2 \\ + \left[a_0 - i\Delta t(b_0^{[2]}\lambda_i + b_0^{[3]}\nu) + \Delta t(b_0^{[1]}\mu^2 - b_0^{[2]}\lambda_r)\right]\xi \\ + a_1 - i\Delta t(b_1^{[2]}\lambda_i + b_1^{[3]}\nu) + \Delta t(b_1^{[1]}\mu^2 - b_1^{[2]}\lambda_r) = 0.\end{aligned}$$

IIE-CNLF Method

A two-step, second-order IIE-LMM can be obtained by applying something similar to the Crank–Nicolson leapfrog (CNLF) method for the implicit terms $f^{[1]}(t_{n+1}, \mathbf{y}_{n+1})$ and $f^{[2]}(t_{n+1}, \mathbf{y}_{n+1})$ and the leapfrog method for the explicit term $f^{[3]}(t_n, \mathbf{y}_n)$ as follows

$$\begin{aligned}\mathbf{y}_{n+1} &= \mathbf{y}_{n-1} + \Delta t \left(\mathbf{f}^{[1]}(t_{n+1}, \mathbf{y}_{n+1}) + \mathbf{f}^{[1]}(t_{n-1}, \mathbf{y}_{n-1})\right) \\ &\quad + 2\Delta t \left(\mathbf{f}^{[2]}(t_{n+1}, \mathbf{y}_{n+1}) - \mathbf{f}^{[2]}(t_n, \mathbf{y}_n) + \mathbf{f}^{[2]}(t_{n-1}, \mathbf{y}_{n-1})\right) + 2\Delta t \left(\mathbf{f}^{[3]}(t_n, \mathbf{y}_n)\right).\end{aligned} \tag{3.17}$$

We refer this method to as IIE-CNLF2.

Stability Analysis

We discuss the stability of the IIE-CNLF2 method (3.17) for three cases.

- Case 1 (diffusive reaction term $\lambda = -\mu^2$)

  The stability region for this case is presented in table 3.3, and it can be shown that the method is A_0-stable.
- Case 2 (non-diffusive reaction term $\lambda = i\nu$)

  The stability region for this case is presented in table 3.3, and it can be shown that the method is A-stable.
- Case 3: (mixed reaction term $\lambda = -\mu^2 + i\nu$)

  The stability region is presented in table 3.3, and it can be shown that the method is A_0-stable with a band of linear stability around the negative real axis that is larger than that of Case 1.

In addition, we compare the stability of IIE-CNLF2 method (3.17) with the stability of the second-order IMEX Modified CNAB (MCNAB2) method,

$$\begin{aligned}\mathbf{y}_{n+1} &= \mathbf{y}_n + \Delta t \left(\frac{9}{16}\mathbf{g}(t_{n+1}, \mathbf{y}_{n+1}) + \frac{3}{8}\mathbf{g}(t_n, \mathbf{y}_n) + \frac{1}{16}\mathbf{g}(t_{n-1}, \mathbf{y}_{n-1})\right) \\ &\quad + \Delta t \left(\frac{3}{2}\mathbf{f}(t_n, \mathbf{y}_n) - \frac{1}{2}\mathbf{f}(t_{n-1}, \mathbf{y}_{n-1})\right),\end{aligned} \tag{3.18}$$

and the SBDF2 method,

$$\mathbf{y}_{n+1} = \frac{4}{3}\mathbf{y}_n - \frac{1}{3}\mathbf{y}_{n-1} + \frac{2}{3}\Delta t \left(\mathbf{g}(t_{n+1}, \mathbf{y}_{n+1})\right) + \Delta t \left(\frac{4}{3}\mathbf{f}(t_n, \mathbf{y}_n) - \frac{2}{3}\mathbf{f}(t_{n-1}, \mathbf{y}_{n-1})\right). \tag{3.19}$$

It can be noted from table 3.3 that the stability region for IIE-CNLF2 method (3.17) is significantly larger than the stability regions for MCNAB2 and SBDF2 methods for the case $\lambda = i\nu$.

3.3.2 Three-step, third-order IIE-LMMs

Three-step IIE-LMMs for IVP (2.2) is presented table 3.1. A third-order method is obtained provided the IIE3 order conditions in table 3.2 are satisfied.

Stability Analysis for $k = p = 3$ IIE-LMMs

The third-degree characteristic polynomial resulting from (IIE2) applied to the test equation (3.11) is given by

$$\begin{aligned}\Phi(\xi) \equiv &\left[1 + b_{-1}^{[1]}\Delta t\mu^2 - \Delta t b_{-1}^{[2]}(\lambda_r + i\lambda_i)\right]\xi^3 + \left[a_0 - i\Delta t(b_0^{[2]}\lambda_i + b_0^{[3]}\nu) + \Delta t(\mu^2 b_0^{[1]} - b_0^{[2]}\lambda_r)\right]\xi^2 \\ &+ \left[a_1 - i\Delta t(b_1^{[2]}\lambda_i + b_1^{[3]}\nu) + \Delta t(\mu^2 b_1^{[1]} - b_1^{[2]}\lambda_r)\right]\xi \\ &+ \left[a_2 - i\Delta t(b_2^{[2]}\lambda_i + b_2^{[3]}\nu) + \Delta t(\mu^2 b_2^{[1]} - b_2^{[2]}\lambda_r)\right] = 0.\end{aligned}$$

IIE-MBDF3

A three-step, third-order method can be derived as follows

$$\begin{aligned}\mathbf{y}_{n+1} - \frac{18}{11}\mathbf{y}_n + \frac{9}{11}\mathbf{y}_{n-1} - \frac{2}{11}\mathbf{y}_{n-2} \; &= \Delta t\left(\frac{6}{11}\mathbf{f}^{[1]}(t_{n+1}, \mathbf{y}_{n+1})\right) \\ + \Delta t\left(\frac{1}{2}\mathbf{f}^{[2]}(t_{n+1}, \mathbf{y}_{n+1}) + \frac{3}{22}\mathbf{f}^{[2]}(t_n, \mathbf{y}_n) - \frac{3}{22}\mathbf{f}^{[2]}(t_{n-1}, \mathbf{y}_{n-1}) + \frac{1}{22}\mathbf{f}^{[2]}(t_{n-2}, \mathbf{y}_{n-2})\right) \quad &(3.20) \\ + \Delta t\left(\frac{18}{11}\mathbf{f}^{[3]}(t_n, \mathbf{y}_n) - \frac{18}{11}\mathbf{f}^{[3]}(t_{n-1}, \mathbf{y}_{n-1}) + \frac{6}{11}\mathbf{f}^{[3]}(t_{n-2}, \mathbf{y}_{n-2})\right).\end{aligned}$$

Because this method applies the third-order BDF method for the implicit term, we refer this method to IIE-MBDF3 (modified third-order BDF).

Stability Analysis

We discuss the stability of the IIE-MBDF3 method (3.20) for three cases.

- Case 1 (diffusive reaction term $\lambda = -\mu^2$)

  The stability region for the method (3.20) is presented in table 3.3, and it can be shown that the method is A_0-stable.

- Case 2 (non-diffusive reaction term $\lambda = i\nu$)

  The stability region for the method (3.20) is presented in table 3.3, and it can be shown that the method is A_0-stable.

- Case 3: (mixed reaction term $\lambda = -\mu^2 + i\nu$)

  The stability region for the method (3.20) is presented in table 3.3, and it can be shown that the method is A_0-stable.

In addition, we compare the stability of IIE-MBDF3 method (3.20) with the stability of IMEX Adams–Bashforth (IMEX-AB3) method,

$$\begin{aligned}\mathbf{y}_{n+1} = \mathbf{y}_n &+ \Delta t\left(\frac{4661}{10000}\mathbf{g}(t_{n+1},\mathbf{y}_{n+1}) + \frac{324}{625}\mathbf{g}(t_n,\mathbf{y}_n) + \frac{13}{200}\mathbf{g}(t_{n-1},\mathbf{y}_{n-1}) - \frac{247}{5000}\mathbf{g}(t_{n-2},\mathbf{y}_{n-2})\right)\\ &+ \Delta t\left(\frac{23}{12}\mathbf{f}(t_n,\mathbf{y}_n) - \frac{4}{3}\mathbf{f}(t_{n-1},\mathbf{y}_{n-1}) + \frac{5}{12}\mathbf{f}(t_{n-2},\mathbf{y}_{n-2})\right).\end{aligned} \tag{3.21}$$

It can be noted from table 3.3 that the stability region for the IIE-MBDF3 method (3.20) is larger than that of the IMEX-AB3 method.

3.3.3 Four-step, fourth-order IIE-LMMs

Four-step IIE-LMMs for (2.2) is presented in table 3.1. A fourth-order method is obtained provided the IIE4 order conditions in table 3.2 are satisfied.

Stability Analysis for $k = p = 4$ IIE-LMMs

The fourth-degree characteristic polynomial resulting from (IIE2) applied to the test equation (3.11) is given by

$$\begin{aligned}\Phi(\xi) \equiv &\left[1 + b_{-1}^{[1]}\Delta t\mu^2 - \Delta t b_{-1}^{[2]}(\lambda_r + i\lambda_i)\right]\xi^4 + \left[a_0 - i\Delta t(b_0^{[2]}\lambda_i + b_0^{[3]}\nu) + \Delta t(\mu^2 b_0^{[1]} - b_0^{[2]}\lambda_r)\right]\xi^3\\ &+ \left[a_1 - i\Delta t(b_1^{[2]}\lambda_i + b_1^{[3]}\nu) + \Delta t(\mu^2 b_1^{[1]} - b_1^{[2]}\lambda_r)\right]\xi^2\\ &+ \left[a_2 - i\Delta t(b_2^{[2]}\lambda_i + b_2^{[3]}\nu) + \Delta t(\mu^2 b_2^{[1]} - b_2^{[2]}\lambda_r)\right]\xi\\ &+ \left[a_3 - i\Delta t(b_3^{[2]}\lambda_i + b_3^{[3]}\nu) + \Delta t(\mu^2 b_3^{[1]} - b_3^{[2]}\lambda_r)\right] = 0.\end{aligned}$$

IIE-MBDF4

A four-step, fourth-order method can be obtained by applying the fourth-order BDF method for the implicit terms as follows

$$\begin{aligned}\mathbf{y}_{n+1} - \frac{48}{25}\mathbf{y}_n + \frac{36}{25}\mathbf{y}_{n-1} - \frac{16}{25}\mathbf{y}_{n-2} + \frac{3}{25}\mathbf{y}_{n-3} &= \Delta t \left(\frac{12}{25}\mathbf{f}^{[1]}(t_{n+1}, \mathbf{y}_{n+1})\right) \\ &\quad - \Delta t \left(\frac{12}{25}\mathbf{f}^{[2]}(t_{n+1}, \mathbf{y}_{n+1}) - \frac{96}{25}\mathbf{f}^{[2]}(t_n, \mathbf{y}_n)\right. \\ &\quad + \frac{144}{25}\mathbf{f}^{[2]}(t_{n-1}, \mathbf{y}_{n-1}) - \frac{96}{25}\mathbf{f}^{[2]}(t_{n-2}, \mathbf{y}_{n-2}) \\ &\quad \left. + \frac{24}{25}\mathbf{f}^{[2]}(t_{n-3}, \mathbf{y}_{n-3})\right) \\ &\quad + \Delta t \left(\frac{48}{25}\mathbf{f}^{[3]}(t_n, \mathbf{y}_n) - \frac{72}{25}\mathbf{f}^{[3]}(t_{n-1}, \mathbf{y}_{n-1})\right. \\ &\quad \left. + \frac{48}{25}\mathbf{f}^{[3]}(t_{n-2}, \mathbf{y}_{n-2}) - \frac{12}{25}\mathbf{f}^{[3]}(t_{n-3}, \mathbf{y}_{n-3})\right). \end{aligned} \tag{3.22}$$

We refer this method to IIE-MBDF4. We discuss the stability of the IIE-MBDF4 method (3.22) for three cases.

- Case 1: (diffusive reaction term $\lambda = -\mu^2$)

  The stability region for the method (3.22) is presented in table 3.3, where we see a relatively small linear stability region.
- Case 2: (non-diffusive reaction term $\lambda = i\nu$)

  The stability region for the method (3.22) in this case is presented in table 3.3, and it can be shown that the method is A_0-stable.
- Case 3: (mixed reaction term $\lambda = -\mu^2 + i\nu$)

  The stability region for the method (3.22) in this case is presented in table 3.3, where we see a relatively small region even compared to that of Case 1.

In summary, we get A-stability in many situations for first and second-order IIE-LMMs and A_0-stability for the second, third, and fourth-order IIE-LMMs. Accordingly, second- and third-order IIE-LMMs could be highly suitable choices for DRA problems that suffer from time step size restrictions due to stability. In addition, we compare the IIE-MBDF4 method

(3.22) with the fourth-order IMEX-SBDF4 method,

$$\begin{aligned}\mathbf{y}_{n+1} =& \frac{48}{25}\mathbf{y}_n - \frac{36}{25}\mathbf{y}_{n-1} + \frac{16}{25}\mathbf{y}_{n-2} - \frac{3}{25}\mathbf{y}_{n-3} + \Delta t\left(\frac{12}{25}\mathbf{g}(t_{n+1}, \mathbf{y}_{n+1})\right) \\ &+ \Delta t\left(\frac{48}{25}\mathbf{f}(t_n, \mathbf{y}_n) - \frac{72}{25}\mathbf{f}(t_{n-1}, \mathbf{y}_{n-1}) + \frac{48}{25}\mathbf{f}(t_{n-2}, \mathbf{y}_{n-2}) - \frac{12}{25}\mathbf{f}(t_{n-3}, \mathbf{y}_{n-3})\right).\end{aligned} \tag{3.23}$$

It can be noted from table 3.3 that the stability region for the IIE-MBDF4 method (3.22) is larger than the stability region of the IMEX-SBDF4 method for the case $\lambda = i\nu$.

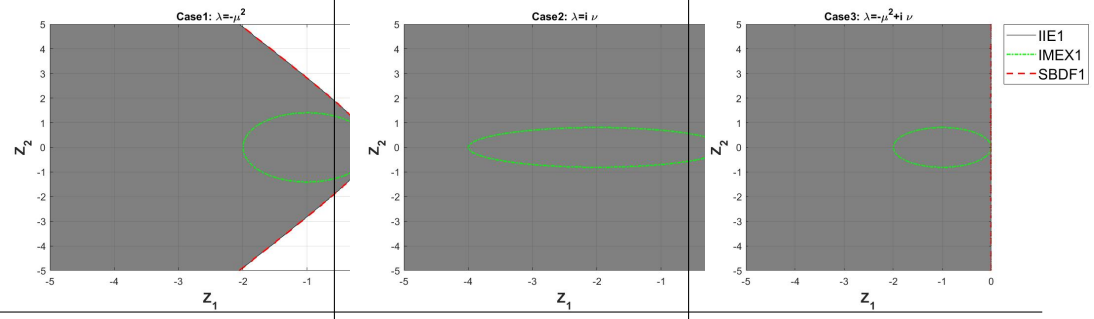
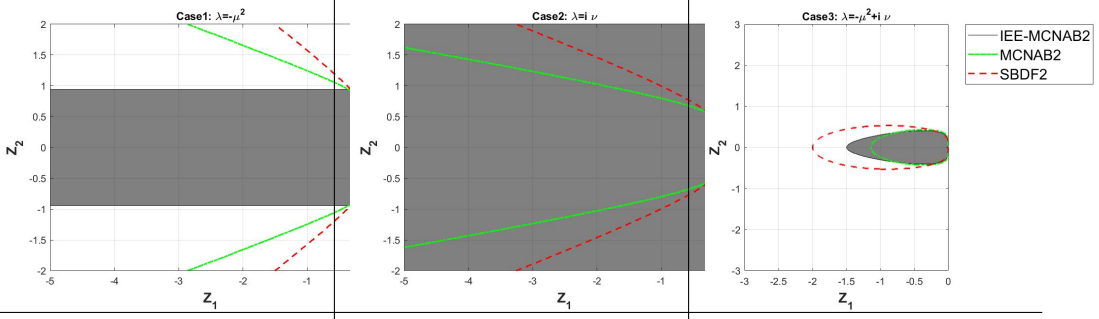
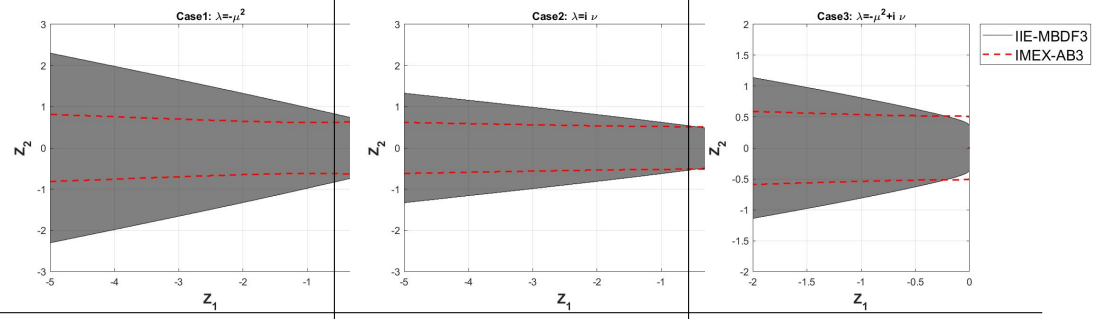
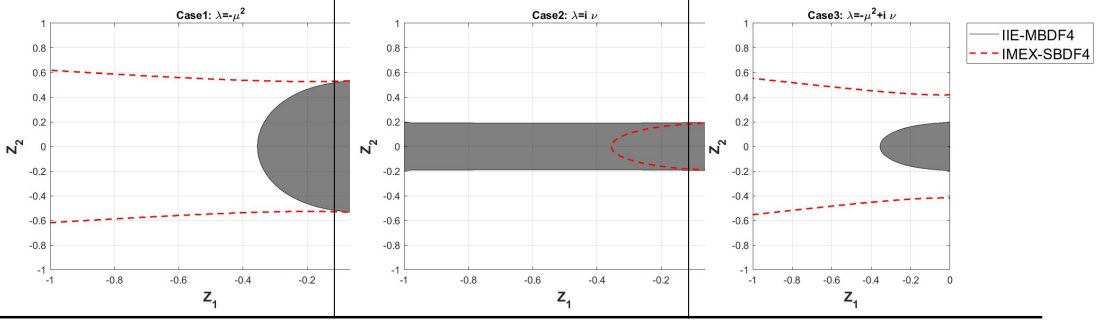
P	Case 1: $\lambda = -\mu^2$	Case 2: $\lambda = i\nu$	Case 3: $\lambda = -\mu^2 + i\nu$
$P = 1$			
$P = 2$			
$P = 3$			
$P = 4$			

Table 3.3: Stability regions for IIE-LMMs

3.4 Numerical experiments

In this section, we test the convergence order of the constructed IIE methods on a spatially discretized nonlinear DRA problem. Also, we compare the performance of the IIE splitting methods with some well-known IMEX methods on Brusselator DRA model, which is used to describe a reaction-diffusion system with non-linear oscillations [41]. All numerical experiments are performed using MATLAB. Highly accurate starting values for the IIE-LMMs as well as reference solutions for the method-of-lines (MOL) ODEs obtained from spatial discretization of the PDEs are calculated using the variable-stepsize, variable-order solver ode15s in MATLAB with an absolute tolerance of (10^{-14}) and a relative tolerance of 2.5×10^{-14}. The nonlinear algebraic equations associated with the implicit methods are solved using a modified Newton's method and the linear systems arising at each Newton iteration are solved using the MATLAB mldivide operator (\) [1]. The mldivide operator applied to sparse matrices implements a specific direct solver depending on the characteristics of the input matrix. For general sparse matrix, a sparse LU solver is used.

3.4.1 Order of convergence

To examine the convergence order of the constructed IIE methods for solving DRA problems, we consider the nonlinear problem

$$\begin{aligned} u_t + uu_x &= u_{xx} + u + f(x,t), \\ u(x,0) &= \sin(2\pi x), \\ u(0,t) &= u(1,t), \end{aligned} \tag{3.24}$$

where

$$f(x,t) = \cos(2\pi x + t) + 2\pi \sin(2\pi x + t)\cos(2\pi x + t) + 4\pi^2 \sin(2\pi x + t) - \sin(2\pi x + t)$$

The problem (3.24) is posed on $t \in [0, 10]$ and $x \in [0, 1]$. We note that the equation (3.24) is equivalent to $u_t + (\frac{u^2}{2})_x = u_{xx} + u + f(x,t)$. The spatial domain is uniformly discretized into N subintervals with spacing $\Delta x = 1/N$; a second-order centered finite discretization for

advection and diffusion terms yields a system of ODEs written as

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\left(\frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x}\right) + \left(\frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2}\right) + u_i + f(i\Delta x, t), \tag{3.25}$$

where $i = 1, 2, \ldots, N$. In this experiment, we also investigate the order of convergence of the methods constructed in chapter 3 in the maximum norm $\|.\|_\infty$ of errors computed by comparing against a reference solution for equation 3.25 generated using the ode15s solver in MATLAB, at the end point of the interval of integration as a function of the time step. Specifically, we calculate the slope of the line of best fit to the $\log(\|\text{error}\|_\infty)$-$\log(\Delta t)$ data. The results for the constructed IIE methods are presented in fig. 3.1. We can see that all methods achieve their expected orders of accuracy.

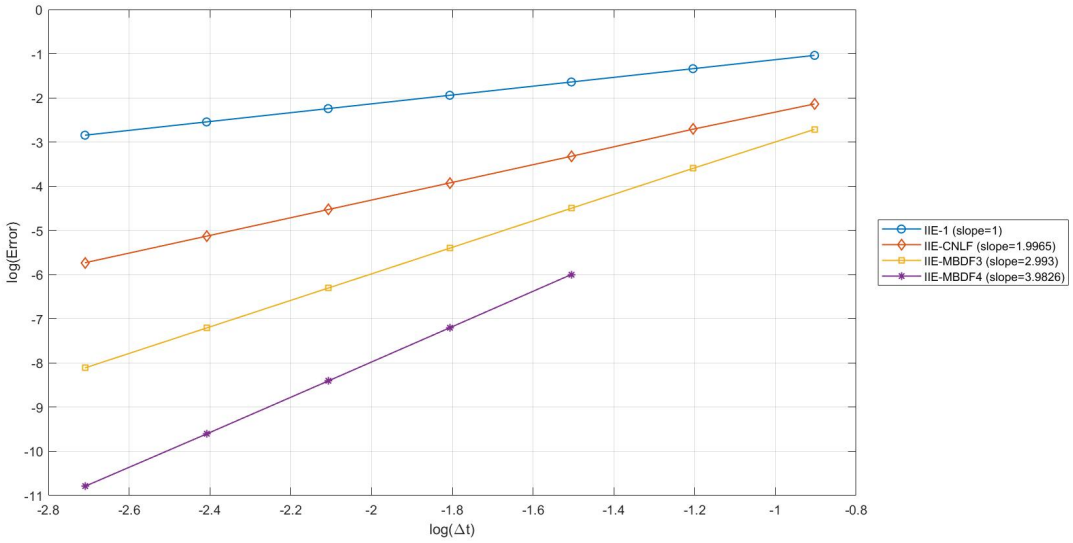


Figure 3.1: log-log plot of error vs. time step and line of best fit for DRA equation (3.25) solved by IIE splitting methods.

3.4.2 IIE-LMMs Performance Results

This section compares the performance results of the IIE methods introduced in section 3 with some popular IMEX splitting methods derived in [51]. The comparisons are made for

the following stiff variation of the standard Brusselator problem [18],

$$\begin{aligned} u_t &= \alpha_u \nabla^2 u - \rho_u \nabla u + a - (w+1)u + u^2 v, \\ v_t &= \alpha_v \nabla^2 v - \rho_v \nabla v + wu - u^2 v, \\ w_t &= \alpha_w \nabla^2 w - \rho_w \nabla w + \frac{b-w}{\epsilon} - wu, \end{aligned} \tag{3.26}$$

solved on $t \in [0, 10]$ and $x \in [0, 1]$ using stationary boundary conditions,

$$u_t(t,0) = u_t(t,1) = v_t(t,0) = v_t(t,1) = w_t(t,0) = w_t(t,1) = 0,$$

and initial values

$$\begin{aligned} u(0,x) &= a + 0.1\sin(\pi x), \\ v(0,x) &= \frac{b}{a} + 0.1\sin(\pi x), \\ w(0,x) &= b + 0.1\sin(\pi x), \end{aligned} \tag{3.27}$$

with parameters $\alpha_u = \alpha_v = \alpha_w = 10^{-2}$, $\rho_u = \rho_v = \rho_w = 10^{-3}$, $a = 0.6$, $b = 2$, and $\epsilon = 10^{-2}$. The problem (3.26) is discretized uniformly in space using a second-order accurate centered difference approximation with 100 grid points. The right-hand side of problem (3.26) is split into three terms as

$$\mathbf{f}^{[1]} = \begin{bmatrix} \alpha_u \nabla^2 u \\ \alpha_v \nabla^2 v \\ \alpha_w \nabla^2 w \end{bmatrix}, \quad \mathbf{f}^{[2]} = \begin{bmatrix} a - (w+1)u + u^2 v \\ wu - u^2 v \\ \frac{b-w}{\epsilon} - wu \end{bmatrix}, \quad \text{and } \mathbf{f}^{[3]} = \begin{bmatrix} -\rho_u \nabla u \\ -\rho_v \nabla v \\ -\rho_w \nabla w \end{bmatrix}. \tag{3.28}$$

A reference solution for the MOL ODEs arising from the spatial discretization of equation (3.26) as described is generated using ode15s with an absolute tolerance of 10^{-20} and a relative tolerance of 2.5×10^{-14}. The error is measured by the mixed root mean square (MRMS) error [14]

$$\text{MRMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{Y_i - y_i}{1 + |Y_i|}\right)^2},$$

where N is the number of solution points, Y_i, and y_i denote the reference and numerical solutions, respectively. Each experiment is performed ten times to mitigate random influences on it; the minimum time measurement of the ten runs is considered because a minimum

time gives an indicator of how well a given method performs under ideal conditions. We use a uniform time step size in each experiment, $\Delta t = \frac{2^{-J}}{80}, J = 1, 2, \ldots, 5$. Work-precision diagrams are used to compare the performance of the various methods, where the log(MRMS error) of the numerical methods are plotted on the x-axis and the log(CPU time) is plotted on the y-axis.

IIE-LMMs Performance Results

First-Order Methods

We compare the first-order IIE1 method (3.14) with the IMEX1 method (A.17) and the SBDF1 (3.16).

Second-order Methods

We now compare the IIE-CNLF2 method (3.17) with the second-order IMEX Modified CNAB (MCNAB2) method (3.18) and the SBDF2 method (3.19).

Third-order Methods

Here, we compare the IIE-MBDF3 method (3.20) with the IMEX Adams–Bashforth (IMEX-AB3) method (3.21).

Fourth-order Methods

Finally, we compare the IIE-MBDF4 method (3.22) with the fourth-order method IMEX-SBDF4 (3.23).

The work-precision diagram of these methods is given fig. 3.2, and it shows that the IIE-MBDF3 and IIE-MBDF4 outperform the other methods tested for small stepsize. The IIE1 significantly outperforms its first-order IMEX counterparts.

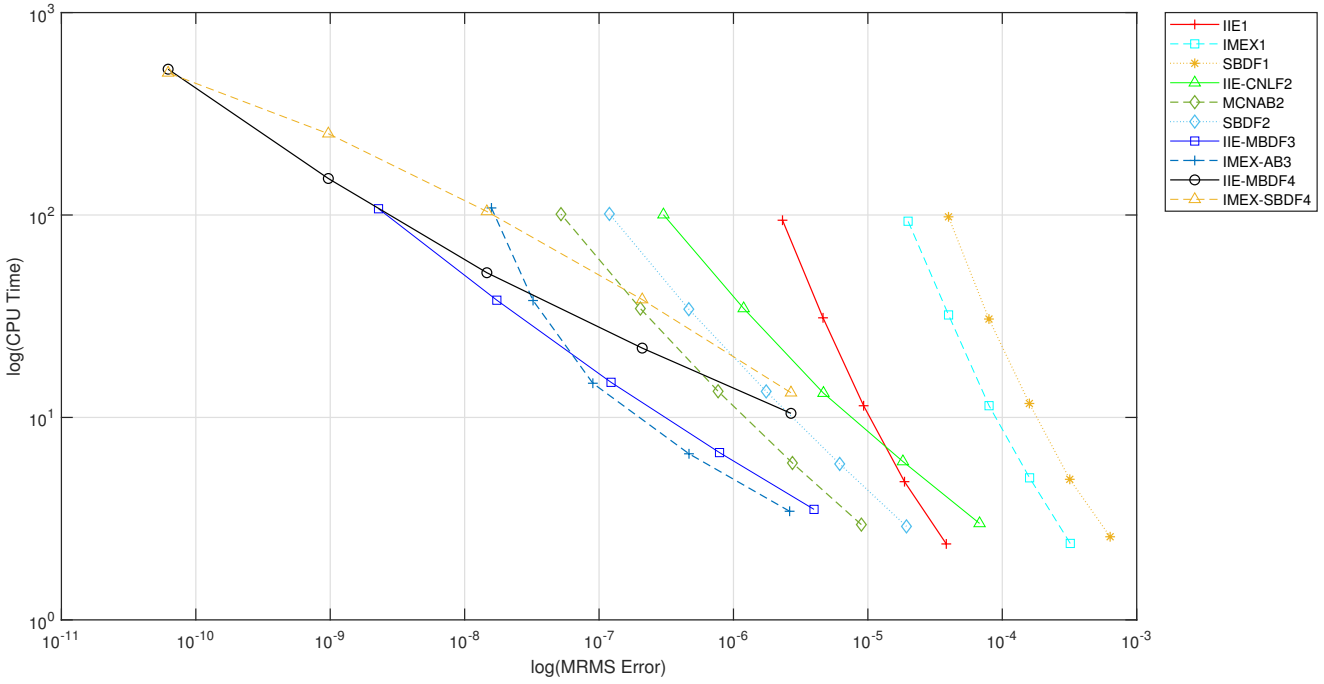


Figure 3.2: CPU time versus accuracy

Chapter 4

k-step IEE linear multistep methods.

In this chapter, we explore efficient 3-additive splitting method schemes for solving DRA problems. These methods treat the diffusion term implicitly, while handling the reaction and advection terms explicitly. We also present a stability analysis and discuss the numerical performance of the derived methods.

4.1 k-step IIE linear multistep methods

In this section, we derive k-step IEE methods for the IVP (2.2). If the implicit method is applied to $\mathbf{f}^{[1]}(t, \mathbf{y})$, whereas the explicit methods are applied to $\mathbf{f}^{[2]}(t, \mathbf{y})$ and $\mathbf{f}^{[3]}(t, \mathbf{y})$, then k-step IEE-LMMs for (2.2) can be written as

$$\frac{1}{\Delta t}\mathbf{y}_{n+1} + \frac{1}{\Delta t}\sum_{j=0}^{k-1} a_j \mathbf{y}_{n-j} = \sum_{j=-1}^{k-1} b_j^{[1]} \mathbf{f}^{[1]}(t_{n-j}, \mathbf{y}_{n-j}) + \sum_{j=0}^{k-1} b_j^{[2]} \mathbf{f}^{[2]}(t_{n-j}, \mathbf{y}_{n-j}) + \sum_{j=0}^{k-1} b_j^{[3]} \mathbf{f}^{[3]}(t_{n-j}, \mathbf{y}_{n-j}), \tag{4.1}$$

where we assume that $b_{-1}^{[1]} \neq 0$. Expanding (4.1) in a Taylor series about t_n to obtain the truncation error yields

$$
\begin{aligned}
1+\sum_{j=0}^{k-1} a_j &= 0,\\
1-\sum_{j=1}^{k-1} j a_j &= \sum_{j=-1}^{k-1} b_j^{[1]} = \sum_{j=0}^{k-1} b_j^{[2]} = \sum_{j=0}^{k-1} b_j^{[3]},\\
\frac{1}{2}+\sum_{j=1}^{k-1} \frac{j^2}{2} a_j &= b_{-1}^{[1]} - \sum_{j=1}^{k-1} j b_j^{[1]} = -\sum_{j=1}^{k-1} j b_j^{[2]} = -\sum_{j=1}^{k-1} j b_j^{[3]},\\
&\vdots\\
\frac{1}{p!}+\sum_{j=1}^{k-1} \frac{(-j)^p}{p!} a_j &= \frac{b_{-1}^{[1]}}{(p-1)!} + \sum_{j=1}^{k-1} \frac{(-j)^{p-1} b_j^{[1]}}{(p-1)!} = \sum_{j=1}^{k-1} \frac{(-j)^{p-1} b_j^{[2]}}{(p-1)!} = \sum_{j=1}^{k-1} \frac{(-j)^{p-1} b_j^{[3]}}{(p-1)!}.
\end{aligned}
\tag{4.2}
$$

The system (4.2) gives the conditions for a k-step IEE-LMM to have order p. The following theorem investigates the maximal order for a given k

Theorem 2. *For the k-step IEE-LMMs (4.1), we have:*

I. If $p \leq k$, then the $3p+1$ conditions of the system (4.2) are linearly independent. So there exist k-step IEE-LMMs of order k.

II. The order of accuracy of a k-step LMM is at most k.

III. The family of k-step IEE-LMM of order k has k free parameters.

Similar to the argument in section 3, one can formulate the following proof.

Proof. I. Because linear independence for $p = k$ implies linear independence for $p \leq k$, it is sufficient to prove the case $p = k$ only. The system (4.2) can be represented in the matrix form

$$\mathbf{WV} = \mathbf{U}, \tag{4.3}$$

where

$$\mathbf{V} = [a_0, \ldots, a_{k-1}, b_{-1}^{[1]}, b_0^{[1]}, \ldots, b_{k-1}^{[1]}, b_0^{[2]}, \ldots, b_{k-1}^{[2]}, b_0^{[3]}, b_1^{[3]}, \ldots, b_{k-1}^{[3]}]^T \in \mathbb{R}^{4k+1}, \tag{4.4}$$

and

$$\mathbf{U} = [-1, \ldots, -1_{k+1}, 0, \ldots, 0_{2k}]^T \in \mathbb{R}^{3k+1}, \tag{4.5}$$

the coefficients matrix is defined as:

$$\mathbf{W} = \left[\begin{array}{c|c|ccc} \mathbf{A} & \mathbf{L} & 0 & \dots & 0 \\ \mathbf{H} & & -\mathbf{DA} & \mathbf{0} & \mathbf{0} \\ \hline & & & & \\ \mathbf{0} & \mathbf{1} & \mathbf{A} & -\mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A} & -\mathbf{A} \end{array}\right]_{(3k+1)\times(4k+1)}, \tag{4.6}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & -1 & -2 & \dots & 1-k \\ 0 & 1 & 4 & \dots & (1-k)^2 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & (-1)^{k-1} & (-2)^{k-1} & \dots & (1-k)^{k-1} \end{bmatrix}_{k\times k}, \tag{4.7}$$

$$\mathbf{D} = \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & s \end{bmatrix}_{k\times k}, \tag{4.8}$$

$$\mathbf{H} = \Big[0, (-1)^k, \dots, (1-k)^k\Big]_{1\times k}, \tag{4.9}$$

$$\mathbf{L} = \Big[0, -1, -2, \dots, -k\Big]^T_{1\times(k+1)}, \tag{4.10}$$

and $\mathbf{1}$ is a vector of ones with size k.

The matrix $\mathbf{A}$ is a Vandermonde matrix for the distinct numbers $\{0, -1, -2, \dots, 1-k\}$. Because the Vandermonde matrix $\mathbf{A}$ is nonsingular [27], it can be noted that columns $1, k{+}2, \dots, 4k{+}1$, of the matrix $\mathbf{W}$ are linearly independent. Thus all $(3p{+}1)$ conditions are linearly independent and the system (4.2) admits a $(4k-3p+1)$-parameter family of solutions for $p \le k$. Since column $(k+1)$ of matrix $\mathbf{W}$ does not figure into the above proof we know $b_{-1}^{[1]}$ can have any nonzero value. This property verifies that the family of methods is IEE-LMM.

II. Assume the IEE-LMM method has accuracy $\mathcal{O}((\Delta t)^{k+r})$, $r \ge 1$; then the following

conditions should be satisfied:

$$\begin{aligned}
\sum_{j=-1}^{k-1} b_j^{[1]} &= \sum_{j=0}^{k-1} b_j^{[3]}, \\
b_{-1}^{[1]} + \sum_{j=1}^{k-1} (-j) b_j^{[1]} &= \sum_{j=1}^{k-1} (-j) b_j^{[3]}, \\
&\vdots \\
b_{-1}^{[1]} + \sum_{j=1}^{k-1} (-j)^k b_j^{[1]} &= \sum_{j=1}^{k-1} (-j)^k b_j^{[3]}.
\end{aligned} \tag{4.11}$$

Let $\mu_j = b_j^{[1]} - b_j^{[3]}$, then the system (4.11) can be written as

$$\begin{aligned}
b_{-1}^{[1]} + \sum_{j=0}^{k-1} \mu_j &= 0, \\
b_{-1}^{[1]} + \sum_{j=1}^{k-1} (-j) \mu_j &= 0, \\
&\vdots \\
b_{-1}^{[1]} + \sum_{j=1}^{k-1} (-j)^k \mu_j &= 0.
\end{aligned} \tag{4.12}$$

Writing the system (4.12) in a matrix form yields

$$\begin{bmatrix}
1 & 1 & 1 & \dots & 1 \\
1 & 0 & -1 & \dots & 1-k \\
\vdots & \vdots & \vdots & \dots & \vdots \\
1 & 0 & (-1)^k & \dots & (1-k)^k
\end{bmatrix}_{(k+1)\times(k+1)}
\begin{bmatrix}
b_{-1}^{[1]} \\ \mu_0 \\ \vdots \\ \mu_{k-1}
\end{bmatrix}_{(k+1)\times 1}
= \mathbf{0}, \tag{4.13}$$

implying that $b_{-1}^{[1]} = 0$ and $b_j^{[2]} = b_j^{[3]}, j = 1, 2, \dots, k-1$, because the matrix of coefficient is nonsingular. This contradicts the assumption $b_{-1}^{[1]} \neq 0$.

III. Consider a k-step IIE-LMM method that achieves the greatest accuracy. Then, as shown above, the family of such method has exactly $(k+1)$ free parameters.

□

Corollary 1. *Any k-step IEE-LMM of order k reduces to an IMEX-LMM.*

Proof. Assume the k-step IEE-LMM has order k; then we have

$$\begin{aligned}
\sum_{j=0}^{k-1} b_j^{[2]} &= \sum_{j=0}^{k-1} b_j^{[3]}, \\
\sum_{j=1}^{k-1} (-j) b_j^{[2]} &= \sum_{j=1}^{k-1} (-j) b_j^{[3]}, \\
&\vdots \\
\sum_{j=1}^{k-1} (-j)^{k-1} b_j^{[2]} &= \sum_{j=1}^{k-1} (-j)^{k-1} b_j^{[3]}.
\end{aligned} \tag{4.14}$$

Let $\mu_j = b_j^{[2]} - b_j^{[3]}$, then the system (4.14) is written as

$$\begin{aligned}
\sum_{j=0}^{k-1} \mu_j &= 0, \\
\sum_{j=1}^{k-1} (-j) \mu_j &= 0, \\
&\vdots \\
\sum_{j=1}^{k-1} (-j)^{k-1} \mu_j &= 0.
\end{aligned} \tag{4.15}$$

Writing the system (4.15) in a matrix form yields

$$\begin{bmatrix}
1 & 1 & \dots & 1 & \\
0 & -1 & \dots & 1-k & \\
\vdots & \vdots & \vdots & \dots & \vdots \\
0 & (-1)^k & \dots & (1-k)^k &
\end{bmatrix}_{k \times k}
\begin{bmatrix}
\mu_0 \\ \mu_1 \\ \vdots \\ \mu_{k-1}
\end{bmatrix}_{k \times 1}
= \mathbf{0},$$

implying that $b_j^{[2]} = b_j^{[3]}, j = 1, 2, \dots, k-1$, because the matrix of coefficient is nonsingular. So the method is IMEX-LMM. □

Therefore, we restrict the analysis to k-step IEE-LMMs of order $p = k - 1$.

4.1.1 Linear stability analysis for IEE-LMMs

Consider the scalar test problem (3.11). Applying IEE-LMM (4.1) to (3.11) yields

$$\left[1+\Delta t b_{-1}^{[1]}\mu^2\right] y_{n+1} + \sum_{j=0}^{k-1}\left[a_j - i\Delta t(b_j^{[2]}\lambda_i + b_j^{[3]}\nu) + \Delta t(b_j^{[1]}\mu^2 - b_j^{[2]}\lambda_r)\right] y_{n-j} = 0.$$

If this linear constant-coefficient difference equation has solutions of the form $\mathbf{y}_n = \xi^n$, then the characteristic equation is given by

$$\Phi(\xi) \equiv \left[1+b_{-1}^{[1]}\Delta t\mu^2\right]\xi^k + \sum_{j=0}^{k-1}\left[a_j - i\Delta t(b_j^{[2]}\lambda_i + b_j^{[3]}\nu) + \Delta t(b_j^{[1]}\mu^2 - b_j^{[2]}\lambda_r)\right]\xi^{k-j-1}. \quad (4.16)$$

Similar to analysis in chapter 3, we discuss the stability of IEE-LMMs in three cases $\lambda = i\nu$, $\lambda = -\mu^2$ and $\lambda = -\mu^2 + i\nu$.

4.2 Construction and Stability of IEE-LMMs

In this section, we construct IEE-LMMs by using the coefficients of well-known IMEX methods for the $\mathbf{f}^{[1]}$ (diffusion) and $\mathbf{f}^{[3]}$ (advection) terms, and then derive coefficients of $\mathbf{f}^{[2]}$ (reaction) terms that satisfy the relevant order conditions (4.2). The IEE-LMMs up to third order are presented in 4.1. The order conditions for IEE-LMMs methods are summarized in table 4.2. We then also analyze the linear stability of the constructed methods using eq. (3.11) in the three cases described. The linear stability regions for IEE-LMMs up to fourth order are presented in table 4.3.

k	Methods	
$k=2$	$\mathbf{y}_{n+1}+a_0\mathbf{y}_n+a_1\mathbf{y}_{n-1}=\Delta t\left(b_{-1}^{[1]}\mathbf{f}^{[1]}(t_{n+1},\mathbf{y}_{n+1})+b_0^{[1]}\mathbf{f}^{[1]}(t_n,\mathbf{y}_n)+b_1^{[1]}\mathbf{f}^{[1]}(t_{n-1},\mathbf{y}_{n-1})\right)+$ $\Delta t\left(b_0^{[2]}\mathbf{f}^{[2]}(t_n,\mathbf{y}_n)+b_1^{[2]}\mathbf{f}^{[2]}(t_{n-1},\mathbf{y}_{n-1})\right)+\Delta t\left(b_0^{[3]}\mathbf{f}^{[3]}(t_n,\mathbf{y}_n)+b_1^{[3]}\mathbf{f}^{[3]}(t_{n-1},\mathbf{y}_{n-1})\right).$	IEE2
$k=3$	$\mathbf{y}_{n+1}+a_0\mathbf{y}_n+a_1\mathbf{y}_{n-1}+a_2\mathbf{y}_{n-2}=$ $\Delta t\left(b_{-1}^{[1]}\mathbf{f}^{[1]}(t_{n+1},\mathbf{y}_{n+1})+b_0^{[1]}\mathbf{f}^{[1]}(t_n,\mathbf{y}_n)+b_1^{[1]}\mathbf{f}^{[1]}(t_{n-1},\mathbf{y}_{n-1})+b_2^{[1]}\mathbf{f}^{[1]}(t_{n-2},\mathbf{y}_{n-2})\right)+$ $\Delta t\left(b_0^{[2]}\mathbf{f}^{[2]}(t_n,\mathbf{y}_n)+b_1^{[2]}\mathbf{f}^{[2]}(t_{n-1},\mathbf{y}_{n-1})+b_2^{[2]}\mathbf{f}^{[2]}(t_{n-2},\mathbf{y}_{n-2})\right)+$ $\Delta t\left(b_0^{[3]}\mathbf{f}^{[3]}(t_n,\mathbf{y}_n)+b_1^{[3]}\mathbf{f}^{[3]}(t_{n-1},\mathbf{y}_{n-1})+b_2^{[3]}\mathbf{f}^{[3]}(t_{n-2},\mathbf{y}_{n-2})\right).$	IEE3
$k=4$	$\mathbf{y}_{n+1}+a_0\mathbf{y}_n+a_1\mathbf{y}_{n-1}+a_2\mathbf{y}_{n-2}+a_3\mathbf{y}_{n-3}=$ $\Delta t\left(b_{-1}^{[1]}\mathbf{f}^{[1]}(t_{n+1},\mathbf{y}_{n+1})+b_0^{[1]}\mathbf{f}^{[1]}(t_n,\mathbf{y}_n)+b_1^{[1]}\mathbf{f}^{[1]}(t_{n-1},\mathbf{y}_{n-1})+b_2^{[1]}\mathbf{f}^{[1]}(t_{n-2},\mathbf{y}_{n-2})+b_3^{[1]}\mathbf{f}^{[1]}(t_{n-3},\mathbf{y}_{n-3})\right)$ $+\Delta t\left(b_0^{[2]}\mathbf{f}^{[2]}(t_n,\mathbf{y}_n)+b_1^{[2]}\mathbf{f}^{[2]}(t_{n-1},\mathbf{y}_{n-1})+b_2^{[2]}\mathbf{f}^{[2]}(t_{n-2},\mathbf{y}_{n-2})+b_3^{[2]}\mathbf{f}^{[2]}(t_{n-3},\mathbf{y}_{n-3})\right)$ $+\Delta t\left(b_0^{[3]}\mathbf{f}^{[3]}(t_n,\mathbf{y}_n)+b_1^{[3]}\mathbf{f}^{[3]}(t_{n-1},\mathbf{y}_{n-1})+b_2^{[3]}\mathbf{f}^{[3]}(t_{n-2},\mathbf{y}_{n-2})+b_3^{[3]}\mathbf{f}^{[3]}(t_{n-3},\mathbf{y}_{n-3})\right).$	IEE4

Table 4.1: IEE-LMMs

Table 4.2: IEE-LMMs Order Conditions

k	Order Conditions	
$k=2$	$1+a_0+a_1 = 0,$ $1-a_1 = b_{-1}^{[1]}+b_0^{[1]}+b_1^{[1]},$ $1-a_1 = b_0^{[2]}+b_1^{[2]},$ $1-a_1 = b_0^{[3]}+b_1^{[3]}.$	IEE2 order conditions
$k=3$	$1+a_0+a_1+a_2 = 0,$ $1-a_1-2a_2 = b_{-1}^{[1]}+b_0^{[1]}+b_1^{[1]}+b_2^{[1]},$ $1-a_1-2a_2 = b_0^{[2]}+b_1^{[2]}+b_2^{[2]},$ $1-a_1-2a_2 = b_0^{[3]}+b_1^{[3]}+b_2^{[3]},$ $1+a_1+4a_2 = 2b_{-1}^{[1]}-2b_1^{[1]}-4b_2^{[1]},$ $1+a_1+4a_2 = -2b_1^{[2]}-4b_2^{[2]},$ $1+a_1+4a_2 = -2b_1^{[3]}-4b_2^{[3]}.$	IEE3 order conditions
$k=4$	$1+a_0+a_1+a_2+a_3 = 0,$ $1-a_1-2a_2-3a_3 = b_{-1}^{[1]}+b_0^{[1]}+b_1^{[1]}+b_2^{[1]}+b_3^{[1]},$ $1-a_1-2a_2-3a_3 = b_0^{[2]}+b_1^{[2]}+b_2^{[2]}+b_3^{[2]},$ $1-a_1-2a_2-3a_3 = b_0^{[3]}+b_1^{[3]}+b_2^{[3]}+b_3^{[3]},$ $1+a_1+4a_2+9a_3 = 2b_{-1}^{[1]}-2b_1^{[1]}-4b_2^{[1]}-6b_3^{[1]},$ $1+a_1+4a_2+9a_3 = -2b_1^{[2]}-4b_2^{[2]}-6b_3^{[2]},$ $1+a_1+4a_2+9a_3 = -2b_1^{[3]}-4b_2^{[3]}-6b_3^{[3]},$ $1-a_1-8a_2-27a_3 = 3b_{-1}^{[1]}+3b_1^{[1]}+12b_2^{[1]}+27b_3^{[1]},$ $1-a_1-8a_2-27a_3 = 3b_1^{[2]}+12b_2^{[2]}+27b_3^{[2]},$ $1-a_1-8a_2-27a_3 = 3b_1^{[3]}+12b_2^{[3]}+27b_3^{[3]}.$	IEE4 order conditions

One-step, first-order IEE-LMMs

The first-order IEE-LMMs for the IVP (2.2) can be written as

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \left[\alpha \mathbf{f}^{[1]}(t_{n+1}, \mathbf{y}_{n+1}) + \beta \mathbf{f}^{[2]}(t_n, \mathbf{y}_n) + \gamma \mathbf{f}^{[3]}(t_n, \mathbf{y}_n)\right]. \tag{4.17}$$

For a smooth function $\mathbf{y}(t)$, we expand (4.17) in a Taylor series about $t_n = n\Delta t$ to obtain the truncation error. This yields

$$\mathbf{y}_n + \Delta t\dot{\mathbf{y}}_n + \frac{(\Delta t)^2}{2}\ddot{\mathbf{y}}_n + \cdots = \mathbf{y}_n + \Delta t\alpha \left[\mathbf{f}^{[1]}(t_n, \mathbf{y}_n) + \Delta t \frac{\mathrm{d}\mathbf{f}^{[1]}(t_n, \mathbf{y}_n)}{\mathrm{d}t} + \frac{(\Delta t)^2}{2}\frac{\mathrm{d}^2\mathbf{f}^{[1]}(t_n, \mathbf{y}_n)}{\mathrm{d}t^2} + \ldots\right] + \Delta t\beta \mathbf{f}^{[2]}(t_n, \mathbf{y}_n) + \Delta t\gamma \mathbf{f}^{[3]}(t_n, \mathbf{y}_n). \tag{4.18}$$

Applying (2.2) to the truncation error (4.18) provides:

- Coefficients of $\mathbf{f}^{[1]}(t_n, \mathbf{y}_n)$: $\Delta t(1-\alpha) = 0$, implying that $\alpha = 1$.
- Coefficients of $\mathbf{f}^{[2]}(t_n, \mathbf{y}_n)$: $\Delta t(1-\beta) = 0$, implying that $\beta = 1$.
- Coefficients of $\mathbf{f}^{[3]}(t_n, \mathbf{y}_n)$: $\Delta t(1-\gamma) = 0$, implying that $\gamma = 1$.

Therefore, the method (4.17) turns into an IMEX method, in agreement with the result of Corollary 1.

Two-step, first-order IEE-LMMs

Two-step IEE-LMMs for IVP (2.2) is presented in table 4.1. A first-order method is obtained provided the IEE2 order conditions in table 4.2 are satisfied.

Stability analysis for two-step, first-order IEE-LMMs

The second-degree characteristic polynomial resulting from IEE2 applied to the test equation (3.11) is given by

$$\begin{aligned}\Phi(\xi) \equiv &\left[1 + b_{-1}^{[1]}\Delta t\mu^2\right]\xi^2 + \left[a_0 - i\Delta t(b_0^{[2]}\lambda_i + b_0^{[3]}\nu) + \Delta t(b_0^{[1]}\mu^2 - b_0^{[2]}\lambda_r)\right]\xi \\ &+ a_1 - i\Delta t(b_1^{[2]}\lambda_i + b_1^{[3]}\nu) + \Delta t(b_1^{[1]}\mu^2 - b_1^{[2]}\lambda_r) = 0\,.\end{aligned}$$

IEE-MCNAB1 method

A two-step, first-order IEE-LMM can be obtained by applying something similar to the Crank–Nicolson (CN) method for the implicit term $\mathbf{f}^{[1]}(t_{n+1}, \mathbf{y}_{n+1})$, the second-order Adams–Bashforth (AB) method for the explicit term $\mathbf{f}^{[3]}(t_n, \mathbf{y}_n)$, and a method for $\mathbf{f}^{[2]}(t_n, \mathbf{y}_n)$ derived from the solution of the IEE2 order conditions as follows:

$$\begin{aligned}\mathbf{y}_{n+1} = &\mathbf{y}_n + \frac{\Delta t}{2}\left(\mathbf{f}^{[1]}(t_{n+1}, \mathbf{y}_{n+1}) + \mathbf{f}^{[1]}(t_n, \mathbf{y}_n)\right) + \frac{\Delta t}{2}\left(\mathbf{f}^{[2]}(t_n, \mathbf{y}_n) + \mathbf{f}^{[2]}(t_{n-1}, \mathbf{y}_{n-1})\right) \\ &+ \frac{\Delta t}{2}\left(3\mathbf{f}^{[3]}(t_n, \mathbf{y}_n) - \mathbf{f}^{[3]}(t_{n-1}, \mathbf{y}_{n-1})\right).\end{aligned} \tag{4.19}$$

We refer this method to as IEE-MCNAB1 (modified first-order CNAB) method.

Stability Analysis

We discuss the stability of the IEE-MCNAB1 method (4.19) for three cases.

- Case 1: (diffusive reaction term $\lambda = -\mu^2$)

  The stability region for the method (4.19) is presented in table 4.3, and it can be shown that the method is A_0-stable.
- Case 2: (non-diffusive reaction term $\lambda = i\nu$)

  The stability region for the method (4.19) is presented in table 4.3, and it can be shown that the method is A_0-stable and appears to capture the negative real axis better than Case 1.
- Case 3: (mixed reaction term $\lambda = -\mu^2 + i\nu$)

  The stability region for the method (4.19) is presented in table 4.3. In this case, the method can be shown to be A_0-stable and captures the negative real axis intermediate to Cases 1 and 2.

In addition, we compare the two-step, first-order IEE-MCNAB1 method (4.19) with the IMEX1 method (A.17) and the semi-implicit BDF (SBDF1) (3.16).

Three-step, second-order IEE-LMMs

Three-step IEE-LMMs for IVP (2.2) are presented in table 4.1. A second-order method is obtained provided the IEE3 order conditions in table 4.2 are satisfied.

Stability Analysis for three-step, second-order IEE-LMMs

The third-degree characteristic polynomial resulting from IEE3 applied to the test equation (3.11) is given by

$$\begin{aligned}\Phi(\xi) \equiv & \left[1 + \Delta t b_{-1}^{[1]}\mu^2\right]\xi^3 + \left[a_0 - i\Delta t(b_0^{[2]}\lambda_i + b_0^{[3]}\nu) + \Delta t(\mu^2 b_0^{[1]} - b_0^{[2]}\lambda_r)\right]\xi^2 \\ & + \left[a_1 - i\Delta t(b_1^{[2]}\lambda_i + b_1^{[3]}\nu) + \Delta t(\mu^2 b_1^{[1]} - b_1^{[2]}\lambda_r)\right]\xi \\ & + \left[a_2 - i\Delta t(b_2^{[2]}\lambda_i + b_2^{[3]}\nu) + \Delta t(\mu^2 b_2^{[1]} - b_2^{[2]}\lambda_r)\right] = 0\end{aligned} \tag{4.20}$$

IEE-MCNAB2 method

A three-step, second-order method can be derived as follows

$$\begin{aligned}\mathbf{y}_{n+1}-\mathbf{y}_n &= \Delta t\left(\frac{1}{2}\mathbf{f}^{[1]}(t_{n+1},\mathbf{y}_{n+1})+\frac{1}{2}\mathbf{f}^{[1]}(t_n,\mathbf{y}_n)\right)+\Delta t\left(\frac{3}{2}\mathbf{f}^{[2]}(t_n,\mathbf{y}_n)-\frac{1}{2}\mathbf{f}^{[2]}(t_{n-1},\mathbf{y}_{n-1})\right)\\ &+\Delta t\left(\frac{4}{3}\mathbf{f}^{[3]}(t_n,\mathbf{y}_n)-\frac{1}{6}\mathbf{f}^{[3]}(t_{n-1},\mathbf{y}_{n-1})-\frac{1}{6}\mathbf{f}^{[3]}(t_{n-2},\mathbf{y}_{n-2})\right).\end{aligned} \tag{4.21}$$

This method uses the second-order CN method for the implicit term $\mathbf{f}^{[1]}(t_{n+1},\mathbf{y}_{n+1})$, the second-order AB for the explicit term $\mathbf{f}^{[2]}(t_n,\mathbf{y}_n)$, and a method for $\mathbf{f}^{[3]}(t_n,\mathbf{y}_n)$ derived from IEE3 order conditions. We refer this method to as IEE-MCNAB2 (modified second-order CNAB).

Stability Analysis

We discuss the stability of the IEE-MCNAB2 method (4.21) for three cases.

- Case 1: (diffusive reaction term $\lambda=-\mu^2$)

  The stability region for the method (4.21) is presented in table 4.3, from where we see the stability region is bounded.
- Case 2: (non-diffusive reaction term $\lambda=i\nu$)

  The stability region for the method (4.21) is presented in table 4.3, from where we see the stability region is smaller than in Case 1.
- Case 3: (mixed reaction term $\lambda=-\mu^2+i\nu$)

  The stability region for the method (4.21) is presented in table 4.3, from where we see the size of the stability region is intermediate to Cases 1 and 2.

In addition, we compare the stability region of the IEE-MCNAB2 method (4.21) with the stability region of the second-order IMEX MCNAB2 method (3.18) and the SBDF2 method (3.19).

Four-step, third-order IEE-LMMs

Four-step IEE-LMMs for IVP (2.2) is presented in table 4.1. A third-order method is obtained provided the IEE4 order conditions in table 4.2 are satisfied.

Stability Analysis for four-step, third-order IEE-LMMs

The fourth-degree characteristic polynomial resulting from IEE4 applied to the test equation (3.11) is given by

$$\begin{aligned}\Phi(\xi) \equiv & \left[1 + b_{-1}^{[1]}\Delta t\mu^2\right]\xi^4 \\ & + \left[a_0 - i\Delta t(b_0^{[2]}\lambda_i + b_0^{[3]}\nu) + \Delta t(\mu^2 b_0^{[1]} - b_0^{[2]}\lambda_r)\right]\xi^3 \\ & + \left[a_1 - i\Delta t(b_1^{[2]}\lambda_i + b_1^{[3]}\nu) + \Delta t(\mu^2 b_1^{[1]} - b_1^{[2]}\lambda_r)\right]\xi^2 \\ & + \left[a_2 - i\Delta t(b_2^{[2]}\lambda_i + b_2^{[3]}\nu) + \Delta t(\mu^2 b_2^{[1]} - b_2^{[2]}\lambda_r)\right]\xi \\ & + \left[a_3 - i\Delta t(b_3^{[2]}\lambda_i + b_3^{[3]}\nu) + \Delta t(\mu^2 b_3^{[1]} - b_3^{[2]}\lambda_r)\right] \\ = & \, 0.\end{aligned}$$

IEE-MBDF3

A four-step, third-order method can be obtained as follows:

$$\begin{aligned}\mathbf{y}_{n+1} - \frac{18}{11}\mathbf{y}_n + \frac{9}{11}\mathbf{y}_{n-1} - \frac{2}{11}\mathbf{y}_{n-2} = & \Delta t\left(\frac{6}{11}\mathbf{f}^{[1]}(t_{n+1}, \mathbf{y}_{n+1})\right) + \Delta t\left(\frac{18}{11}\mathbf{f}^{[2]}(t_n, \mathbf{y}_n)\right. \\ & \left. - \frac{18}{11}\mathbf{f}^{[2]}(t_{n-1}, \mathbf{y}_{n-1}) + \frac{6}{11}\mathbf{f}^{[2]}(t_{n-2}, \mathbf{y}_{n-2})\right) \\ & + \Delta t\left(\frac{47}{22}\mathbf{f}^{[3]}(t_n, \mathbf{y}_n) - \frac{69}{22}\mathbf{f}^{[3]}(t_{n-1}, \mathbf{y}_{n-1})\right. \\ & \left. + \frac{45}{22}\mathbf{f}^{[3]}(t_{n-2}, \mathbf{y}_{n-2}) - \frac{1}{2}\mathbf{f}^{[3]}(t_{n-3}, \mathbf{y}_{n-3})\right).\end{aligned} \tag{4.22}$$

Because this method applies the third-order BDF method for the implicit term, we refer this method to IEE-MBDF3 (modified third-order BDF).

Stability analysis

We discuss the stability of method (4.22) for three cases

- Case 1: (diffusive reaction term $\lambda = -\mu^2$)

  The stability region for the IEE-MBDF3 method (4.22) is presented in table 4.3, from where we see the stability region is bounded but captures some of the imaginary axis.

- Case 2: (non-diffusive reaction term $\lambda = i\nu$)

  The stability region for the method (4.22) is presented in table 4.3, and it can be shown that the method is A_0-stable.

- Case 3: (mixed reaction term $\lambda = -\mu^2 + i\nu$)

  The stability region for the method (4.22) is presented in table 4.3, where we see a bounded stability region that is smaller than Case 1.

  Finally, we compare the stability region of the IEE-MBDF3 method (4.22) with the stability region of the IMEX-AB3 method (3.21).

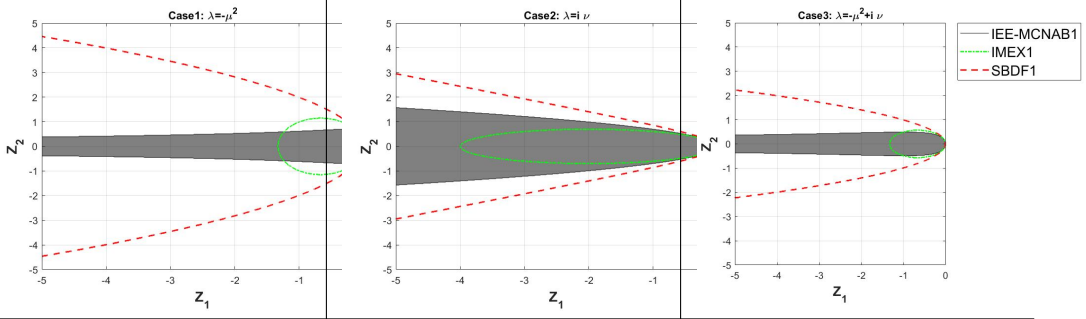
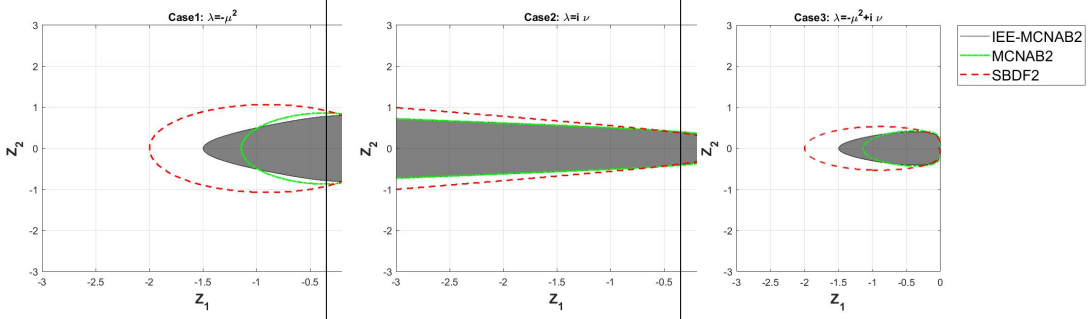
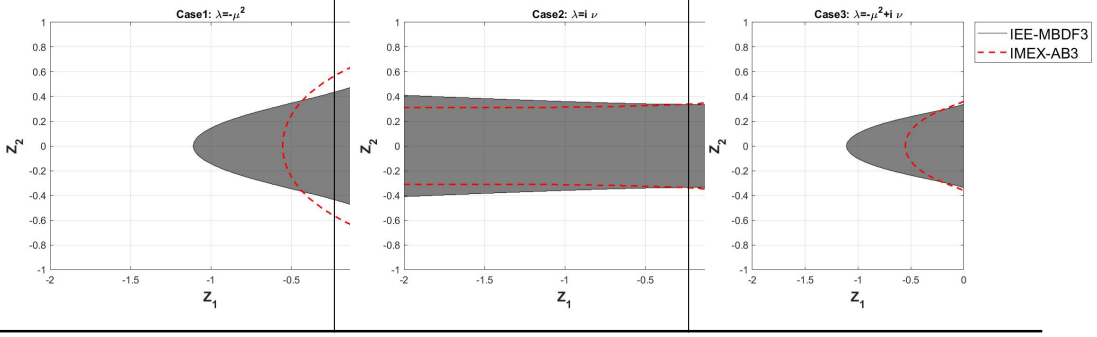
k	Case 1: $\lambda = -\mu^2$	Case 2: $\lambda = i\nu$	Case 3: $\lambda = -\mu^2 + i\nu$
$P = 2$			
$P = 3$			
$P = 4$			

Table 4.3: Stability regions for IEE-LMMs

4.3 Numerical experiments

In this section, we test the convergence order of the constructed IEE methods on the nonlinear DRA problem (3.25). Also, we compare the performance of the IEE splitting methods

with some well-known IMEX methods on the Brusselator DRA model defined by 3.26. The numerical experiments are performed using MATLAB. Highly accurate starting values for the IEE-LMMs as well as reference solutions for the method-of-lines (MOL) ODEs obtained from spatial discretization of the PDEs are calculated using the variable-stepsize, variable-order solver ode15s in MATLAB with an absolute tolerance of (10^{-14}) and a relative tolerance of 2.5×10^{-14}. The nonlinear algebraic equations associated with the implicit methods are solved using the same approach in section 3.4.2.

4.3.1 Order of convergence

The convergence order of the constructed IEE methods is tested using the nonlinear DRA problems defined by 3.24, and using the metric given in 3.4.1. The results for the constructed IEE splitting methods are presented in fig. 4.1. We can see that all methods achieve their expected orders of accuracy.

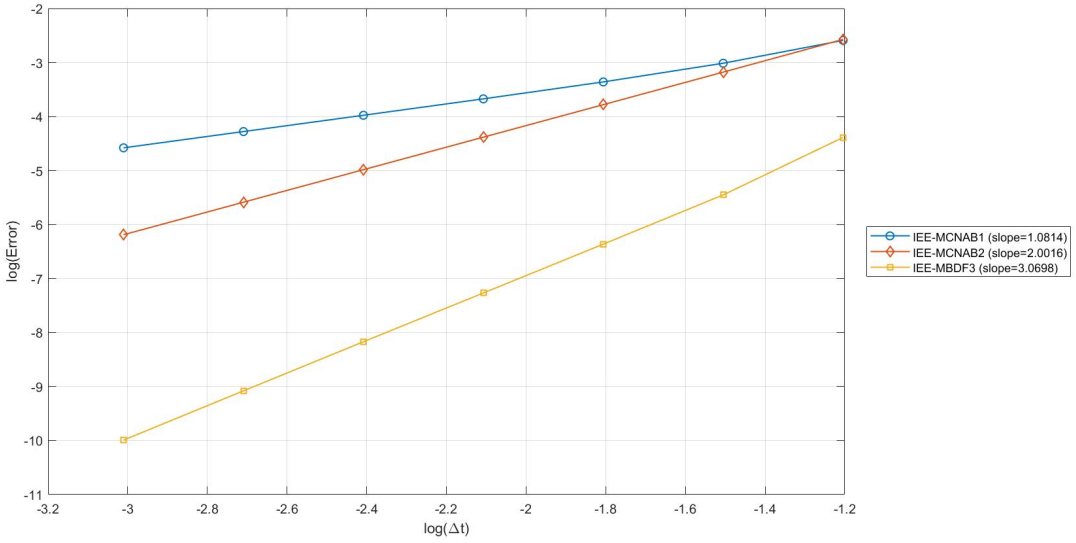


Figure 4.1: log-log plot of error vs. time step and line of best fit for DRA equation (3.25) solved by 3-additive splitting methods.

4.3.2 IEE-LMMs Performance Results

This section compares the performance results of the IEE-LMMs introduced in section 4.2 with some popular IMEX splitting methods derived in [51]. The comparisons are made for the stiff variation of the standard Brusselator problem defined by eq. (3.26). The experiment

is implemented in the same way as for the IEE methods in section 3.4.2.

First-Order Methods

We compare the two-step, first-order IEE-MCNAB1 method (4.19) with the IMEX1 method (A.17) and the semi-implicit BDF (SBDF) (3.16).

Second-order Methods

We compare the IEE-MCNAB2 method (4.21) with the second-order IMEX MCNAB2 method(3.18) and SBDF2 method (3.19)

Third-order Methods

Finally, we compare the IEE-MBDF3 method (4.22) with the IMEX-AB3 method (3.21).

The work-precision diagram of these methods is given in fig. 4.2, and it shows that the IEE-MBDF3 method outperforms the other methods tested. The IEE-MCNAB2 method outperforms the SBDF2 method and is comparable with the MCNAB2 method.

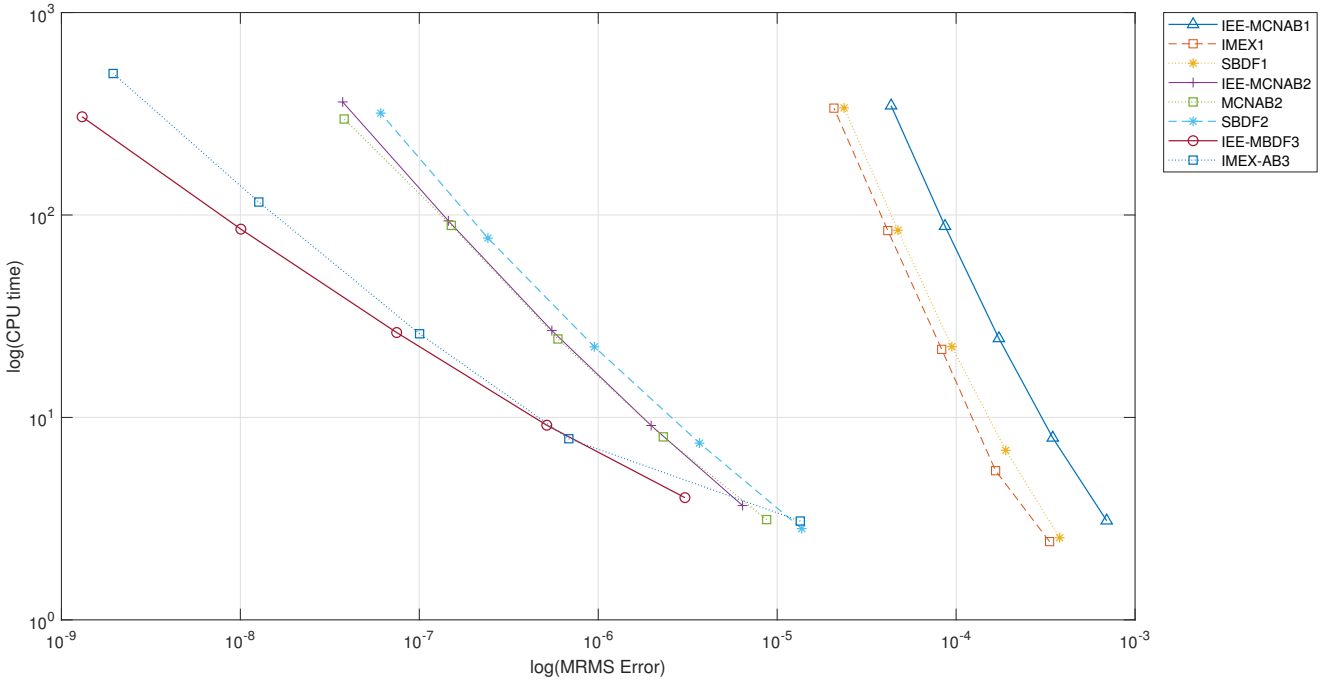


Figure 4.2: CPU time versus accuracy

Chapter 5

N-additive linear multi-step methods

In this chapter, we generalize the concept of additive splitting methods to N-additive splitting methods.

5.1 Extension to $\mathrm{I}^{N-1}\mathrm{E}$

Consider a differential equation of the form

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}(t) = \mathbf{f}^{[1]}(t,\mathbf{y}) + \mathbf{f}^{[2]}(t,\mathbf{y}) + \cdots + \mathbf{f}^{[N]}(t,\mathbf{y}). \tag{5.1}$$

If an implicit method is applied to $\mathbf{f}^{[1]}(t,\mathbf{y}), \mathbf{f}^{[2]}(t,\mathbf{y}), \ldots, \mathbf{f}^{[N-1]}(t,\mathbf{y})$, while an explicit method is applied to $\mathbf{f}^{[N]}(t,\mathbf{y})$, then the k-step I^{N-1}E-LMM can be formulated as:

$$\begin{aligned}\frac{1}{\Delta t}\mathbf{y}_{n+1} + \frac{1}{\Delta t}\sum_{j=0}^{k-1} a_j \mathbf{y}_{n-j} = &\sum_{j=-1}^{k-1} b_j^{[1]}\mathbf{f}^{[1]}(t_{n-j},\mathbf{y}_{n-j}) + \sum_{j=-1}^{k-1} b_j^{[2]}\mathbf{f}^{[2]}(t_{n-j},\mathbf{y}_{n-j}) + \cdots \\ &+ \sum_{j=-1}^{k-1} b_j^{[N-1]}\mathbf{f}^{[N-1]}(t_{n-j},\mathbf{y}_{n-j}) + \sum_{j=0}^{k-1} b_j^{[N]}\mathbf{f}^{[N]}(t_{n-j},\mathbf{y}_{n-j}),\end{aligned} \tag{5.2}$$

where we assume that $b_{-1}^{[1]}, b_{-1}^{[2]}, \ldots, b_{-1}^{[N-1]} \neq 0$.

Order conditions

The order conditions for k-step I^{N-1}E-LMMs can be derived similarly by substituting the exact solution to (5.1) into (5.2) and expanding in Taylor series about t_n and analyzing the resulting truncation error. An order-p method is obtained provided

$$
\begin{aligned}
1+\sum_{j=0}^{k-1} a_j &= 0,\\
1-\sum_{j=1}^{k-1} j a_j &= \sum_{j=-1}^{k-1} b_j^{[1]} = \cdots = \sum_{j=-1}^{k-1} b_j^{[N-1]} = \sum_{j=0}^{k-1} b_j^{[N]},\\
\frac{1}{2}+\sum_{j=1}^{k-1} \frac{j^2}{2} a_j &= b_{-1}^{[1]} - \sum_{j=1}^{k-1} j b_j^{[1]} = \cdots = b_{-1}^{[N-1]} - \sum_{j=1}^{k-1} j b_j^{[N-1]} = -\sum_{j=1}^{k-1} j b_j^{[N]},\\
&\vdots\\
\frac{1}{p!}+\sum_{j=1}^{k-1} \frac{(-j)^p}{p!} a_j &= \frac{b_{-1}^{[1]}}{(p-1)!} + \sum_{j=1}^{k-1} \frac{(-j)^{p-1} b_j^{[1]}}{(p-1)!} = \cdots = \frac{b_{-1}^{[N-1]}}{(p-1)!} + \sum_{j=1}^{k-1} \frac{(-j)^{p-1} b_j^{[N-1]}}{(p-1)!} = \sum_{j=1}^{k-1} \frac{(-j)^{p-1} b_j^{[N]}}{(p-1)!}.
\end{aligned}
\tag{5.3}
$$

Theorem 3. *For the k-step $I^{N-1}E$-LMM (5.2), we have:*

I. If $p \le k$, then the $Np+1$ conditions of the system (5.3) are linearly independent. So there exist k-step $I^{N-1}E$ methods of order k.

II. The order of accuracy of a k-step $I^{N-1}E$ method is at most k.

III. The family of k-step $I^{N-1}E$ methods of order k has $(N+k-2)$ free parameters.

Proof. I. Because linear independence for $p = k$ implies linear independence for $p \le k$, it is sufficient to prove the case $p = k$ only.

The system (5.3) can be represented in matrix form

$$\mathbf{W}\mathbf{V} = \mathbf{U},$$

where

$$\mathbf{V} = [a_0, \ldots, a_{k-1}, b_{-1}^{[1]}, b_0^{[1]}, \ldots, b_{k-1}^{[1]}, \ldots, b_{-1}^{[N-1]}, b_0^{[N-1]}, \ldots, b_{k-1}^{[N-1]}, b_0^{[N]}, b_1^{[N]}, \ldots, b_{k-1}^{[N]}]^T \in \mathbb{R}^{(N+1)k+N-1}$$

and

$$\mathbf{U} = [-1, \ldots, -1_{k+1}, 0, \ldots, 0_{(N-1)k}]^T \in \mathbb{R}^{Nk+1},$$

with the coefficient matrix defined as

$$\mathbf{W} = \left[\begin{array}{c:c:ccccccccc} \mathbf{A} & \mathbf{L} & 0 & \dots & 0 & & & & & & \\ \mathbf{H} & & -\mathbf{DA} & \mathbf{0} & \mathbf{0} & & & & & & \\ \hdashline & & & & & & & & & & \\ \mathbf{0} & \mathbf{1} & \mathbf{A} & -\mathbf{1} & -\mathbf{A} & & & & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{A} & -\mathbf{1} & -\mathbf{A} & & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \ddots & \ddots & \ddots & & & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & \mathbf{0} & \mathbf{1} & \mathbf{A} & -\mathbf{1} & -\mathbf{A} & & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{A} & -\mathbf{A} \end{array}\right]_{(Nk+1)\times((N+1)k+N-1)},$$

where $\mathbf{A}$, $\mathbf{D}$, $\mathbf{H}$, and $\mathbf{L}$ are defined in (3.5)–(3.8), respectively, and $\mathbf{1}$ is a vector of ones with size k.

Because the Vandermonde matrix $\mathbf{A}$ is nonsingular, it can be noted that columns $\{1, k+2, \dots, 2k+1, 2k+3, \dots, 4k+2, \dots, (N-2)k+N-1, \dots, (N-1)k+N-2, (N-1)k+N, \dots, (N+1)k+(N-1)\}$ of the matrix $\mathbf{W}$ are linearly independent. Thus, all $(Np+1)$ conditions are linearly independent and the system (5.3) admits an $(N(p-k+1)+(p-2))$-parameter family of solutions provided that $p \le k$. Because columns $k+1, 2k+2, \dots, (N-1)k+(N-1)$ of matrix $\mathbf{W}$ do not enter into the proof, $b_{-1}^{[1]}, b_{-1}^{[2]}, \dots, b_{-1}^{[N-1]}$ can have any nonzero value. This property verifies that the family of methods is I^{N-1}E-LMM.

II. Assume the I^{N-1}E-LMM method has accuracy $\mathcal{O}((\Delta t)^{k+r})$, $r \ge 1$; then the following order conditions hold:

$$\begin{aligned} \sum_{j=-1}^{k-1} b_j^{[N-1]} &= \sum_{j=0}^{k-1} b_j^{[N]}, \\ b_{-1}^{[N-1]} + \sum_{j=1}^{k-1} (-j) b_j^{[N-1]} &= \sum_{j=1}^{k-1} (-j) b_j^{[N]}, \\ &\vdots \\ b_{-1}^{[N-1]} + \sum_{j=1}^{k-1} (-j)^k b_j^{[N-1]} &= \sum_{j=1}^{k-1} (-j)^k b_j^{[N]}. \end{aligned} \tag{5.4}$$

Let $\mu_j = b_j^{[N-1]} - b_j^{[N]}$; then the system (5.4) can be written as

$$\begin{aligned} b_{-1}^{[N-1]} + \sum_{j=0}^{k-1} \mu_j &= 0, \\ b_{-1}^{[N-1]} + \sum_{j=1}^{k-1} (-j)\mu_j &= 0, \\ &\vdots \\ b_{-1}^{[N-1]} + \sum_{j=1}^{k-1} (-j)^k \mu_j &= 0. \end{aligned} \tag{5.5}$$

Writing the system (5.5) in matrix form yields,

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & -1 & \dots & 1-k \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 0 & (-1)^k & \dots & (1-k)^k \end{bmatrix}_{(k+1)\times(k+1)} \begin{bmatrix} b_{-1}^{[N-1]} \\ \mu_0 \\ \vdots \\ \mu_{k-1} \end{bmatrix}_{(k+1)\times 1} = \mathbf{0},$$

implying that $b_{-1}^{[N-1]} = 0$ and $b_j^{[N-1]} = b_j^{[N]}, j = 1, 2, \dots, k-1$, and contradicting the assumption $b_{-1}^{[N-1]} \neq 0$. So the k-step I^{N-1}E-LMM cannot achieve more than order-k accuracy.

III. Consider the k-step I^{N-1}E-LMM that achieves the highest order of accuracy. Then, as shown above, the family of such method has exactly $(N + k - 2)$ free parameters.

□

5.2 Extension to IE^{N-1}

Consider a differential equation of the form (5.1). If an implicit method is applied to $\mathbf{f}^{[1]}(t, \mathbf{y})$, while explicit methods are applied to $\mathbf{f}^{[2]}(t, \mathbf{y}), \mathbf{f}^{[3]}(t, \mathbf{y}), \dots, \mathbf{f}^{[N]}(t, \mathbf{y})$, then the k-step IE^{N-1}-LMM can be formulated as:

$$\frac{1}{\Delta t}\mathbf{y}_{n+1}+\frac{1}{\Delta t}\sum_{j=0}^{k-1}a_j\mathbf{y}_{n-j}=\sum_{j=-1}^{k-1}b_j^{[1]}\mathbf{f}^{[1]}(t_{n-j},\mathbf{y}_{n-j})+\sum_{j=0}^{k-1}b_j^{[2]}\mathbf{f}^{[2]}(t_{n-j},\mathbf{y}_{n-j})+\cdots \\ +\sum_{j=0}^{k-1}b_j^{[N-1]}\mathbf{f}^{[N-1]}(t_{n-j},\mathbf{y}_{n-j})+\sum_{j=0}^{k-1}b_j^{[N]}\mathbf{f}^{[N]}(t_{n-j},\mathbf{y}_{n-j}), \tag{5.6}$$

where we assume that $b_{-1}^{[1]}\neq 0$ and $b_j^{[1]}\neq b_j^{[2]}\neq\ldots\neq b_j^{[N]}$ for some $j\in\{0,1,\ldots,k-1\}$.

Order conditions

The order conditions for k-step IE^{N-1}-LMMs can be derived similarly to analysis of IIE-LMM in section 5.1 by expanding section 5.2 in a Taylor series about t_n and applying eq. (2.2) to the resulting truncation error. An order-p method is obtained provided

$$\begin{aligned}
1+\sum_{j=0}^{k-1}a_j&=0,\\
1-\sum_{j=1}^{k-1}ja_j&=\sum_{j=-1}^{k-1}b_j^{[1]}=\sum_{j=0}^{k-1}b_j^{[2]}=\cdots=\sum_{j=0}^{k-1}b_j^{[N]},\\
\frac{1}{2}+\sum_{j=1}^{k-1}\frac{j^2}{2}a_j&=b_{-1}^{[1]}-\sum_{j=1}^{k-1}jb_j^{[1]}=-\sum_{j=1}^{k-1}jb_j^{[2]}=\cdots=-\sum_{j=1}^{k-1}jb_j^{[N]},\\
&\vdots\\
\frac{1}{p!}+\sum_{j=1}^{k-1}\frac{(-j)^p}{p!}a_j&=\frac{b_{-1}^{[1]}}{(p-1)!}+\sum_{j=1}^{k-1}\frac{(-j)^{p-1}b_j^{[1]}}{(p-1)!}=\sum_{j=1}^{k-1}\frac{(-j)^{p-1}b_j^{[2]}}{(p-1)!}=\cdots=\sum_{j=1}^{k-1}\frac{(-j)^{p-1}b_j^{[N]}}{(p-1)!}.
\end{aligned} \tag{5.7}$$

It is not difficult to generalize Theorem 2 as follows.

Theorem 4. *For the k-step IE^{N-1}-LMM (5.6), we have:*

I. If $p\leq k$, then the $Np+1$ conditions of the system (5.7) are linearly independent. So there exist k-step IE^{N-1}-LMMs of order k.

II. The order of accuracy of a k-step IE^{N-1}-LMM is at most k.

III. The family of k-step IE^{N-1}-LMM of order k has k free parameters.

Similar to the argument in chapter 4, one can generalize the following corollary.

Corollary 2. *Any k-step IE^{N-1}-LMM ($N \geq 3$) of order k reduces to an IMEX-LMM.*

Proof. We use induction. The statement holds for $N = 3$ by Corollary 1. Next, we assume that the statement holds for some $N - 2$. Now, an IE^{N-1}-LMM is equivalent to an IE^{N-2}E-LMM, which by assumption reduces to an IEE-LMM, and consequently by Corollary 1, it reduces to an IMEX-LMM. □

Chapter 6

Conclusions and Future Work

In this work, A comparison study of the performance of VSSBDF splitting methods with physics-based splitting and dynamic linearization on diffusion–reaction-advection models is conducted. In addition, a systematic study for 3-additive splitting methods is implemented, further exploring their properties and efficiency.

6.1 Contributions of this Thesis

The contributions of this thesis can be summarized as follows:

First, a study of 2-additive LMM splitting methods on IVPs representing the discretization of ADR equations is conducted. Specifically, we developed a solver that dynamically adjusts the time-step size by leveraging IMEX SBDF methods, incorporating newly derived expressions for the LTEs for VSSBDF methods of up to fourth order based on step doubling. We evaluate the performance of the IMEX VSSBDF methods by comparing them for two types of additive splitting, physics-based and Jacobian-based splitting. This comparison is conducted across a diverse set of advection-diffusion-reaction models.

In general, the results indicate that the VSSBDF methods utilizing Jacobian splitting outperform those utilizing the physics-based splitting in several experiments in 1D Advection-Diffusion model, 1D Advection-Diffusion-Reaction models, CUSP, and 2D Heat model (for the higher order VSSBDF methods). However, performance is comparable between the both splitting techniques in the combustion model and the 2D Brusselator model, and the performance is improved when using physics-based splitting in the 1D Brusselator model. The results support the Jacobian splitting as a worthy splitting strategy for solving ADR IVPs. Although this approach necessitates computing the Jacobian of the IVP RHSs, it eliminates the need to split the RHS itself. Moreover, calculating the Jacobian does not

always pose a significant challenge, especially when leveraging sparse representation and numerical approximation techniques.

Second, new 3-additive linear multi-step methods are introduced based on two different approaches: two implicit and one explicit discretizations (IIE-LMMs) and one implicit and two explicit discretizations (IEE-LMMs). We systematically derive 3-additive splitting methods up to fourth order and verify their convergence order. A stability analysis is performed on a prototype scalar linear DRA equation, showing the new methods give satisfactory stability results and in some cases are A- or A_0-stable. In addition, the stability regions of the new 3-additive methods are compared with some popular IMEX methods, and it is shown that the new methods have larger stability regions in some cases. Results on maximal attainable order for a given number of steps are provided, along with corresponding generalizations to N-additive splitting methods. In particular, we show that the class of k-step, order-k (N-additive) IE^{N-1}-LMMs reduce to the class of (2-additive) IMEX methods. The performance of the 3-additive methods is tested by comparing them with some popular IMEX methods on a stiff variation of the standard Brusselator problem. The experiments show 3-additive splitting methods can outperform comparable IMEX methods through the combination of improved stability, improved computational expense per step, and no significant degradation in accuracy, the results of this part have been published in [43].

6.2 Future Work

Future research could expand upon the current study by extensively testing the efficacy of both 2- and 3-additive splitting methods across a broader range of ADR models. Additionally, the performance of variable stepsize algorithms warrants further investigation, particularly examining how varying algorithm parameters impact outcomes. There is also significant potential in developing more 3-additive splitting methods, specifically tailored to optimize stability and enhance numerical performance.

Appendix A

Derivation of the Local Truncation Error Formula for IMEX SBDF Methods

In this appendix, the error approximation and extrapolation formula for IMEX-SBDF methods used in Subsection 2.2 are derived. Derivations are shown up to order four.

A.0.1 VSSBDF1

Consider the ODE

$$\frac{dy}{dt} = f(t, y) + g(t, y). \tag{A.1}$$

Substituting $y_{n+1} = y(t + \Delta t_{n+1})$ into the VSSBDF1 scheme and apply a multi-variable Taylor expansion around time t, and employing $y'(t) = f(y(t)) + g(y(t))$, we obtain the LTE as follows:

$$LTE = (y'(t) - f(y(t)) - g(y(t)))\,\Delta t_{n+1} + \left(\frac{1}{2}y''(t) + g'(y(t))\,y'(t))\right)\Delta t_{n+1}^2 \ + O(\Delta t_{n+1}^3). \tag{A.2}$$

For a solution $y(t)$ of the ODE that is twice differentiable, the linear term in Δt_{n+1} of the LTE vanishes, simplifying the expression to:

$$LTE = \left(\frac{1}{2}y''(t) + g'(y(t))\,y'(t))\right)\Delta t_{n+1}^2 \ + O(\Delta t_{n+1}^3) \tag{A.3}$$

Accordingly, the LTE for the coarse step can be given as

$$\epsilon^c = y_{n+1}^c - y(t_{n+1}) \approx C\Delta t_{n+1}^2. \tag{A.4}$$

To obtain the LTE for y_{n+1}^f, we first calculate the LTE for $y_{n+\frac{1}{2}}^f$ by replacing Δt_{n+1} in Equation (A.4) by $\frac{\Delta t_{n+1}}{2}$. Subsequently, determining the LTE for y_{n+1}^f requires adjusting Δt_{n+1} to $\frac{\Delta t_{n+1}}{2}$. The aggregation of these errors leads to the LTE for y_{n+1}^f:

$$\epsilon^f = y_{n+1}^f - y(t_{n+1}) \approx \frac{1}{2}\,C\Delta t_{n+1}^2. \tag{A.5}$$

Using Equations (A.4) and (A.5), the constant C can be approximated using of y^c, y^f, Δt_{n+1}, and Δt_n, which is then used in equation (A.4) to deduce

$$\epsilon^c \approx \ 2\left(y^c - y^f\right).$$

Finally, we subtract ϵ^c from y_{n+1}^c to create a more accurate approximation of $y(t_{n+1})$. This

results in the extrapolation formula $y(t_{n+1}) \approx y^c_{n+1} - \epsilon^c$.

A.0.2 VSSBDF2

Making the substitutions $y_{n+1} = y(t + \Delta t_{n+1})$ and $y_{n-1} = y(t - \Delta t_n)$ and performing a multivariable Taylor expansion about t, the local truncation error is found to take the form

$$\begin{aligned} LTE = &\left(y'(t) - f(y(t)) - g(y(t))\right)\Delta t_{n+1} + \left(y''(t) - f'(y(t))\,y'(t) - g'(y(t))\,y'(t)\right)\Delta t^2_{n+1} \\ &+ \left(\frac{1}{2}f''(y(t))\,y'(t)^2 + \frac{1}{2}f'(y(t))\,y''(t) - \frac{1}{6}y^{(3)}(t)\right)\Delta t_n\,\Delta t^2_{n+1} \\ &+ \left(-\frac{1}{2}g''(y(t))\,y'(t)^2 - \frac{1}{2}g'(y(t))\,y''(t) + \frac{1}{3}y^{(3)}(t)\right)\Delta t^3_{n+1} + O(\Delta t^4), \end{aligned} \tag{A.6}$$

where $O(\Delta t^4)$ denotes terms where Δt_{n+1} and Δt_n combined appear four or more times. If $y(t)$ is a three-time differentiable solution of the ODE, the Δt_{n+1} and Δt^2_{n+1} terms vanish and the cubic term can be simplified:

$$\begin{aligned} LTE &= \left(\frac{1}{2}f''(y(t))\,y'(t)^2 + \frac{1}{2}f'(y(t))\,y''(t) - \frac{1}{6}y^{(3)}(t)\right)\Delta t_n\Delta t^2_{n+1} \\ &\quad + \left(-\frac{1}{2}g''(y(t))\,y'(t)^2 - \frac{1}{2}g'(y(t))y''(t) + \frac{1}{3}y^{(3)}(t)\right)\Delta t^3_{n+1} + O(\Delta t^4) \qquad \text{(A.7)} \\ &= \left(\frac{1}{3}y^{(3)}(t) - \frac{1}{2}g''(y(t))\,y'(t)^2 - \frac{1}{2}g'(y(t))\,y''(t)\right)\Delta t^2_{n+1}\left(\Delta t_{n+1} + \Delta t_n\right) + O(\Delta t^4). \end{aligned} \tag{A.8}$$

Thus if $y^{(3)}(t)$ is bounded, the local truncation error for the coarse step is

$$\epsilon^c = y^c_{n+1} - y(t_{n+1}) \approx C\Delta t^2_{n+1}\left(\Delta t_{n+1} + \Delta t_n\right). \tag{A.9}$$

To obtain the LTE for y^f_{n+1}, we first calculate the LTE for $y^f_{n+\frac{1}{2}}$ by replacing Δt_{n+1} and Δt_n in (A.9) by $\frac{\Delta t_{n+1}}{2}$ and $\frac{\Delta t_n}{2}$, respectively. We then find the local truncation error for y^f_{n+1} by setting both Δt_{n+1} and Δt_n to $\frac{\Delta t_{n+1}}{2}$. Adding these errors yields to the LTE for y^f_{n+1},

$$\epsilon^f = y^f_{n+1} - y(t^{n+1}) \approx \frac{C\Delta t^2_{n+1}\left(\Delta t_{n+1} + \Delta t_n\right)}{8} + C\,\frac{\Delta t^3_{n+1}}{4}. \tag{A.10}$$

Using equations (A.9) and (A.10), the constant C can be approximated using y^c, y^f, Δt_{n+1}, and Δt_n in (A.9) to yield

$$\epsilon^c \approx \frac{y^c - y^f}{\frac{5}{8}\Delta t_{n+1} + \frac{7}{8}\Delta t_n}\;\left(\Delta t_{n+1} + \Delta t_n\right). \tag{A.11}$$

Finally, we subtract ϵ^c from y^c_{n+1} to create a more accurate approximation of $y(t^{n+1})$.

A.0.3 VSSBDF3

Substituting $y_{n+1} = y(t + \Delta t_{n+1})$, $y_{n-1} = y(t - \Delta t_n)$, and $y_{n-2} = y(t - (\Delta t_n + \Delta t_{n-1}))$ in VSSBDF3 method and perform multi variable Taylor expansion about t, and employing $y'(t) = f(y(t)) + g(y(t))$, the LTE can be written as:

$$\begin{aligned} LTE = &\left(y'(t) - f(y(t)) - g(y(t))\right)\Delta t_{n+1} + \left(y''(t) - f'(y(t))\,y'(t) - g'(y(t))\,y'(t)\right)\Delta t_{n+1}^2 \\ &+ \frac{1}{2}\left(y^{(3)}(t) - f''(y(t))\,y'(t)^2 - f'(y(t))\,y''(t) - g''(u(t))\,y'(t)^2 - g'(y(t))\,y''(t)\right)\Delta t_{n+1}^3 \\ &- \left(24y^{(4)}(t) - \frac{1}{6}f^{(3)}(y(t))\,y'(t)^3 - \frac{1}{2}f''(y(t))\,y'(t)\,y''(t) + \frac{1}{6}f'(y(t))y^{(3)}(t)\right) \\ &\Delta t_{n+1}^2(\Delta t_{n+1} + \Delta t_n)(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1}) + O(\Delta t^5), \end{aligned}$$

where $O(\Delta t^5)$ denotes terms where Δt_{n+1}, Δt_n and Δt_{n-1} appear in combination five or more times. Given that the solution $y(t)$ is a four-time differentiable, the terms involving Δt_{n+1}, Δt_{n+1}^2, and Δt_{n+1}^3 in the LTE disappear, and the quartic terms cab be written as:

$$\begin{aligned} LTE = &-\left(24y^{(4)}(t) - \frac{1}{6}f^{(3)}(y(t))\,y'(t)^3 - \frac{1}{2}f''(y(t))\,y'(t)\,y''(t) + \frac{1}{6}f'(y(t))y^{(3)}(t)\right) \\ &\Delta t_{n+1}^2(\Delta t_{n+1} + \Delta t_n)(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1}) + O(\Delta t^5). \end{aligned}$$

Thus if $y^{(4)}(t)$ is bounded, then the LTE for the coarse step can be given as

$$\epsilon^c = y_{n+1}^c - y(t_{n+1}) \approx C\Delta t_{n+1}^2(\Delta t_{n+1} + \Delta t_n)(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1}). \tag{A.12}$$

To obtain the LTE for y_{n+1}^f, we first calculate the LTE for $y_{n+\frac{1}{2}}^f$ by replacing Δt_{n+1}, Δt_n, and Δt_{n-1} in equation (A.12) by $\frac{\Delta t_{n+1}}{2}$ $\frac{\Delta t_n}{2}$, and $\frac{\Delta t_n}{2}$, respectively. Subsequently, determining the LTE for y_{n+1}^f by adjusting both Δt_{n+1} and Δt_n to $\frac{\Delta t_{n+1}}{2}$, and setting Δt_{n-1} to $\frac{\Delta t_n}{2}$. Adding these errors results the LTE for y_{n+1}^f:

$$\begin{aligned} \epsilon^f = y_{n+1}^f - y(t_{n+1}) \approx &\frac{1}{4}C\Delta t_{n+1}^2\left(\frac{1}{2}\Delta t_{n+1} + \frac{1}{2}\Delta t_n\right)\left(\frac{1}{2}\Delta t_{n+1} + \Delta t_n\right) \\ &+ \frac{1}{4}C\Delta t_{n+1}^3\left(\Delta t_{n+1} + \frac{1}{2}\Delta t_n\right). \end{aligned} \tag{A.13}$$

Using equations (A.12) and (A.13), the constant C can be approximated using y^c, y^f, Δt_{n+1}, Δt_n, and Δt_{n-1}, which is then used in equation (A.12) to deduce

$$\epsilon^c \approx \frac{16(y^c - y^f)}{\left(14\,\Delta t_n^2 + 16\,\Delta t_n\Delta t_{n-1} + 27\,\Delta t_n\Delta t_{n+1} + 16\,\Delta t_{n-1}\Delta t_{n+1} + 11\,\Delta t_{n+1}^2\right)}\Delta T_3,$$

where $\Delta T_3 = (\Delta t_n + \Delta t_{n+1})(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1})$.

Lastly, we subtract ϵ^c from y_{n+1}^c to create a more accurate approximation of $y(t_{n+1})$.

A.0.4 VSSBDF4

Similar to the previous analysis, one can find the LTE for the VSSBDF4 method as follows:

$$
\begin{aligned}
LTE = &\left(y'(t) - f(y(t)) - g(y(t))\right)\Delta t_{n+1} + \left(y''(t) - f'(y(t))\,y'(t) - g'(y(t))\,y'(t)\right)\Delta t_{n+1}^2 \\
&+ \frac{1}{2}\left(y^{(3)}(t) - f''(y(t))\,y'(t)^2 - f'(y(t))\,y''(t) - g''(u(t))\,y'(t)^2 - g'(y(t))\,y''(t)\right)\Delta t_{n+1}^3 \\
&+ \frac{1}{6}(y^{(4)}(t) - f^{(3)}(y(t))\cdot(y'(t))^3 - 3f''(y(t))\cdot y'(t)\cdot y''(t) - g^{(3)}(y(t))\cdot(y'(t))^3 \\
&- 3g''(y(t))\cdot y'(t)\cdot y''(t) - (f''(y(t)) - g''(y(t)))\cdot y^{(3)}(t) - (f'(y(t)) - g'(y(t)))\cdot y^{(4)}(t))\Delta t_{n+1}^4 \\
&- \frac{1}{6}\left(\frac{1}{20}y^{(5)}(t) - \frac{1}{4}f^{(4)}(y(t))\,y'(t)^4 - \frac{3}{2}f^{(3)}(y(t))\,y'(t)^2\,y''(t) + \frac{1}{4}f'(y(t))y^{(4)}(t)\right. \\
&\left.-y^{(3)}(t)y'(t)f''(y(t)) + \frac{3}{4}y''(t)^2 f''(y(t))\right)\Delta t_{n+1}^2(\Delta t_{n+1} + \Delta t_n)(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1}) \\
&(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1} + \Delta t_{n-2}) + O(\Delta t^6),
\end{aligned}
$$

where $O(\Delta t^6)$ denotes terms where Δt_{n+1}, Δt_n, Δt_{n-1} and Δt_{n-2} in combination six or more times. Given the solution $y(t)$ is a five-time differentiable bounded solution of the ODE, then the Δt_{n+1}, Δt_{n+1}^2 and Δt_{n+1}^3 terms in the LTE disappear, and the quintic term can be rewritten as:

$$
\begin{aligned}
LTE = &-\frac{1}{6}\left(\frac{1}{20}y^{(5)}(t) - \frac{1}{4}f^{(4)}(y(t))\,y'(t)^4 - \frac{3}{2}f^{(3)}(y(t))\,y'(t)^2\,y''(t) + \frac{1}{4}f'(y(t))y^{(4)}(t)\right. \\
&\left.-y^{(3)}(t)y'(t)f''(y(t)) + \frac{3}{4}y''(t)^2 f''(y(t))\right) + \Delta t_{n+1}^2(\Delta t_{n+1} + \Delta t_n)(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1}) \\
&(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1} + \Delta t_{n-2}) + O(\Delta t^6)
\end{aligned}
\tag{A.14}
$$

Thus if $y^{(5)}(t)$ is bounded, the LTE for the coarse step can be given as

$$
\begin{aligned}
\epsilon^c &= y_{n+1}^c - y(t_{n+1}) \\
&\approx C\Delta t_{n+1}^2(\Delta t_{n+1} + \Delta t_n)(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1})(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1} + \Delta t_{n-2}) \\
&\qquad + O(\Delta t^6).
\end{aligned}
\tag{A.15}
$$

To obtain the LTE for y_{n+1}^f, we first calculate the LTE for $y_{n+\frac{1}{2}}^f$ by replacing Δt_{n+1}, Δt_n, Δt_{n-1}, and Δt_{n-2} in equation (A.15) by $\frac{\Delta t_{n+1}}{2}$, $\frac{\Delta t_n}{2}$, $\frac{\Delta t_n}{2}$, and $\frac{\Delta t_{n-1}}{2}$ respectively. After that, we obtain the LTE for y_{n+1}^f by adjusting both Δt_{n+1} and Δt_n to $\frac{\Delta t_{n+1}}{2}$ and Δt_{n-1}, and Δt_{n-2} to $\frac{\Delta t_n}{2}$. Adding these errors results the LTE for y_{n+1}^f,

$$
\begin{aligned}
\epsilon^f &= y_{n+1}^f - y(t_{n+1}) \\
&\approx \frac{1}{4}\,C\Delta t_{n+1}^2\left(\frac{1}{2}\,\Delta t_{n+1} + \frac{1}{2}\,\Delta t_n\right)\left(\frac{1}{2}\,\Delta t_{n+1} + \Delta t_n\right) \\
&\left(\frac{1}{2}\,\Delta t_{n+1} + \Delta t_n + \frac{1}{2}\Delta t_{n-1}\right) + \frac{1}{4}C\Delta t_{n+1}^3\left(\Delta t_{n+1} + \frac{1}{2}\Delta t_n\right)(\Delta t_{n+1} + \Delta t_n)\,.
\end{aligned}
$$

Using equations (A.15) and (A.16), the constant C can be approximated using y^c, y^f,

and $\Delta t_{n-2}, \Delta t_{n-1}, \Delta t_n$, Δt_{n+1}, which is then used in (A.15) to deduce

$$\epsilon^c \approx \frac{32(y^c - y^f)}{C_1 + C_2} \Delta T_4,$$

where

$$\Delta T_4 = (\Delta t_{n+1} + \Delta t_n)(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1})(\Delta t_{n+1} + \Delta t_n + \Delta t_{n-1} + \Delta t_{n-2}),$$

$$\begin{aligned} C_1 =& 28\,\Delta t_n^3 + 32\,\Delta t_n^2 \Delta t_{n-2} + 62\,\Delta t_n^2 \Delta t_{n-1} + 84\,\Delta t_n^2 \Delta t_{n+1} + \\ & 32\,\Delta t_n \Delta t_{n-2} \Delta t_{n-1} + 64\,\Delta t_n \Delta t_{n-2} \Delta t_{n+1} + 32\,\Delta t_n \Delta t_{n-1}^2 \end{aligned} \tag{A.16}$$

and

$$\begin{aligned} C_2 =& 125\,\Delta t_n \Delta t_{n-1} \Delta t_{n+1} + 79\,\Delta t_n \Delta t^2{}_{n+1} + 32\,\Delta t_{n-2} \Delta t_{n-1} \Delta t_{n+1} + \\ & 32\,\Delta t_{n-2} \Delta t_{n+1}^2 + 32\,\Delta t_{n-1}^2 \Delta t_{n+1} + 63\,\Delta t_{n-1} \Delta t_{n+1}^2 + 23\,\Delta t_{n+1}^3, \end{aligned} \tag{A.17}$$

Finally, we subtract ϵ^c from y_{n+1}^c to create a more accurate approximation of $y(t_{n+1})$.

References

[1] Solve systems of linear equations ax = b for x - matlab mldivide.

[2] G. Akrivis. Implicit–explicit multistep methods for nonlinear parabolic equations. *Mathematics of Computation*, 2012.

[3] G. Akrivis. Stability of implicit and implicit-explicit multistep methods for nonlinear parabolic equations. *IMA Journal of Numerical Analysis*, 2018.

[4] G. Akrivis, M. Crouzeix, and C. Makridakis. Implicit-explicit multistep finite element methods for nonlinear parabolic problems. *Mathematics of Computation*, 1998.

[5] G. Albi, G. Dimarco, and L. Pareschi. Implicit-explicit multistep methods for hyperbolic systems with multiscale relaxation, 2019.

[6] U. M. Ascher and L. R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM: Society for Industrial and Applied Mathematics, 1998.

[7] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. 25:151–167, 1997.

[8] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.

[9] J. G. Berryman and C. J. Holland. Nonlinear diffusion problem arising in plasma physics. *Physical Review Letters*, 1978.

[10] M. Bertsch. Asymptotic behavior of solutions of a nonlinear diffusion equation . *SIAM Journal on Applied Mathematics*, 1982.

[11] F. Bianco, S. Chibbaro, and R. Prud. Etude d'une ´equation de convection-r´eaction-diffusion en ´ecoulement compressible. 2011.

[12] F. Bianco, S. Chibbaro, and R. Prud'homme. Étude d'une équation de convection-réaction-diffusion en écoulement compressible. Presented at: Rencontre du non linéaire, March 2011.

[13] L. S. Blackford, A. Petitet, R. Pozo, K. Remington, R. C. Whaley, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, et al. An updated set of basic linear algebra subprograms (blas). *ACM Transactions on Mathematical Software*, 28(2):135–151, 2002.

[14] J. Blair, I. A. Frigaard, H. Kunze, R. Makarov, R. Melnik, and R. J. Spiteri. *Mathematical and Computational Approaches in Advancing Modern Science and Engineering*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[15] T. Bodnár and A. Sequeira. Numerical simulation of the coagulation dynamics of blood. *Computational and Mathematical Methods in Medicine*, 2008.

[16] A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang. Extrapolated Implicit-Explicit Runge-Kutta Methods. *Mathematical Modelling and Analysis*, 19(1):18–43, Jan 2014.

[17] A. Cardone, Z. Jackiewicz, A. Sandu, and H. Zhang. Extrapolation-based implicit-explicit general linear methods. *Numerical Algorithms*, 2014.

[18] R. Chinomona and D. R. Reynolds. Implicit-explicit multirate infinitesimal GARK methods, 2021.

[19] R. H. Clayton. Computational models of normal and abnormal action potential propagation in cardiac tissue: Linking experimental and clinical cardiology, 2001.

[20] M. Crouzeix. Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques. *Numerische Mathematik*, 1980.

[21] G. Dimarco and L. Pareschi. Implicit-explicit linear multistep methods for stiff kinetic equations. *SIAM Journal on Numerical Analysis*, 55, 02 2016.

[22] H. Do, A. A. Owida, W. Yang, and Y. S. Morsi. Numerical simulation of the haemodynamics in end-to-side anastomoses. *International Journal for Numerical Methods in Fluids*, 2011.

[23] E. Griepentrog. Gemischte Runge-Kutta-verfahren fur steife systeme. *Seminarbereicht Sekt. Math.*, 11:19–29, 1978.

[24] T. Echekki and E. Mastorakos. *Turbulent Combustion Modeling*, volume 95 of *Fluid Mechanics and Its Applications*. Springer Netherlands, 2011.

[25] N. El Khatib, S. Génieys, and V. Volpert. Atherosclerosis initiation modeled as an inflammatory process. *Mathematical Modelling of Natural Phenomena*, 2007.

[26] J. Frank, W. Hundsdorfer, and J. G. Verwer. On the stability of implicit-explicit linear multistep methods. *Applied Numerical Mathematics*, 1997.

[27] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[28] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2008.

[29] E. Hairer and G. Wanner. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, volume 14 of *Springer series in computational mathematics*. Springer-Verlag, Berlin, 1996.

[30] S. L. Heston. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *Review of Financial Studies*, 1993.

[31] A. Hidalgo, L. Tello, and E. F. Toro. Numerical and analytical study of an atherosclerosis inflammatory disease model. *Journal of Mathematical Biology*, 2014.

[32] E. Hofer. A Partially Implicit Method for Large Stiff Systems of ODE's with Only Few Equations Introducing Small Time-Constants. *SIAM Journal on Numerical Analysis*, 1976.

[33] F. Huang and J. Shen. A new class of implicit–explicit bdfk sav schemes for general dissipative systems and their error analysis. *Computer Methods in Applied Mechanics and Engineering*, 392:114718, 2022.

[34] W. Hundsdorfer and S. J. Ruuth. IMEX extensions of linear multistep methods with general monotonicity and boundedness properties. *Journal of Computational Physics*, 2007.

[35] W. Hundsdorfer and J. G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Number 33 in Springer Series in Computational Mathematics. Springer-Verlag, Berlin, 2003.

[36] L. M. Jiji. *Heat Convection*. Springer Berlin Heidelberg, 2nd edition, 2009.

[37] C. A. Kennedy and M. H. Carpenter. Additive Runge–Kutta schemes for convection-diffusion-reaction equations. Technical report, "NASA", 2001.

[38] C. A. Kennedy and M. H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Applied Numerical Mathematics*, 2003.

[39] H. Kotakemori, H. Hasegawa, T. Kajiyama, A. Nukada, R. Suda, and A. Nishida. Performance evaluation of parallel sparse matrix–vector products on sgi altix3700. In M. S. Mueller, B. M. Chapman, B. R. de Supinski, A. D. Malony, and M. Voss, editors, *OpenMP Shared Memory Parallel Programming*, pages 153–163, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[40] R. Lefever and G. Nicolis. Chemical instabilities and sustained oscillations. *Journal of Theoretical Biology*, 30(2):267–284, 1971.

[41] R. Lefever and G. Nicolis. Chemical instabilities and sustained oscillations. *Journal of Theoretical Biology*, 30(2):267–284, 1971.

[42] K. C. Loy and Y. Bourgault. Robust high-order semi-implicit backward difference formulas for navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 90(4):195–216, 2019.

[43] R. A. Mara'Beh, R. J. Spiteri, P. González, and J. M. Mantas. 3-additive linear multistep methods for diffusion-reaction-advection models. *Applied Numerical Mathematics*, 183:15–38, 2023.

[44] G. J. McRae, W. R. Goodin, and J. H. Seinfeld. Numerical solution of the atmospheric diffusion equation for chemically reacting flows, 1982.

[45] R. Natalini. Convergence to equilibrium for the relaxation approximations of conservation laws. *Communications on Pure and Applied Mathematics*, 1996.

[46] A. Nishida. Experience in developing an open source scalable software infrastructure in japan. In D. Taniar, O. Gervasi, B. Murgante, E. Pardede, and B. O. Apduhan, editors, *Computational Science and Its Applications – ICCSA 2010*, pages 448–462, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[47] L. Pareschi and G. Russo. Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. In *Journal of Scientific Computing*, 2005.

[48] R. I. A. Patterson and W. Wagner. A Stochastic Weighted Particle Method for Coagulation–Advection Problems. *SIAM Journal on Scientific Computing*, 2012.

[49] A. Preuss. A study of 2-additive splitting for solving advection-diffusion-reaction equations. Master's thesis, University of Saskatchewan, Canada, 2014. Master degree in Computer Science.

[50] A. Preuss, J. Lipoth, and R. J. Spiteri. When and how to split? a comparison of two imex splitting techniques for solving advection–diffusion–reaction equations. *Journal of Computational and Applied Mathematics*, 414:114418, 2022.

[51] S. J. Ruuth. Implicit-explicit methods for reaction-diffusion problems in pattern formation. *Journal of Mathematical Biology*, 1995.

[52] Y. Saad. A flexible inner-outer preconditioned gmres algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[53] Y. Saad. Ilut: A dual threshold incomplete lu factorization. *Numerical Linear Algebra with Applications*, 1(4):387–402, 1994.

[54] S. Scarle. Implications of the Turing completeness of reaction-diffusion models, informed by GPGPU simulations on an XBox 360: Cardiac arrhythmias, re-entry and the Halting problem. *Computational Biology and Chemistry*, 2009.

[55] L. F. Shampine, I. Gladwell, and S. Thompson. *Solving ODEs with MATLAB*. Cambridge University Press, 2003.

[56] S. Sherwin, J. Peiro, O. Shah, G.-S. Karamanos, and D. Doorly. Computational haemodynamics: geometry and non-newtonian modelling using spectral/hp element methods. *Computing and Visualization in Science*, 3:77–83, 05 2000.

[57] R. J. Spiteri and R. C. Dean. On the performance of implicit-explicit Runge–Kutta methods in models of cardiac electrical activity. *IEEE Transactions on Biomedical Engineering*, 55(5):1488–1495, May 2008.

[58] D. Wang and S. J. Ruuth. Variable step-size implicit-explicit linear multistep methods for time-dependent partial differential equations. *Journal of Computational Mathematics*, 2008.

[59] D. Wang and S. J. Ruuth. Variable step-size implicit-explicit linear multistep methods for time-dependent partial differential equations. *Journal of Computational Mathematics*, 26(6):838–855, 2008.

[60] W. Wang, Y. Chen, and H. Fang. On the variable two-step imex bdf method for parabolic integro-differential equations with nonsmooth initial data arising in finance. *SIAM Journal on Numerical Analysis*, 57(3):1289–1317, 2019.

[61] W. Wang, M. Mao, and Y. Huang. A posteriori error control and adaptivity for the imex bdf2 method for pides with application to options pricing models. *Journal of Scientific Computing*, 93:55, 2022.

[62] D. Yan, M. Pugh, and F. Dawson. Adaptive time-stepping schemes for the solution of the poisson-nernst-planck equations. *Applied Numerical Mathematics*, 163:254–269, 2021.

[63] N. Yousefzadeh, G. Hojjati, and A. Abdi. Construction of implicit–explicit second-derivative bdf methods. *Bulletin of the Iranian Mathematical Society*, 44:991–1006, 2018.

[64] H. Zhang and A. Sandu. A second-order diagonally-implicit-explicit multi-stage integration method. In *Procedia Computer Science*, 2012.

[65] H. Zhang, A. Sandu, and S. Blaise. Partitioned and implicit-explicit general linear methods for ordinary differential equations. *Journal of Scientific Computing*, 2014.

[66] Z. Zlatev. Numerical Treatment of Large Air Pollution Models. 1995.