Numerical Evolution Methods of Rational Form for Reaction Diffusion Equations

Said Algarni

Thesis Submitted to the Faculty of Graduate and Postdoctoral Studies
In partial fulfilment of the requirements for the degree of Doctor of Philosophy in Mathematics [1]

School of Mathematics and Statistics
Faculty of Science
Carleton University

[1] The Ph.D. program is a joint program with University of Ottawa, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-89323-4

Our file Notre référence
ISBN: 978-0-494-89323-4

Abstract

The purpose of this study was to investigate select numerical methods that demonstrate good performance in solving PDEs that couple diffusion and reaction terms. These types of equations have numerous fields of application such as environmental studies, biology, chemistry, medicine, and ecology. Our aim was to investigate and develop accurate and efficient approaches which compare favourably to other applicable methods. In particular, we investigated and adapted a relatively new class of methods based on rational polynomials. Namely, Padé time stepping (PTS), which is highly stable for the purposes of the present application and is associated with lower computational costs. Furthermore, PTS was optimized for our study to focus on reaction diffusion equations. Due to the rational form of PTS method, a local error control threshold (LECT) was proposed. Numerical runs were conducted to obtain the optimal LECT. In addition, new schemes based on both PTS and splitting methods were established.

Based on the results, we found PTS alone and combined via splitting with other approaches provided favourable performance in certain and wide ranging parameter regimes.

Acknowledgements

I would like to thank all of those who helped make this dissertation possible. In particular, my mom for standing beside me throughout my research and helping me writing up this thesis. She has been my inspiration and motivation for continuing in my studies. I also wish to thank my wife for her personal support and great patience at all times. Last but not least, I thank my supervisor, Dr. David Amundsen, for all his guidance, encouragement, and support.

Dedication

I dedicate this dissertation to my wonderful family. Particularly to my parents, my wife, my siblings, my children.

List of Symbols

Symbols	Definition
$\triangle$	Laplace operator
α	Diffusion coefficient
β_s	The large number in magnitude of real axis of stability boundary, where s is the number of stages
ϵ	Local error control threshold
ϵ^*	Optimum value for local error control threshold

List of Acronyms

Acronyms	Definition
FDM	Finite Difference Method
FEM	Finite Element Method
SM	Spectral Method
CFL	Courant–Friedrichs–Lewy
PTS	Padé Time Stepping
LECT	Local Error Control Threshold
$\text{PTS}[1/1]_{\text{Taylor}}$	PTS[1/1] with Taylor as a threshold control
$\text{PTS}[1/1]_{[0/2]}$	PTS[1/1] with PTS[0/2] as a threshold control
$\text{PTS}[1/1]_{\frac{h}{s}}$	PTS[1/1] with $\text{PTS}[1/1]_{\frac{h}{s}}$ as a threshold control
RK2-PTSVS	Runge–Kutta Method of order two with PTS[1/1] of variable step-size
RKC2-PTSVS	Runge–Kutta–Chebyshev Method of order two with PTS[1/1] of variable step-size

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

In this work, we discuss numerical methods for solving partial differential equations coupling diffusion and reaction terms. The simple form of a reaction diffusion equation is the following

$$u_t(t,x) = \alpha\, u_{xx}(t,x) + f(u), \tag{1.0.1}$$

where u is an order-parameter field, e.g., population density, chemical concentration, magnetization, which depends on space x and time t. The order-parameter may be either scalar or vector, depending on the number of variables that describe the physical system. The order-parameter evolves in time due to a local reaction, described by the nonlinear term $f(u)$, in conjunction with spatial diffusion. The coefficient α can be a diagonal matrix or in some cases a full matrix to account for so-called cross-diffusion terms. In most cases, however, α can be a scalar where the amount of diffusion is the same in all coordinate directions, or it could be dependent on time and space $\alpha(t,x)$. These reaction diffusion equations appear in many types of real life problems.

In this chapter, we provide a motivation, basic definitions, and space-time discretizations. Then, time integration methods for ODEs are illustrated. In the ensuing

chapters, splitting methods, stabilized Runge–Kutta methods, and schemes that solve the linear part exactly (ELP) will be discussed. Finally, numerical examples are presented.

According to the relevant literature, other numerical methods that solve reaction diffusion equations exist. One of these methods is the spectral method (SM), which is good for wave problems. Precisely, it works well for constant coefficient, simple boundary conditions, and regular domains. Another method is finite the element method (FEM), an appropriate method to handle complicated geometries, which is not the case with our study. In general, FEM is less easy to code, though some packages are available. The finite difference method (FDM), which is simple and easy to implement, is another way to solve numerically PDEs. It is also good for simple domains. Often, FDM is the fastest way to get a feel for a problem. For useful references in this field see, [34, 41, 46, 47, 75, 80, 81].

For an introduction to fast solvers for large systems of ODEs, see [63, 74]. Moreover, if the underlying system of mathematical equations is singular, then the numerical method must be chosen or designed to deal with the singularities, see [40]. However, since singularities will not arise for reaction diffusion equations in general the numerical approaches used here are exclusively FDMs.

1.0.1 Motivation

Applications of the reaction diffusion equations are numerous. They have been involved and used to simulate a variety of different phenomena, from environmental studies [54, 55], biology [23, 37, 54, 55, 51], chemistry [54, 55, 44, 62, 66, 82, 79], medicine [33, 50, 54, 55, 51], ecology [48, 61, 54, 55], and epidemiology [16, 17]. Many reaction diffusion equations that have been studied in the literature differ in their reaction terms. Such examples are the Fisher model in population genetics [58, 59, 91], the Kolmogorov–Petrovskii–Piskunov (KPP) planar model for the evolu-

tion of advantageous genes [42, 91], Turing's continuous model of morphogenesis [82], Maginu model of morphogenesis [49], FitzHugh's model (Nagumo) of the propagation of voltage impulse through a nerve axon [3], Brusselator's model for the Belousov-Zhabotinsky reaction (BZR) [28, 83], bistable transmission lines [56], wave system in plane Poiseuille flow [71], Gierer Meinhardt [53], generating waves by wind [10] and also in the modelling of fire spread [29].

A simple example for reaction diffusion equation is the Gray–Scott model which involves two chemical species, with concentrations u and v. The equations are

$$\begin{aligned} u_t &= D_1 \triangle u - uv^2 + \gamma(1-u), \\ v_t &= D_2 \triangle v + uv^2 - (\gamma + k)v, \end{aligned} \tag{1.0.2}$$

where D_1, D_2, γ and k are constants. The following example of (1.0.2) was described in [60]. Two spatial dimensions were considered with $0 \leq x, y \leq 2.5$ and a homogeneous Neumann condition was imposed to both u and v. The initial value is

$$\begin{aligned} u(x,y,0) &= 1 - 2v(x,y,0), \\ v(x,y,0) &= \left\{ \begin{array}{ll} \frac{1}{4}\sin^2(4\pi x)\sin^2(4\pi y), & \text{if } 1 \leq x, y \leq 1.5, \\ 0, & \text{elsewhere.} \end{array} \right\} \end{aligned} \tag{1.0.3}$$

Also, the parameters were taken as $D_1 = 8 \times 10^{-5}$, $D_2 = 4 \times 10^{-5}$, $\gamma = 0.024$ and $k = 0.06$. We noticed here that D_1 and D_2 are small, which will lead to a stiff problem (it will be defined in the preliminaries section). Figure 1.1, copied from [39], shows the evolution of the v-component by contour lines in the x, y-plane at different times. So, as t gets larger the number of spots will increase and the size will change. These modifications are basically due to diffusion and reaction terms.

1.0.2 Preliminaries

First, to study numerically these PDEs, we need to convert the PDE to ODEs using spatial discretization. Thus, we need some basic definitions for numerical methods of

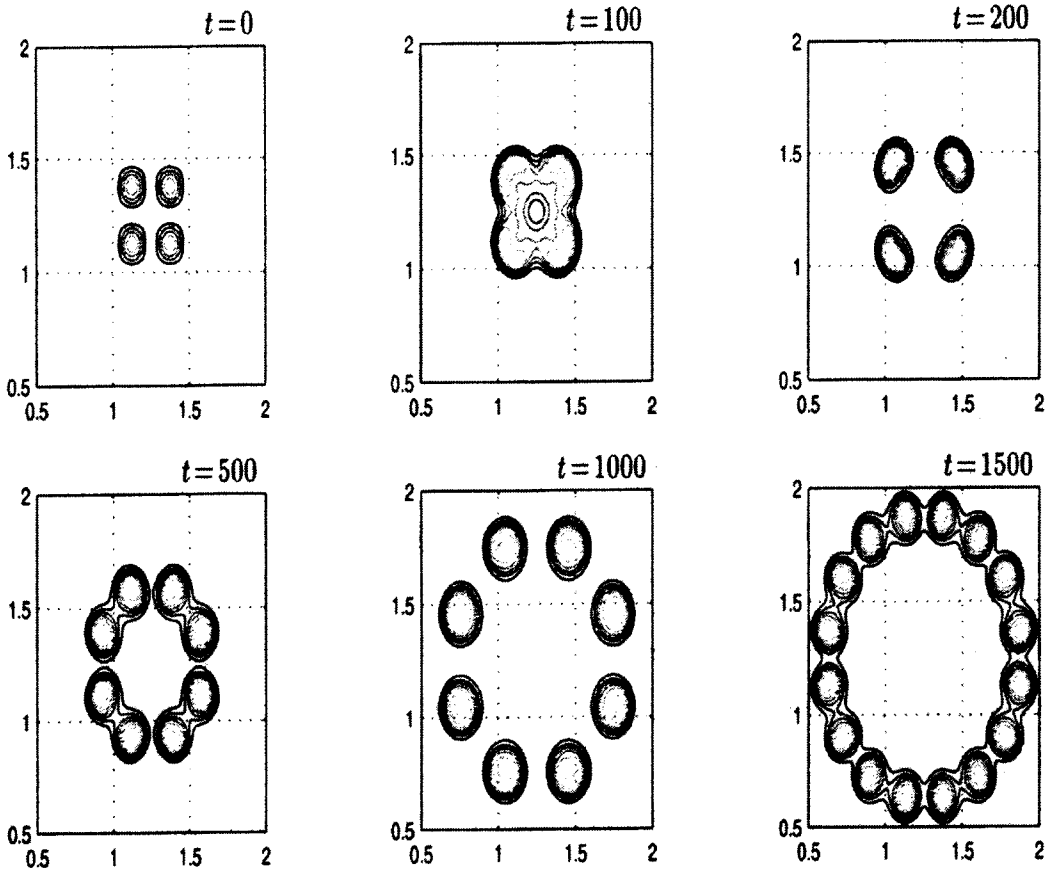

Figure 1.1: Time evolution (v-component) in the Gray–Scott model.

ODEs. Consider the initial value problem

$$w_t(t) = F(t, w(t)), \quad t > 0, \quad w(0) = w_0, \tag{1.0.4}$$

where $w(t) \in \mathbb{R}^m$. Let us first use forward Euler's (FE) method, which is the simplest numerical method, to approximate that system which gives

$$w_{n+1} = w_n + hF(t_n, w_n), \quad t_n = nh, \quad n = 0, 1, 2, \ldots, \tag{1.0.5}$$

where $w_n = w(t_n)$ and h is a time step.

Let $\|\cdot\|$ be a vector norm and consider a time interval $[0, T]$. For $K_0 \geq 0$, let

$$\mathcal{C} = \{(t, v) \in \mathbb{R} \times \mathbb{R}^m : 0 \leq t \leq T, \|v - w_0\| \leq K_0\}.$$

Definition 1.0.1 *If a function $F(t, v)$ satisfies the condition*

$$\|F(t, \tilde{v}) - F(t, v)\| \leq L\|\tilde{v} - v\| \text{ for all } (t, \tilde{v}), (t, v) \in \mathcal{C}, \tag{1.0.6}$$

then, it satisfies a Lipschitz condition and we call L the Lipschitz constant.

If F is continuous on $\mathcal{C}$ then (1.0.4) has a solution on some interval $[0, T^*]$ where $T^* > 0$. Moreover, if F is continuous on $\mathcal{C}$ and satisfies the Lipschitz condition then the solution is unique. Considering the scalar case, to obtain an error with Euler's method, we insert the exact solution values in the scheme to get

$$w(t_{n+1}) = w(t_n) + hF(t_n, w(t_n)) + h\rho_n. \tag{1.0.7}$$

Definition 1.0.2 (Truncation error) *This quantity*

$$\rho_n = \frac{w(t_{n+1}) - w(t_n)}{h} - F(t_n, w(t_n))$$

is called the (local) truncation error.

From first-order Taylor series expansions the truncation error could be simplified to be

$$\rho_n = \frac{1}{2}hw''(t_n) + \mathcal{O}(h^2),$$

provided that w is 3 times continuously differentiable on $[0, T]$.

Definition 1.0.3 (Discretization error) *This quantity*

$$\varepsilon_n = w(t_n) - w_n$$

is called the (global) discretization error.

It is easy to find a bound for ε_n using the Lipschitz condition.

Definition 1.0.4 (Consistency) *We say the scheme is consistent if* $\varepsilon_n \to 0$ *as* $h \to 0$.

If we have the following

$$\max_{n=[0,\frac{T}{h}]} \|w_n - w(t_n)\| \leq \mathcal{O}(h^p)$$

as $h \to 0$ we say the scheme is of order p.

Definition 1.0.5 (Stability) *We say the scheme is stable if* w_n, *where* $n = \frac{t}{h}$, *remains bounded as* $h \to 0$.

Let us apply the forward Euler (1.0.5) to the test equation $w'(t) = \lambda w(t)$ to get

$$w_{n+1} = w_n + \lambda h w_n = (1+z)w_n = R(z)w_n,$$

where $z = \lambda h$. Here we use $R(z)$ to represent the stability polynomial of the FE method. Also, the set of points that satisfies this condition $|R(z)| \leq 1$ is called the stability region and it is denoted by $\mathcal{S}$, i.e., $\mathcal{S} = \{z\colon |R(z)| \leq 1\}$. We will be considering different kinds of numerical stability.

Definition 1.0.6 (A-stable) *If the stability region of an ODE method contains the left half-plane* $\mathbb{C}^- = \{z \in \mathbb{C}\colon Re\ z \leq 0\}$, *then the system is A-stable.*

Definition 1.0.7 (strongly A-stable) *If the method is A-stable and* $|R(-\infty)| \leq 1$, *then it is strongly A-stable.*

Definition 1.0.8 (L-stable) *If the method is A-stable and* $|R(\infty)| = 0$, *then it is L-stable.*

Both strongly A-stable and L-stable methods are recommended for parabolic PDE's and stiff chemical systems [39].

Definition 1.0.9 (Convergence) *We say the scheme is convergent if*

$$\max_{n=[0,\frac{T}{h}]} \|w_n - w(t_n)\| \to 0 \quad as\ h \to 0,$$

where T *is the final time.*

Stiff Problem The phenomenon of stiffness has no closed definition in the literature, however, many attempts have been made, for more details and examples see [35, 46, 47]. One character of a stiff problem is it contains widely varying time scales, i.e., some components of the solution grow or decay much more rapidly than others. Due to this, in general, it is found that implicit methods perform better than explicit ones. A typical definition of stiffness you might find in the literature is based on the ratio of the smallest and largest real parts of eigenvalues of the associated linearized systems. This is usually coupled with the assumption that no real parts are positive. The larger that ratio the stiffer the problem.

1.0.3 Spatial Discretizations

To study numerical methods for time-dependent PDEs, we consider the discretization of the spatial operator $\triangle$ separately from the temporal operator $\frac{d}{dt}$. When we discretize the spatial operator on a chosen grid, the PDE becomes a system of ODEs. This is called a semi-discrete system. Here, we consider the simple spatial discretizations of a diffusion problem on a uniform grid $\Omega_h = x_1, x_2, \ldots, x_m$ with grid points $x_j = jh$ and mesh width $h = \frac{1}{m}$. We have $w_j(t) \approx u(x_j, t)$ on this spatial grid.

Consider the constant coefficient diffusion equation

$$u_t = \alpha u_{xx}, \tag{1.0.8}$$

with $\alpha > 0$, and assume periodicity $u(x \pm 1, t) = u(x, t)$ in space. Now the spatial derivative can be approximated by the difference formula

$$\frac{1}{h^2}\left(u(x-h) - 2u(x) + u(x+h)\right) = u_{xx} + \mathcal{O}(h^2). \tag{1.0.9}$$

Using this second-order formula on the uniform spatial grid Ω_h will lead to the second-order centred discretization

$$w'_j = \frac{\alpha}{h^2}\left(w_{j-1}(t) - 2w_j(t) + w_{j+1}(t)\right), \quad j = 1\,,\ \ldots,\ m\,, \tag{1.0.10}$$

where $w_0(t) = w_m(t)$ and $w_{m+1}(t) = w_1(t)$ from the periodicity condition. Equation (1.0.10) can be written as the following ODE system

$$w'(t) = Aw(t),$$

where $w = (w_1, \ldots, w_m)^T$, A is given by

$$A = \frac{\alpha}{h^2}\begin{bmatrix} -2 & 1 & \ldots & \ldots & 1 \\ 1 & -2 & 1 & \ldots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ldots & 1 & -2 & 1 \\ 1 & \ldots & \ldots & 1 & -2 \end{bmatrix}$$

and $w(0)$ is the given initial value. We notice here that A is symmetric and thus has real eigenvalues. This gives a finite difference spatial discretization scheme, which is still time-continuous. Therefore, we need a time-stepping scheme to discretize the problem with respect to time to get a fully discrete system. This question will be discussed shortly.

Fourier Analysis One good tool to analyze a finite difference scheme is the discrete Fourier decomposition. For more details see, for example, [74, 75, 81]. Periodicity conditions will be considered for test purposes. Recall the Fourier modes have the following forms

$$\phi_k(x) = e^{2\pi i k x}, \quad k \in \mathbb{Z}.$$

Here the discrete Fourier modes are

$$\phi_k = (\phi_k(x_1), \phi_k(x_2), \ldots, \phi_k(x_m))^T \in \mathbb{C}^m, \quad k \in \mathbb{Z}.$$

Now, consider the initial value problem

$$w'(t) = Aw(t), \quad w(0) = \sum_{k=1}^{m} \alpha_k \phi_k, \tag{1.0.11}$$

where A is an $m \times m$ circulant matrix which has the form

$$A = \begin{bmatrix} c_0 & c_1 & \cdots & \cdots & c_{m-1} \\ c_{m-1} & c_0 & c_1 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & c_{m-1} & c_0 & c_1 \\ c_1 & \cdots & \cdots & c_{m-1} & c_0 \end{bmatrix}. \tag{1.0.12}$$

This circulant matrix arises when periodic PDE problems with constant coefficients are discretized and has the property $A\phi_k = \lambda_k \phi_k$ where

$$\lambda_k = \sum_{j=0}^{m-1} c_j e^{2\pi i k x_j}.$$

Through some calculations, the solution of (1.0.11) may be expressed as

$$w(t) = \sum_{k=1}^{m} \alpha_k e^{\lambda_k t} \phi_k. \tag{1.0.13}$$

If we have $w(0) = \phi_k$ as the initial condition, then

$$w_j(t) = e^{\lambda_k t} e^{2\pi i k x_j}. \tag{1.0.14}$$

By substituting ϕ_k in (1.0.10), we find that

$$\lambda_k = \frac{2\alpha}{h^2}(\cos(2\pi kh) - 1) = -\frac{4\alpha}{h^2}\sin^2(\pi kh), \quad k = 1, \ldots, m. \tag{1.0.15}$$

We notice that the eigenvalues, λ_k, are negative and real, and thus we have $\|e^{tA}\| \leq 1$.

1.0.4 Time Stepping for PDEs

In the previous subsection, the spatial discretization of our problem leads to a semi-discrete system of ODEs. Here, we discuss the method of lines (MOL) approach, which is basically a time stepping method. One reason to use the MOL approach is that many well-developed ODE methods exist. Assume we have a PDE problem that has $u(x,t)$ as a solution. Then, discretizing in space on Ω_h yields a semi-discrete system

$$w'(t) = F(t, w(t)), \quad 0 < t \leq T, \quad w(0) \text{ given}, \tag{1.0.16}$$

with $w(t) = (w_j(t))_{j=1}^m \in \mathbb{R}^m$ and m is the number of grid points. Now, in order to get a full discrete approximation, we apply a suitable ODE method. For example, applying the θ-method to (1.0.16) with step size τ for $t_n = n\tau$, $n = 1, 2, \ldots$, yields

$$w_{n+1} = w_n + \tau(1-\theta)F(t_n, w_n) + \tau\theta F(t_{n+1}, w_{n+1}). \tag{1.0.17}$$

Here, $w_n = (w_j{}^n)_{j=1}^m \in \mathbb{R}^m$ denotes the fully discrete numerical solution at time t_n. Additional stepping methods are discussed in coming sections. For more on time stepping, see [13, 39, 47, 65, 72]. If we know that the spatial discretization is convergent of order p_1, i.e. $\|u_h(t) - w(t)\| \leq C_1 h^{p_1}$, and the time stepping method is convergent of order p_2, i.e. $\|w(t_n) - w_n\| \leq C_2\tau^{p_2}$, then we can find the error bound

$$\|u_h(t_n) - w_n\| \leq \|u_h(t_n) - w(t_n)\| + \|w(t_n) - w_n\| \leq C_1 h^{p_1} + C_2\tau^{p_2}. \tag{1.0.18}$$

For the stability, a restriction on the temporal step size τ in terms of the mesh width h is imposed. In the numerical literature, this restriction is called the Courant–Friedrichs–Lewy (CFL) condition and is associated with all explicit schemes.

1.1 Time Integration Method

In this section, we will introduce some methods which are good for the temporal discretization of PDEs. As discussed, after discretizing the space variable of a PDE, we will have a system of ODEs. In order to deal with this system we generally need a numerical method. There are two common classes of methods for solving initial value problems for systems of ODEs numerically, Runge–Kutta and linear multistep methods. For more details see [39, 67, 13, 72, 81].

We consider the general form of an initial value problem for a system of ODEs,

$$w'(t) = F(t, w(t)), \quad t > 0, \quad w(0) = w_0, \tag{1.1.1}$$

with given $F\colon \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ and $w_0 \in \mathbb{R}^m$. The exact solution $w(t)$ will be approximated at the points $t_n = n\tau$, $n = 0, 1, 2, \ldots$ with $\tau \geq 0$ being the step size. The numerical approximations are denoted by $w_n \approx w(t_n)$.

1.1.1 Runge–Kutta Methods

Runge–Kutta methods are one-step methods. That means that they step forward from computed approximations w_n at time t_n to new approximations w_{n+1} at the forward times t_{n+1} using only w_n as input. One-step methods have several advantages over multistep methods. One of these is that they are self-starting, which means that from the initial data w_0 the method can be applied immediately whereas multistep methods require other methods to be used initially.

The general form of a Runge–Kutta method is

$$\begin{aligned} w_{n+1} &= w_n + \tau \sum_{i=1}^{s} b_i F(t_n + c_i\tau, w_{ni}), \\ w_{ni} &= w_n + \tau \sum_{j=1}^{s} \alpha_{ij} F(t_n + c_j\tau, w_{nj}), \end{aligned} \tag{1.1.2}$$

with $i = 1, \ldots, s$ and $w_{ni} \approx w(t_n + c_i\tau)$ where the integer s is called the number of stages. Here α_{ij}, b_i are coefficients defining the particular method and $c_i = \sum_{j=1}^{s} \alpha_{ij}$.

The method is called explicit if $\alpha_{ij} = 0$ for $j \geq i$, because the internal approximation w_{ni} can be computed one after another from an explicit relation. Otherwise, the method is called implicit due to the fact that the w_{ni} must be retrieved from a system of linear or nonlinear algebraic relations. The Runge–Kutta method is represented by the Butcher array

$$\begin{array}{c|ccc} c_1 & \alpha_{11} & \cdots & \alpha_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & \alpha_{s1} & \cdots & \alpha_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} = \begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

The coefficients α_{ij}, b_i, c_i from (1.1.2) will determine the order of the method. For example, the following array

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

will provide us with two stages, which in fact is the second-order explicit trapezoidal rule

$$w_{n+1} = w_n + \frac{1}{2}\tau F(t_n, w_n) + \frac{1}{2}\tau F\left(t_n + \tau, w_n + \tau F(t_n, w_n)\right). \tag{1.1.3}$$

For implicit RK, we can see this

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array},$$

which will produce the backward Euler method

$$w_{n+1} = w_n + \tau F(t_{n+1}, w_{n+1}). \tag{1.1.4}$$

Accuracy

In general, the global error, at the fixed time point t_N, is the difference between the exact $w(t_N)$ and the approximation w_N. Clearly, the global error at t_N must depend on errors present in all preceding approximations w_n $(1 \leq n \leq N-1)$. So, the global error can be interpreted as being built up from local errors.

The Stability Analysis

Let us consider the scalar complex equation

$$w_t(t) = \lambda w(t), \tag{1.1.5}$$

where $\lambda \in \mathbb{C}$ with negative real part. We will study the stability behaviour of numerical ODE methods using this test equation. Applying (1.1.2) to the test equation and letting $z = \tau\lambda$, we get

$$w_{n+1} = R(z)w_n,$$

where

$$R(z) = 1 + z + \frac{1}{2}z^2 + \cdots + \frac{1}{s!}z^s, \tag{1.1.6}$$

is the stability function of the explicit RK method for $p = s \leq 4$.

We notice from from Figure 1.2 that the stability region of explicit RK methods, associated with this stability function (1.1.6), generally gets bigger as s gets bigger.

When we solve a stiff system, the order of the error can be reduced compared to what the classical theory predicts. This phenomenon is called the order reduction, though, it is difficult to be detected in practical computations. In general, any explicit or implicit RK method of order $p \geq 3$ and stage order $q \leq p-1$ may suffer from order reduction to the level $q+1$. In addition to implicit RK methods, Rosenbrock Methods are RK type (one-step) methods suitable for stiff autonomous ODEs, see [35].

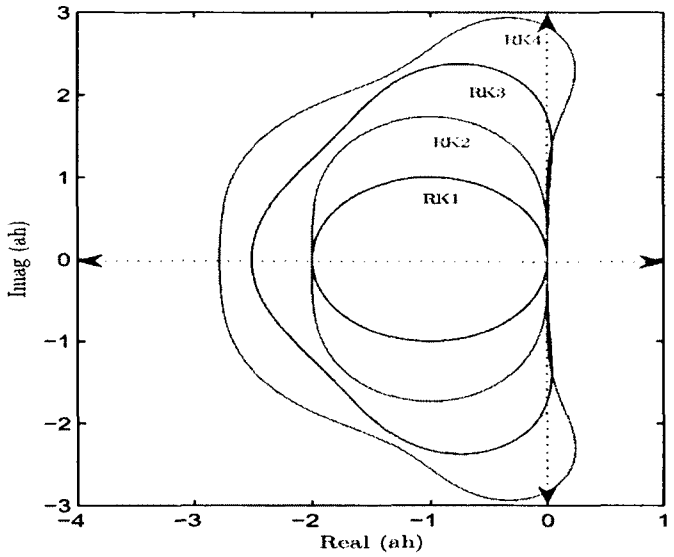

Figure 1.2: Stability regions (interior of the closed curves) for the explicit first-order Runge–Kutta (RK1), second-order Runge–Kutta (RK2), third-order Runge–Kutta (RK3), and fourth-order Runge–Kutta (RK4).

1.1.2 Linear Multistep Methods

Linear multistep methods (LMM) differ from RK methods in that they use additional past results from past times. A linear k-step is method defined by

$$\sum_{j=0}^{k} \alpha_j w_{n+j} = \tau \sum_{j=0}^{k} \beta_j F(t_{n+j}, w_{n+j}), \quad n = 0, 1, \ldots, \tag{1.1.7}$$

where α_j and β_j are constants defining the method. Thus, to compute w_{n+k}, we use the k past values $w_n, \ldots, , w_{n+k-1}$. Given an initial value, we need to generate other starting values by, for example, RK or using a linear one-step method to find w_1, then a linear two-step method for w_2 and so on. If $\beta_k = 0$, then the method is explicit, otherwise it is implicit.

Scaling is used to set $\alpha_k = 1$ or $\sum_{j=0}^{k} \beta_j = 1$. The linear k-step has some computational advantages over one-step s-stage RK method. For example, in the case of explicit methods one instance of F needs to be evaluated versus s in RK methods.

Also, we have to consider the difficulty that (1.1.7) needs k starting values for the first step.

The Order Conditions

We will use the same notations as in RK section. Consider the following

$$\sum_{j=0}^{k-1} \alpha_j w_{n+j} + \alpha_k w^*_{n+k} = \tau \sum_{j=0}^{k-1} \beta_j F(t_{n+j}, w_{n+j}) + \tau \beta_k F(t_{n+j}, w^*_{n+k}). \tag{1.1.8}$$

The LMM is consistent of order p, if the local error satisfies

$$w(t_{n+1}) - w^*_{n+1} = \mathcal{O}(\tau^{p+1}) \tag{1.1.9}$$

and F is sufficiently differentiable. Then the global error would be $\mathcal{O}(\tau^p)$. It is found, see [39], that the method has order p iff it satisfies the order conditions

$$\sum_{j=0}^{k} \alpha_j = 0, \quad \sum_{j=0}^{k} \alpha_j j^i = i \sum_{j=0}^{k} \beta_j j^{i-1} \quad \text{for } i = 1, 2, \ldots, p. \tag{1.1.10}$$

For example, Adams methods are characterized by

$$\alpha_k = 1, \quad \alpha_{k-1} = -1, \quad \alpha_j = 0 \ \ (0 \le j \le k-2)$$

with β_j chosen such that the order is optimal. Adams–Bashforth method (from [47]) of order $k = 2$, which is explicit,

$$w_{n+2} - w_{n+1} = \frac{3}{2}\tau F_{n+1} - \frac{1}{2}\tau F_n, \tag{1.1.11}$$

where $F_i = F(t_i, w_i)$. For implicit Adams methods, the Adams–Moulton method (from [47]) of order $k + 1 = 3$ is

$$w_{n+2} - w_{n+1} = \frac{5}{12}\tau F_{n+2} + \frac{8}{12}\tau F_{n+1} - \frac{1}{12}\tau F_n. \tag{1.1.12}$$

In this implicit case, predictor-corrector ideas are applied. First, a predicted value $\overline{w}_{n+k}$ is computed by an explicit k-step method; then, we insert the value in the right-hand side of the implicit k-step method.

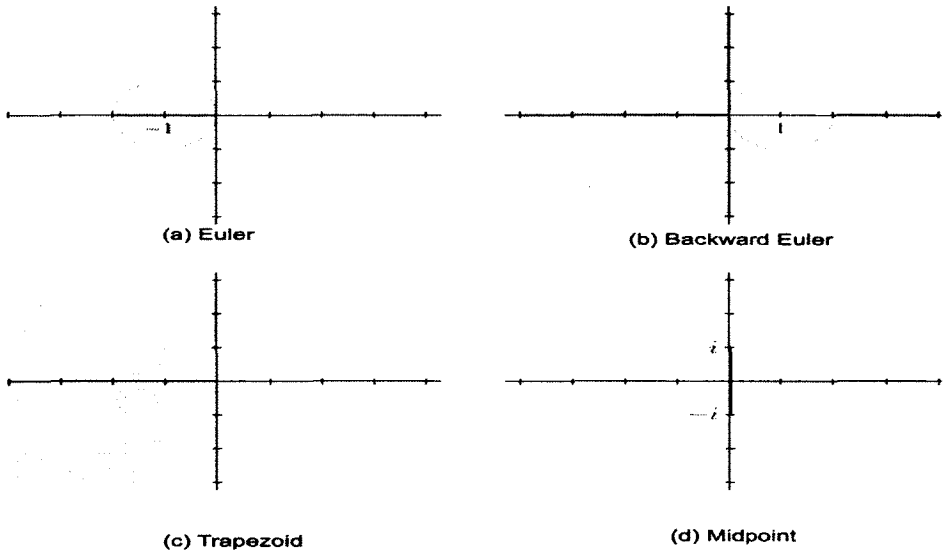

Figure 1.3: Stability regions for Euler, Backward Euler, trapezoidal, and midpoint methods.

In addition to these examples, there are other LMMs which have good stability properties when stiff problems are considered. One class of such methods is the backward differentiation formula (BDF). A BDF method is implicit and is defined by

$$\beta_k = 1, \quad \beta_j = 0 \ \ (0 \leq j \leq k-1), \tag{1.1.13}$$

with α_j chosen such that the order is k. The 2-step BDF is

$$\frac{3}{2}w_{n+2} - 2w_{n+1} + \frac{1}{2}w_n = \tau F_{n+2}. \tag{1.1.14}$$

Stability Analysis

First, let us introduce some examples of LMM in Figure 1.3 with their stability regions. It can be noticed in Figure 1.3 that both the backward Euler and the trapezoidal rule are unconditionally stable, i.e. A-stable.

Now, recall some properties of the scalar linear equation

$$\sum_{j=0}^{k} \gamma_j w_{n+j} = 0, \quad n = 0,\ 1,\ \ldots, \tag{1.1.15}$$

where $\gamma_j \in \mathbb{C}$. Define the characteristic polynomial, which helps to study the behaviour of $\{w_n\}$, as

$$\pi(z) = \sum_{j=0}^{k} \gamma_j z^j. \tag{1.1.16}$$

Let us denote the zeros of $\pi(z)$ by $z_1, z_2, \ldots, z_k$. Thus, w_n is a linear combination of z_i. The characteristic polynomial satisfies the root condition if

$$|z_i| \leq 1 \quad \text{for all } i, \text{ and } |z_i| < 1 \text{ if } z_i \text{ is not simple.} \tag{1.1.17}$$

With a selection of $\{\alpha_j\}$ and $\{\beta_j\}$, the first and second characteristic polynomials

$$\rho(z) = \sum_{j=0}^{k} \alpha_j z^j \qquad \sigma(z) = \sum_{j=0}^{k} \beta_j z^j. \tag{1.1.18}$$

will be used for discussing the stability. If the LMM is implicit then the polynomials ρ and σ have degree exactly k. If σ has degree $< k$ the formula is explicit. So, specifying ρ and σ is equivalent to specifying the LMM by its coefficients. For example, Euler has $\rho(z) = z - 1$ and $\sigma(z) = 1$, and the trapezoidal rule has $\rho(z) = z - 1$ and $\sigma(z) = \frac{1}{2}(z + 1)$ and so on.

Zero Stability We discuss zero stability, which is a minimum condition for a method to be useful. Letting $F = 0$ in (1.1.7), then we have

$$\sum_{j=0}^{k} \alpha_j w_{n+j} = 0, \quad n = 0, 1, \ldots, \tag{1.1.19}$$

with $\rho(z)$ as the characteristic polynomial. We say that a LMM is zero-stable if $\rho(z)$ satisfies the root condition (1.1.17). For example, consider the two-step explicit method

$$w_{n+2} - (1 + \alpha_0)w_{n+1} + \alpha_0 w_n = \frac{1}{2}(3 - \alpha_0)\tau F_{n+1} - \frac{1}{2}(1 + \alpha_0)\tau F_n.$$

With $\alpha_0 = -5$ we get order 3,

$$\rho(z) = \sum_{j=0}^{2} \alpha_j z^j = -5 + 4z + z^2,$$

which implies that $z_1 = 1$ and $z_2 = -5$ and the method does not satisfy the root condition. Therefore, this method is not zero-stable.

Stability Region Consider the scalar test equation $w'(t) = \lambda w(t)$, where $\lambda \in \mathbb{C}$, so our LMM would be

$$\sum_{j=0}^{k} (\alpha_j - Z\beta_j) w_{n+j} = 0, \qquad n = 0, 1, \ldots, \tag{1.1.20}$$

where $Z = \tau\lambda$. It has the characteristic polynomial

$$\pi_Z(z) = \rho(z) - Z\sigma(z) = \sum_{j=0}^{k} (\alpha_j - Z\beta_j) z^j. \tag{1.1.21}$$

Letting $\mathcal{S}$ denote the stability region, it is clear that

$$Z \in \mathcal{S} \iff \pi_Z \text{ satisfies the root condition.}$$

On the boundary of $\mathcal{S}$ one root of π_Z has modulus 1. Since $\pi_Z(z) = 0$ iff $Z = \frac{\rho(z)}{\sigma(z)}$, then any point in the boundary would be of this form

$$Z = \tau\lambda = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})}, \quad 0 \le \theta \le 2\pi. \tag{1.1.22}$$

The stability regions of the explicit Adams–Bashforth (AB), implicit Adams–Moulton (AM), and backward differentiation formulas (BDF) are given in Figures 1.4, 1.5, and 1.6, respectively.

It can be noticed that the stability region of AB is small and becomes smaller as the order increases. Also, in the case of AM, we get a bounded region which is not desirable for implicit methods and is therefore not as advantageous for stiff problems. On the other hand, BDF shows a good unbounded stability region (see Figure 1.6).

***A*-Stability** First, one of the desirable properties for stiff problems is the A-stability property. A linear multistep method is called A-stable if its stability region satisfies

$$\mathcal{S} \supset \{z \in \bar{\mathbb{C}}: \ Re\, z \le 0 \text{ or } z = -\infty\},$$

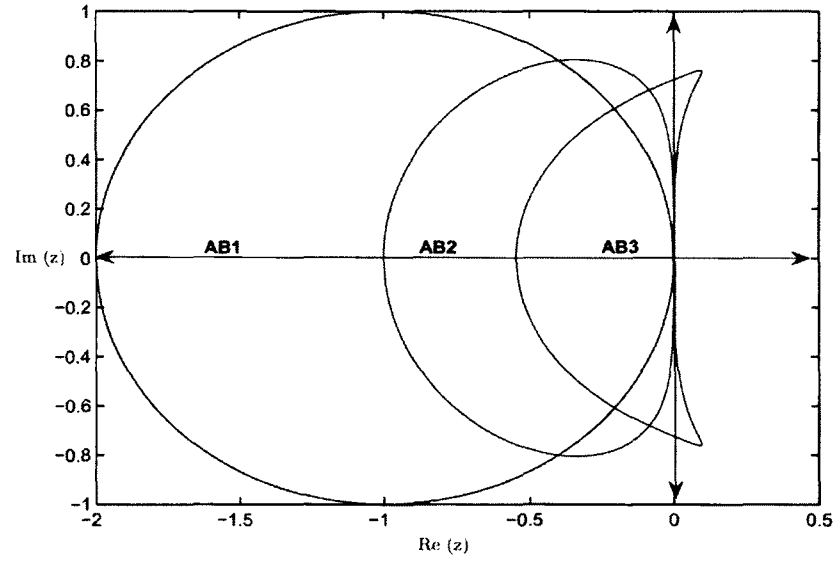

Figure 1.4: Boundaries of stability regions (interior) for 1-step Adams–Bashforth (AB1), 2-step Adams–Bashforth (AB2), and 3-step Adams–Bashforth (AB3).

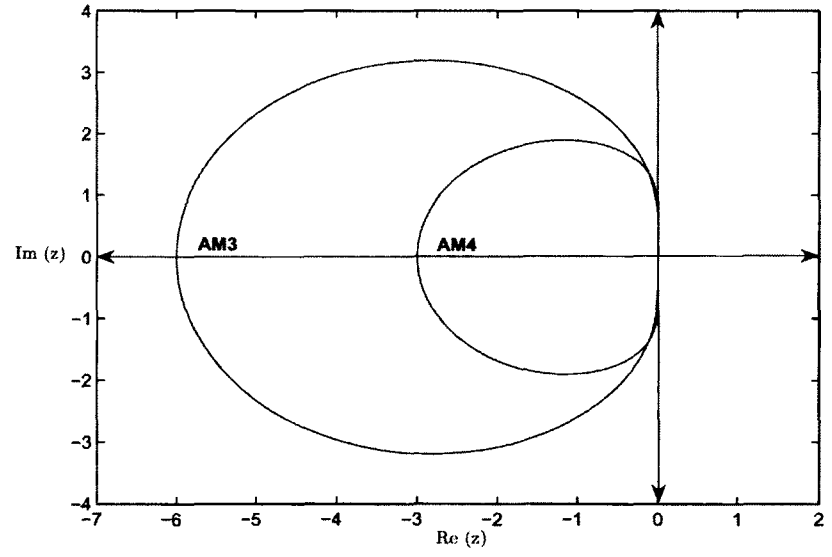

Figure 1.5: Boundaries of stability regions (interior) for 2-step Adams–Moulton (AM3) and 3-step Adams–Moulton (AM4).

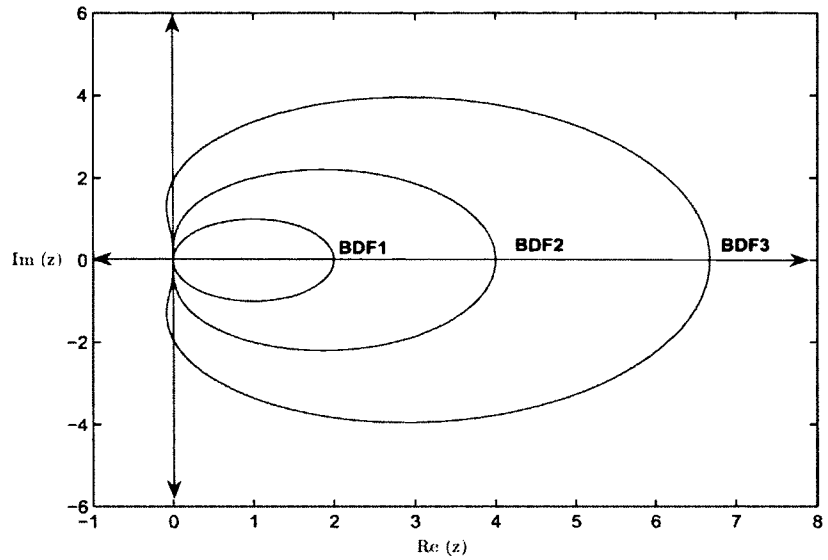

Figure 1.6: Boundaries of stability regions (exterior) for 1-step backward differentiation formula (BDF1), 2-step backward differentiation formula (BDF2), and 3-step backward differentiation formula (BDF3).

where $\bar{\mathbb{C}} = \mathbb{C} \cup \infty$. For LMM, it is convenient to include $z = -\infty$. There are fewer A-stable LMMs than in the RK case.

Another property that could be introduced here is $A(\alpha)$-stability, with angles $\alpha \in [0, \frac{1}{2}\pi]$. For $z = re^{i\phi}$ with $r > 0$, $\phi \in (-\pi, \pi]$, let $\arg(z) = \phi$. Thus, a method is said to be $A(\alpha)$-stable if

$$\mathcal{S} \supset \left\{ z \in \bar{\mathbb{C}} \colon \; z = 0, \infty \;\text{ or }\; |\arg(-z)| \leq \alpha \right\}.$$

The angle α will be read from the negative real axis. For example, in Figure 1.6, $\alpha = \frac{\pi}{2}$ for $k = 1, 2$ but $\alpha = \frac{43\pi}{90} < \frac{\pi}{2}$ for $k = 3$. From [39], we have the following table

k	1	2	3	4	5	6	7
α	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{43\pi}{90}$	$\frac{73\pi}{180}$	$\frac{51\pi}{180}$	$\frac{71\pi}{180}$	not zero-stable

1.1.3 Comparisons

The RK methods are self-starting while the LMMs need to generate starting values using other schemes. On the other hand, RK schemes suffer from order reduction. Computationally, the cost of function evaluation in the LMM is much less than in RK. This is because in the case of the RK, we have to compute at fractional times (off-step points), while in the case of the LMM, we use already known points. From a stability point of view, in both cases, we have a number of implicit schemes where the stability region is unbounded. However, using an A-stable implicit RK or LMM scheme may not be as efficient as we need for our reaction diffusion equation, since the eigenvalues are near the negative real axis. Therefore, having an explicit scheme, such as the stabilized Runge–Kutta can reduce computational costs. This will be discussed in Section 3.

1.2 Splitting Methods

Applying one integration formula to the different parts of the reaction diffusion equations is generally inefficient. For example, using a single implicit integration formula for the whole problem will lead to a large nonlinear algebraic system. The idea behind splitting is to break down a complicated problem into smaller parts and then treating each part with a suitable integration formula. For more details see, for instance, [39, 65]. We will introduce briefly three classes of splitting methods: locally one-dimensional (LOD), alternating direction implicit (ADI), and implicit-explicit (IMEX) methods.

1.2.1 Operator Splitting

We focus on concepts rather than on actual methods. Also we will consider the linear case; the nonlinear case can be treated the same way.

First-Order Splitting

Consider a linear constant coefficient homogeneous ODE system

$$w' = Aw(t), \quad t \geq 0, \quad w(0) = w_0. \tag{1.2.1}$$

Its solution may be formally written as

$$w(t_{n+1}) = e^{\tau A} w(t_n). \tag{1.2.2}$$

Now, suppose $A = A_1 + A_2$ and $\tau = t_{n+1} - t_n$. Then, instead of solving (1.2.1), we can solve the two subproblems

$$w_t^*(t) = A_1 w^*(t)$$

for $t_n \leq t \leq t_{n+1}$ with $w^*(t_n) = w_n$ to get $w^*(t_{n+1})$, and

$$w_t^{**}(t) = A_2 w^{**}(t)$$

for $t_n \leq t \leq t_{n+1}$ with $w^{**}(t_n) = w^*(t_{n+1})$, starting from w_n and take $w_{n+1} = w^{**}(t_{n+1})$ to complete the splitting integration step. With this, an error term arises, and the truncation error will satisfy

$$\rho_n = \frac{1}{\tau}(e^{\tau A} - e^{\tau A_2} e^{\tau A_1}) w(t_n) = \frac{1}{2}\tau [A_1, A_2] w(t_n) + \mathcal{O}(\tau^2), \tag{1.2.3}$$

with $[A_1, A_2] = A_1 A_2 - A_2 A_1$ being the commutator of A_1 and A_2. If A_1 and A_2 commute, we have

$$e^{A_2} e^{A_1} = e^{A_2 + A_1} = e^A$$

and it leaves no splitting error. In general, it cannot be assumed that A_1 and A_2 commute. Thus, we discuss the splitting method error. The leading term $\mathcal{O}(\tau)$ in (1.2.3) dominates the higher-order terms which requires $[A_1, A_2] w(t_n) = \mathcal{O}(1)$. In general, there is freedom in the selection of a suitable numerical ODE integration

method to deal with the sub-step problem, but unavoidably the integration error and splitting error accumulate. For stability, if

$$\|e^{\tau A_k}\| \leq 1$$

then

$$\|w_{n+1}\| \leq \|w_n\|.$$

In general, if the sub-steps are stable then the splitting will be stable as well.

The last example can be generalized to multi-components. If we have $A = A_1 + A_2 + A_3$, then the first-order splitting will be

$$w(t_{n+1}) = e^{\tau A_1} e^{\tau A_2} e^{\tau A_3} w(t_n). \tag{1.2.4}$$

Second-Order Splitting

There is a proposed splitting method called Strang splitting [73] based on symmetry. It is represented by

$$w_{n+1} = (e^{\frac{1}{2}\tau A_1} e^{\frac{1}{2}\tau A_2})(e^{\frac{1}{2}\tau A_2} e^{\frac{1}{2}\tau A_1}) w_n = e^{\frac{1}{2}\tau A_1} e^{\tau A_2} e^{\frac{1}{2}\tau A_1} w_n. \tag{1.2.5}$$

The truncation error is

$$\rho_n = \frac{1}{24}\tau^2([A_1, [A_1, A_2]] + 2[A_2, [A_1, A_2]]) w(t_{n+\frac{1}{2}}) + \mathcal{O}(\tau^4). \tag{1.2.6}$$

In general, higher-order splitting methods have been discussed for linear operators [39]. The idea of splitting can be applied not only if A_1 and A_2 are constant matrices, but also for general operators. So the commutator in the case of the operators f_1 and f_2 would be

$$[f_1, f_2](u) = f_1'(u) f_2(u) - f_2'(u) f_1(u),$$

where the primes denote derivatives with respect to u.

Reaction Diffusion Splitting

Consider the system

$$u_t = f_D(u) + f_R(u), \tag{1.2.7}$$

where $f_D(u) = \alpha\triangle u$, and $f_R(u) = F(u)$. This splitting will have some advantages such as offering means for parallel computing. For $[f_D, f_R](u) = 0$, we need

$$f_D'(u)f_R(u) = \alpha\triangle f_R(u)$$

and

$$f_R'(u)f_D(u) = f_R'(u)(\alpha\triangle u)$$

to be equal, and hence, diffusion commutes with reaction if f is linear in u and independent of x. This condition will not be satisfied in a real problem where the reaction term is nonlinear. But if $F(u) = Lu + \tilde{f}(u)$, where L is independent of x, then the truncation error for first-order splitting will be

$$\rho_n = \frac{1}{2}\tau[\alpha\triangle(\tilde{f}(u)) - \tilde{f}'(u)(\alpha\triangle u)](t_n) + \mathcal{O}(\tau^2).$$

1.2.2 Locally One-Dimensional (LOD) Methods

Now, rather than solving a couple of large systems obtained from temporal discretization for all unknowns on the grid simultaneously, we could replace the fully coupled single time steps with a sequence of steps. Each step is coupled in only one space direction. This splitting becomes effectively one-dimensional. For this reason, the method is known as locally one-dimensional (LOD) and will be discussed in coming sections. For more details see [39, 80].

LOD-Backward Euler (LOD-BE) Method

Consider the nonlinear semi-discrete ODE system $w'(t) = F(t, w(t))$ in $\mathbb{R}^m$ and apply the multiple splitting

$$F(t, v) = F_1(t, v) + F_2(t, v) + \cdots + F_s(t, v).$$

Combining first-order splitting with first-order backward Euler method for all fractional (non-integer) steps gives the LOD-BE method

$$\begin{aligned} v_0 &= w_n, \\ v_i &= v_{i-1} + \tau F_i(t_{n+1}, v_i), \quad i = 1, \ldots, s, \\ w_{n+1} &= v_s, \end{aligned} \tag{1.2.8}$$

where $v_1, \ldots, v_{s-1}$ are internal vectors for the step from t_n to t_{n+1}. The order of consistency will depend on the ODE problem.

LOD-Crank–Nicolson (LOD-CN) Methods

Combining first-order splitting with the second-order implicit trapezoidal rule (Crank–Nicolson) method for all fractional steps gives the LOD-CN method,

$$\begin{aligned} v_0 &= w_n, \\ v_i &= v_{i-1} + \frac{1}{2}\tau F_i(t_n + c_{i-1}\tau, v_{i-1}) + \frac{1}{2}\tau F_i(t_n + c_i\tau, v_i), \quad i = 1, \ldots, s, \\ w_{n+1} &= v_s. \end{aligned} \tag{1.2.9}$$

Since the v_i are not consistent approximations to the exact solution, we can choose arbitrary c's such as $c_0 = 0$, $c_s = 1$, and $c_j = \frac{1}{2}$. This method has first-order consistency.

The Trapezoidal Splitting Method

Now, consider a half step with FE followed by a half step with BE, which will lead to the implicit trapezoidal rule. By using this splitting, we have the Trapezoidal Splitting,

$$\begin{aligned} v_0 &= w_n, \\ v_i &= v_{i-1} + \frac{1}{2}\tau F_i(t_n, v_{i-1}), \quad i = 1, \ldots, s, \end{aligned}$$

$$\begin{aligned} v_{s+i} &= v_{s+i-1} + \frac{1}{2}\tau F_{s+1-i}(t_{n+1}, v_{s+i}), \quad i = 1, \ldots, s, \\ w_{n+1} &= v_{2s}, \end{aligned} \tag{1.2.10}$$

where v_i $(1 \leq i \leq 2s-1)$ are internal quantities without physical relevance. In the ODE sense, method (1.2.10) has order two.

1.2.3 Alternating Direction Implicit (ADI) Methods

The alternating direction implicit (ADI) methods are the second most well-known splitting methods. We will introduce two ADI methods, the Peaceman–Rachford method and the Douglas method. For more details see [25, 84].

The Peaceman–Rachford Method

Consider the ODE system $w'(t) = F(t, w(t))$ with the two-term splitting

$$F(t, v) = F_1(t, v) + F_2(t, v),$$

so the Peaceman-Rachford method is

$$\begin{aligned} w_{n+\frac{1}{2}} &= w_n + \frac{1}{2}\tau F_1(t_n, w_n) + \frac{1}{2}\tau F_2(t_{n+\frac{1}{2}}, w_{n+\frac{1}{2}}), \\ w_{n+1} &= w_{n+\frac{1}{2}} + \frac{1}{2}\tau F_2(t_{n+\frac{1}{2}}, w_{n+\frac{1}{2}}) + \frac{1}{2}\tau F_1(t_{n+1}, w_{n+1}). \end{aligned} \tag{1.2.11}$$

This method does not have a natural extension to more than two F-components. Here, both F_1 and F_2 will be used in both stages and the alternate implicit use of F_1 and F_2 leads to the name of the method.

The Douglas Method

The Douglas ADI method is better than the PR ADI method in dealing with multi-component splitting of F. Suppose

$$F(t, v) = F_0(t, v) + F_1(t, v) + \cdots + F_s(t, v), \tag{1.2.12}$$

and assume that F_0 is non-stiff or mildy stiff and all other F_i can be stiff. Then we have

$$\begin{aligned} v_0 &= w_n + \tau F(t_n, w_n), \\ v_i &= v_{i-1} + \theta\tau(F_i(t_{n+1}, v_i) - F_i(t_n, w_n)), \quad i = 1, \ldots, s, \\ w_{n+1} &= v_s. \end{aligned} \tag{1.2.13}$$

If $\theta = 1$ then the method is of order one. Also, the method would be of order two if $\theta = \frac{1}{2}$ and $F_0 = 0$. One property of (1.2.13) is all internal vectors v_i are consistent approximations to $w(t_{n+1})$. For a problem with large boundary conditions, Douglas method performs better than LOD methods.

Finally, comparing the Douglas method to the LOD-CN and trapezoidal, the Douglas method performs better and has second order of convergence, see [39]. Also, the Douglas method does not suffer from order reduction. On the other hand, in the case $s \geq 3$, extra care needs to be taken when dealing with Douglas method.

1.2.4 Implicit-Explicit (IMEX) Methods

We will discuss splitting methods with an improved treatment of the explicit terms in this section. These methods involve a mixture of implicit and explicit schemes, for example, using an explicit scheme such as the second-order Adams–Bashforth formula to advance the nonlinear part of the problem and an implicit scheme like the second-order Adams–Moulton formula, to advance the linear part. We consider here the one-step IMEX-θ and the IMEX multistep methods. Also, the IMEX-RK, a recently developed scheme will be introduced. IMEX schemes are still the subject of active research, see, for example [4, 43, 68, 57, 70, 85].

The IMEX-θ Method

Let us introduce the idea through the one-step IMEX-θ. Consider the semi-discrete system

$$w'(t) = F(t, w(t)) \equiv F_0(t, w(t)) + F_1(t, w(t)), \tag{1.2.14}$$

where F_0 takes care of the non-stiff term and F_1 takes care of the stiff term. The simplest IMEX method, which is a combination between an explicit Euler method and an A-stable implicit θ-method, is the following:

$$w_{n+1} = w_n + \tau F_0(t_n, w_n) + (1-\theta)\tau F_1(t_n, w_n) + \theta\tau F_1(t_{n+1}, w_{n+1}), \tag{1.2.15}$$

where $\theta \geq \frac{1}{2}$. The temporal truncation error is

$$\begin{aligned} \rho_n &= \tau^{-1}(w(t_{n+1}) - w(t_n)) - (1-\theta)F(t_n, w(t_n)) \\ &\quad - \theta F(t_{n+1}, w(t_{n+1})) + \theta(F_0(t_{n+1}, w(t_{n+1})) - F_0(t_n, w(t_n))) \\ &= \left(\frac{1}{2} - \theta\right)\tau w''(t_n) + \theta\tau F_0'(t_n, w(t_n)) + \mathcal{O}(\tau^2). \end{aligned}$$

As we said, F_0 will represent the discretization of the non-stiff term so the smoothness of w will imply the smoothness of F_0, independently of boundary conditions or mesh widths h. For this reason, the truncation error of this method is better than with those splittings with fractional (non-integer) steps. We should, however, be careful when examining the stability.

Stability Consider the scalar complex test equation

$$w'(t) = \lambda_0 w(t) + \lambda_1 w(t), \tag{1.2.16}$$

where $\lambda_j \in \mathbb{C}$ with $z_j = \tau\lambda_j$, $j = 0, 1$. Applying (1.2.15) to this equation, we get

$$w_{n+1} = R(z_0, z_1)w_n,$$

with

$$R(z_0, z_1) = \frac{1 + z_0 + (1-\theta)z_1}{1 - \theta z_1}, \tag{1.2.17}$$

and we need to have $|R(z_0, z_1)| \leq 1$.

First, if we insist on A-stability with respect to the implicit part, we will consider the set

$$\mathcal{D}_0 = \{z_0 \in \mathbb{C}: \text{ the IMEX scheme is stable for any } z_1 \in \mathbb{C}^-\}$$

as the stability region. To plot the stability region, consider $z_1 = it$ where $t \in \mathbb{R}$ and $i = \sqrt{-1}$. So $z_0 = x_0 + iy_0 \in \mathcal{D}_0$ iff

$$(2\theta - 1)t^2 + 2(\theta - 1)y_0 t - (2x_0 + x_0^2 + y_0^2) \geq 0, \quad \text{for all } t \in \mathbb{R},$$

which is equivalent to

$$\frac{y_0^2}{\frac{(2\theta-1)}{\theta^2}} + (1 + x_0)^2 \leq 1, \quad \text{for } \theta > \frac{1}{2}.$$

These are equations of an ellipse, see the left part of Figure 1.7 for the plot of the boundaries of regions $\mathcal{D}_0$. If $\theta = 1$, we recover the stability region of the explicit Euler method and by reducing θ to be $\frac{1}{2}$, we get the negative line $[-2, 0]$.

Or, if we insist on using the full stability region of the explicit method, then we have the set

$$\mathcal{D}_1 = \{z_1 \in \mathbb{C}: \text{ the IMEX scheme is stable for any } z_0 \text{ such that } |1 + z_0| \leq 1\}.$$

We should have $||1 + z_0| + (1 - \theta)z_1| \leq |1 - \theta z_1|$.

Thus, we find that $z_1 \in \mathcal{D}_1$ iff

$$1 + |(1 - \theta)z_1| \leq |1 - \theta z_1|.$$

See the right part of Figure 1.7 for the plot of the boundaries of region $\mathcal{D}_1$.

When $\theta = 1$ the IMEX method would be seen as a time splitting method where we first solve $w'(t) = F_0(t, w(t))$ with forward Euler on $[t_n, t_{n+1}]$ and then solve $w'(t) = F_1(t, w(t))$ with backward Euler.

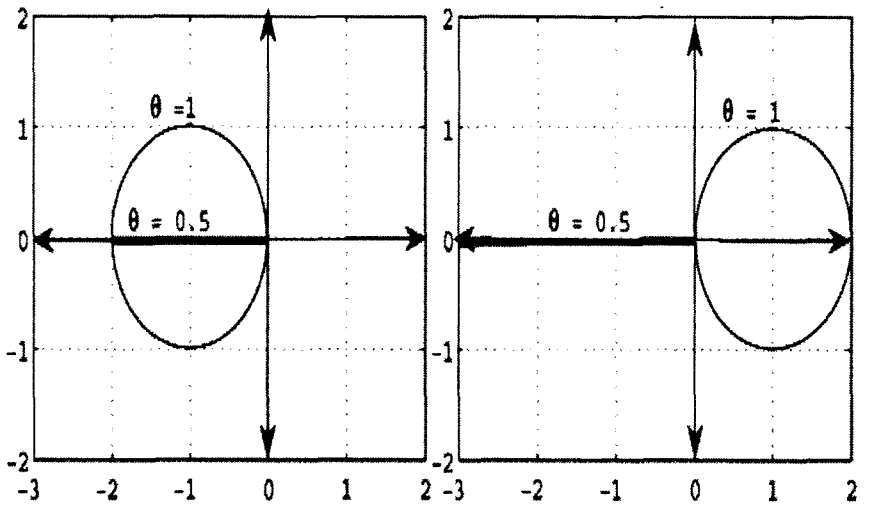

Figure 1.7: Boundaries of regions $\mathcal{D}_0$ (left) and $\mathcal{D}_1$ (right) for $\theta = 0.5, 1$.

The IMEX Multistep Methods

Now we discuss generalizations of the θ-method in order to have a higher-order accuracy. These generalizations are the IMEX Multistep Methods. Consider the implicit linear k-step method of order p

$$\sum_{j=0}^{k} \alpha_j w_{n+j} = \tau \sum_{j=0}^{k} \beta_j (F_0(t_{n+j}, w_{n+j}) + F_1(t_{n+j}, w_{n+j})) \tag{1.2.18}$$

where F_0 and F_1 are the separation terms. The F_0-terms can be simplified by applying the extrapolation formula

$$F_0(t_{n+k}, w(t_{n+k})) = \sum_{j=0}^{k-1} \gamma_j F_0(t_{n+j}, w(t_{n+j})) + \mathcal{O}(\tau^q).$$

Thus, (1.2.18) becomes

$$\sum_{j=0}^{k} \alpha_j w_{n+j} = \tau \sum_{j=0}^{k-1} \beta_j^* F_0(t_{n+j}, w_{n+j}) + \tau \sum_{j=0}^{k} \beta_j F_1(t_{n+j}, w_{n+j}), \tag{1.2.19}$$

with $\beta_j^* = \beta_j + \beta_k\gamma_j$. With $\phi(t) = F_0(t, w(t))$, the local truncation error can be written as

$$\begin{aligned}\rho &= \frac{1}{\tau}\sum_{j=0}^{k}\left(\alpha_j w_{n+j} - \tau\beta_j w'_{n+j}\right) + \beta_k\left(\phi(t_{n+k}) - \sum_{j=0}^{k-1}\gamma_j\phi(t_{n+j})\right) \\ &= C_1\tau^p w^{p+1}(t_n) + C_2\beta_k\tau^q\phi^q(t_n) + \mathcal{O}(\tau^{p+1}) + \mathcal{O}(\tau^{q+1}), \qquad (1.2.20)\end{aligned}$$

where the constants C_1 and C_2 are determined by the method. So the IMEX method (1.2.19) has order $r = \min\{p, q\}$, where p is the order of the implicit linear multistep method and q is the order of the extrapolation formula. It can be noticed that this scheme will not suffer from order reduction since the error is not influenced by the Lipschitz constants in F_0 and F_1.

Stability results for IMEX methods are not easy to find, even for the simple test equation (1.2.16). Some examples have been discussed, particularly when $p = q = 2$. For example, when we use the explicit midpoint (Leap-Frog) method for the explicit part and the trapezoidal rule (Crank–Nicolson) for the implicit part, we have the IMEX-CNLF

$$w_{n+1} - w_{n-1} = 2\tau F_0(t_n, w_n) + \tau F_1(t_{n+1}, w_{n+1}) + \tau F_1(t_{n-1}, w_{n-1}),$$

where the stability region of the explicit method is $[-i, i]$ and the implicit method is A-stable.

The IMEX-RK Method

Several authors have already studied properties of IMEX-RK schemes for general PDEs. Recently in [43, 85], the IMEX-RK Method was thoroughly studied for the case of reaction diffusion equations.

Let us consider a pair of two Runge–Kutta methods defined by the arrays

$$\begin{array}{c|ccccc} 0 & 0 & 0 & & \cdots & 0 \\ c_2 & a_{21} & a_{22} & 0 & \cdots & 0 \\ c_3 & a_{31} & a_{32} & a_{33} & & \\ \vdots & \vdots & & & \ddots & \vdots \\ c_s & a_{s1} & & \cdots & a_{s,s-1} & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} \qquad \begin{array}{c|ccccc} 0 & 0 & 0 & & \cdots & 0 \\ c_2 & \widehat{a}_{21} & 0 & 0 & \cdots & 0 \\ c_3 & \widehat{a}_{31} & \widehat{a}_{32} & 0 & & \\ \vdots & \vdots & & & \ddots & \vdots \\ c_s & \widehat{a}_{s1} & & \cdots & \widehat{a}_{s,s-1} & 0 \\ \hline & \widehat{b}_1 & \widehat{b}_2 & \cdots & \widehat{b}_{s-1} & \widehat{b}_s \end{array}$$

with $c_i = \sum_{j=1}^{i} a_{ij} = \sum_{j=1}^{i-1} \widehat{a}_{ij}$, for $i = 1, \ldots, s$. The left formula determines a diagonally implicit (semi-implicit) Runge–Kutta method and the right formula is an explicit Runge–Kutta method. Consider this semi-discrete system

$$w'(t) = F_1(t, w(t)) + F_2(t, w(t)), \tag{1.2.21}$$

where F_1 takes care of the stiff term and F_2 takes care of the non-stiff term. By applying the left formula to F_1 and the right formula to F_2, we obtain the following scheme for the problem:

$$\begin{aligned} w_{n,i} &= w_n + \tau \sum_{j=1}^{i} a_{ij} F_1(t_n + c_j\tau, w_{nj}) + \tau \sum_{j=1}^{i-1} \widehat{a}_{ij} F_2(t_n + c_j\tau, w_{nj}), \\ w_{n+1} &= w_n + \tau \sum_{i=1}^{s} b_i F_1(t_n + c_i\tau, w_{ni}) + \tau \sum_{i=1}^{s} \widehat{b}_i F_2(t_n + c_i\tau, w_{ni}). \end{aligned} \tag{1.2.22}$$

The IMEX Euler scheme is the simplest example of a combination of implicit and explicit Euler methods

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 0 & 1 \\ \hline & 0 & 1 \end{array} \qquad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1 & 0 \end{array}$$

In this case, scheme (1.2.22) reduces to

$$w_{n+1} = w_n + \tau F_1(t_{n+1}, w_{n+1}) + \tau F_2(t_n, w_n). \tag{1.2.23}$$

Stability Consider the scalar test equation

$$u_t = \lambda u(t) + \mu u(t), \tag{1.2.24}$$

where both λ and μ are real numbers associated with the particular structure of reaction diffusion equations. An application of the IMEX scheme (1.2.22) to the test equation yields

$$\begin{aligned} U_n &= u_n e + \tau\lambda A U_n + \tau\mu \widehat{A} U_n, \\ u_{n+1} &= u_n + \tau\lambda b^T U_n + \tau\mu \widehat{b}^T U_n, \end{aligned} \tag{1.2.25}$$

where $e = [1, \ldots, 1]^T$ and $U_n \in \mathbb{C}^s$. By Cramer's rule, this implies

$$u_{n+1} = R(\alpha, \beta) u_n, \quad \alpha = \tau\lambda, \quad \beta = \tau\mu, \tag{1.2.26}$$

where $R(\alpha, \beta)$ is defined by

$$R(\alpha, \beta) = \frac{\det(I - \alpha A - \beta \widehat{A} + \alpha b^T e + \beta \widehat{b}^T e)}{\det(I - \alpha A)}. \tag{1.2.27}$$

In our case, the λs are real and we assume the μ is real too. To find the boundaries of the stability region, we will plot the curves

$$\begin{aligned} \det(I - \alpha A - \beta \widehat{A} + \alpha b^T e + \beta \widehat{b}^T e) - \det(I - \alpha A) = 0, \\ \det(I - \alpha A - \beta \widehat{A} + \alpha b^T e + \beta \widehat{b}^T e) + \det(I - \alpha A) = 0. \end{aligned}$$

For example, for the IMEX Euler scheme (1.2.23),

$$R(\alpha, \beta) = \frac{1 + \beta}{1 - \alpha},$$

and the stability boundaries will satisfy

$$\begin{aligned} 2 + \beta - \alpha = 0, \\ \beta + \alpha = 0. \end{aligned}$$

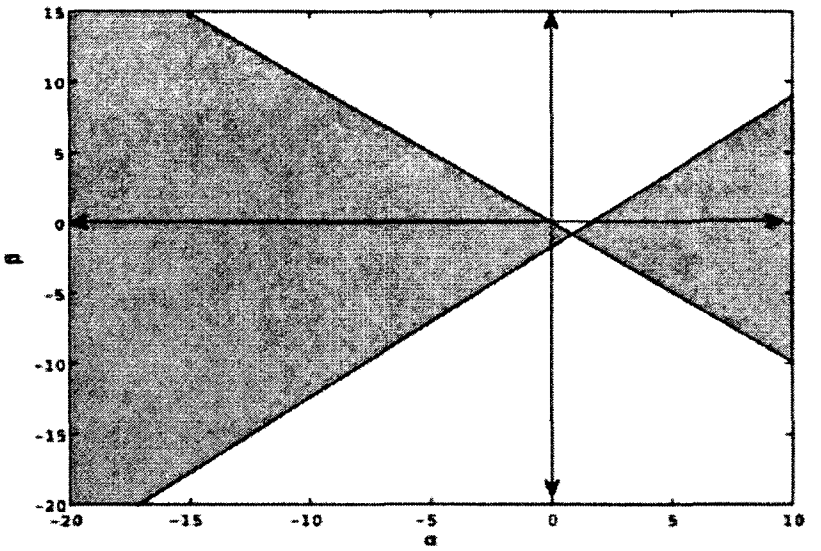

Figure 1.8: Stability regions (shaded) of IMEX Euler where $\alpha = \tau\lambda$ and $\beta = \tau\mu$.

From Figure 1.8, we observe that the IMEX Euler scheme generates a stable solution even for very large τ. Other competitive IMEX RK schemes have been constructed, see [43]. However, these schemes need to be tested to determine if they are useful for practical reaction diffusion problems.

1.2.5 Comparisons

We have discussed three classes of splitting methods, namely, LOD, ADI, and IMEX. Each one of these has different subclasses. The LOD-BE is good for parabolic problems and does not suffer from order reduction as does LOD-CN. The second class is the ADI methods. In contrast to the LOD methods, the intermediate stages of ADI methods yield approximations that are consistent with the full problem. On the other hand, the stability analysis of the ADI is more complicated than with the LOD. Also, one ADI method is restricted to two components, which is the ADI-PR.

The Douglas method is the other ADI method, which deals with more than two components and is unconditionally stable for the case of two components. It is to be preferred if the boundary conditions are important. Thirdly, the IMEX-θ method, the IMEX-multistep methods, and the IMEX-RK methods were discussed. Dealing with these mixtures will result in some CFL condition restrictions. In general, as in [38], the stability properties, when applying splitting methods to stiff systems, are quite poor.

1.3 Stabilized Runge–Kutta Methods

As we know, explicit methods are inefficient for stiff problems due to a strict stability condition. On the other hand, implicit methods tend to be very costly due to the associated algebraic system which need to be solved at each step. This was the motivation to find explicit methods with good stability properties. Such methods include stabilized explicit RK methods. These methods are based on explicit Runge–Kutta methods with extended stability domain along the negative real axis. Here we introduce Runge–Kutta–Chebyshev (RKC) method and briefly introduce Orthogonal-Runge–Kutta–Chebyshev (ROCK) method. These methods are suitable only for problems with real eigenvalues such as the discretizations of heat equations. Therefore, these methods are good candidates for the study of reaction diffusion equations. The detail of stability and convergence analysis can be founded in [39, 47, 86], but a brief overview is presented here.

The Runge–Kutta–Chebyshev (RKC) Family

A RKC method is an s-stage RK method designed for the explicit integration of stiff systems of ODEs originating from spatial discretization of PDEs. When we apply any explicit RK method to the test equation $u'(t) = \lambda u(t)$ $(\lambda \leq 0)$, the stability will be determined by its real stability boundary β, where $[-\beta, 0] \subset \mathcal{S} = \{z \in \mathbb{C}\colon |R(z)| \leq$

1}. The stability region $\mathcal{S}$ is determined by the stability polynomial

$$R(z) = \gamma_0 + \gamma_1 z + \gamma_2 z^2 + \cdots + \gamma_s z^s.$$

We need $\gamma_0 = \gamma_1 = 1$ to get 1st order consistency. Our goal is to choose the coefficient γ_i in order that S contains the interval $[-\beta, 0]$ of the negative real axis with β as large as possible.

First-order stability polynomials The optimal stability polynomial, which provides maximum β for given s, was obtained in [39] to be

$$P_s(z) = T_s\left(1 + \frac{z}{s^2}\right), \quad \beta = 2s^2, \tag{1.3.1}$$

which is the shifted Chebyshev polynomial of the first kind. Recall the Chebyshev polynomial T_s is defined by the relation $T_s(x) = \cos(s\cos^{-1}(x))$, $x \in [-1, 1]$, or for $z \in \mathbb{C}$ by

$$\begin{aligned} T_0(z) &= 1, \\ T_1(z) &= z, \\ T_j(z) &= 2zT_{j-1}(z) - T_{j-2}(z), \quad 2 \le j \le s. \end{aligned}$$

Also, $T_s(x)$ can be written as

$$T_s(x) = \sum_{i=0}^{s} \frac{(-s)_i (s)_i}{(\frac{1}{2})_i i!} \left(\frac{1-x}{2}\right), \tag{1.3.2}$$

where $(a)_i$ is defined as $(a)_0 = 1$ and $(a)_i = a(a+1)\ldots(a+i-1)$ for $a \in \mathbb{R}$, $i \ge 1$. For example, one can find that when $s = 2$,

$$P_2(z) = 1 + z + \frac{1}{8}z^2,$$

and when $s = 5$,

$$P_5(z) = 1 + z + \frac{4}{25}z^2 + \frac{28}{3125}z^3 + \frac{16}{78125}z^4 + \frac{16}{9765625}z^5.$$

In Figure 1.9, the stability region S was plotted for P_2 and P_5. We notice that in Figure 1.9, when $s = 5$, $\beta = 2 \times 5^2 = 50$ which matches the choice in (1.3.1).

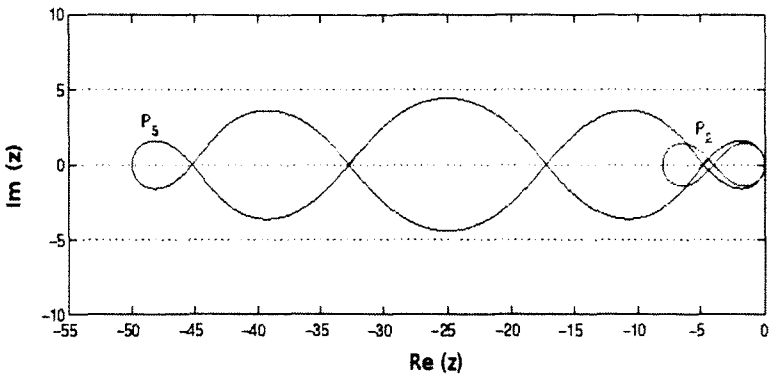

Figure 1.9: Stability regions of the first-order shifted Chebyshev polynomials P_2 and P_5.

Second-order Stability Polynomials Since the first-order accuracy is often too low, we look for a higher order. For second-order consistency, we consider

$$R(z) = 1 + z + \frac{1}{2}z^2 + \gamma_3 z^3 + \cdots + \gamma_s z^s, \tag{1.3.3}$$

where the free constant γ_i is chosen to obtain β as large as possible. Also, we can say that $R(z)$ approximates e^z for $z \to 0$, i.e., $R(z) = e^z + \mathcal{O}(z^3)$. One suitable polynomial was derived, in [8], as

$$B_s(z) = \frac{2}{3} + \frac{1}{3s^2} + \left(\frac{1}{3} - \frac{1}{3s^2}\right) T_s\left(1 + \frac{3z}{s^2 - 1}\right), \quad \beta \approx \frac{2}{3}(s^2 - 1). \tag{1.3.4}$$

Also, the coefficients of the stability polynomial B_s are found to be

$$\begin{aligned} \gamma_2 &= \frac{1}{2}, \\ \gamma_i &= 3\left(\frac{1 - \frac{(i-1)^2}{s^2}}{i(2i-1)(1 - \frac{1}{s^2})}\right)\gamma_{i-1}, \quad i = 3, \ldots, s. \end{aligned}$$

For example, one can find that, when $s = 2$,

$$B_2(z) = 1 + z + \frac{1}{2}z^2,$$

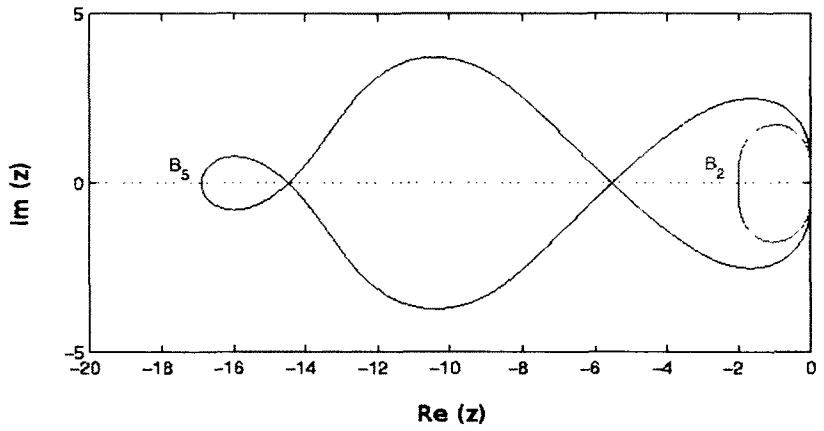

Figure 1.10: Stability regions of the second-order shifted Chebyshev polynomials B_2 and B_5.

and when $s = 5$,

$$B_5(z) = 1 + z + \frac{1}{2}z^2 + \frac{7}{80}z^3 + \frac{1}{160}z^4 + \frac{1}{6400}z^5.$$

In Figure 1.10, the stability region S was plotted for B_2 and B_5. Also, here we notice that in Figure 1.10, when $s = 5$, $\beta \approx \frac{2}{3}(5^2 - 1) = 16$.

Damped stability polynomials Note the stability regions in Figures 1.9 and 1.10 have nodal points where instability may result from a small imaginary perturbation on z. Therefore, it is very useful to introduce damping in order to avoid instability. Thus, the previous polynomials can be modified by introducing a little damping. One choice for damped polynomial, made by Guillou and Lago [31], is

$$P_s(z) = \frac{T_s(\omega_0 + \omega_1 z)}{T_s(\omega_0)}, \quad \omega_1 = \frac{T_s(\omega_0)}{T_s'(\omega_0)}. \tag{1.3.5}$$

The parameter $\omega_0 > 1$ is called the damping parameter and the parameter ω_1 will be chosen such that for any ω_0, we have $P_s'(0) = 1$ implying first-order consistency.

$P_s(z)$ now alternates between $T^{-1}(\omega_0)$ and $-T^{-1}(\omega_0)$ for $z \in [-\beta, 0]$ and the stability interval is determined by the relation $-\omega_0 \leq \omega_0 + \omega_1 z \leq \omega_0$, which gives $\beta = 2\omega_0/\omega_1$. A convenient choice for the damping parameter ω_0 is $\omega_0 = 1 + \epsilon/s^2$, where ϵ is a small positive number. Then, by using the derivative values $T_s'(1) = s^2$ and $T_s''(1) = \frac{1}{3}s^2(s^2-1)$, we get

$$T_s(\omega_0) \approx T_s(1) + (\omega_0 + 1)T_s'(\omega_0) = 1 + \epsilon,$$

and also

$$\begin{aligned} \beta &= \frac{2\omega_0}{\omega_1} \\ &= \frac{2\omega_0 T_s'(\omega_0)}{T_s(\omega_0)} \\ &\approx (2 - \frac{4}{3}\epsilon)s^2. \end{aligned} \tag{1.3.6}$$

A suitable choice for ϵ is 0.05. So, $\beta = (2 - \frac{4}{3} \times 0.05)s^2 \approx 1.93s^2$ which yields about 5% damping since $T_s^{-1} \approx 1-\epsilon$ with only a very small decrease in the stability interval. This is shown in Figure 1.11.

For the 2nd-order polynomial, let

$$B_s(z) = a + bT_s(\omega_0 + \omega_1 z), \tag{1.3.7}$$

where a, b, ω_0, and ω_1 are determined by the condition $|B_s(z)| \leq 1 - \epsilon$, $z \in [-\beta, 0]$, i.e.,

$$a + b = 1 - \epsilon, \quad a \geq 0, \quad \omega_0 - \omega_1 \beta = -1,$$

and the consistency conditions

$$B_s(0) = a + bT_s(\omega_0) = 1, \quad B_s'(0) = b\omega_1 T_s'(\omega_0) = 1, \quad B_s''(0) = b\omega_1^2 T_s''(\omega_0) = 1.$$

From these conditions, we have

$$B_s(z) = 1 + \frac{T_s''(\omega_0)}{(T_s'(\omega_0))^2}\left(T_s(\omega_0 + \omega_1 z) - T_s(\omega_0)\right), \tag{1.3.8}$$

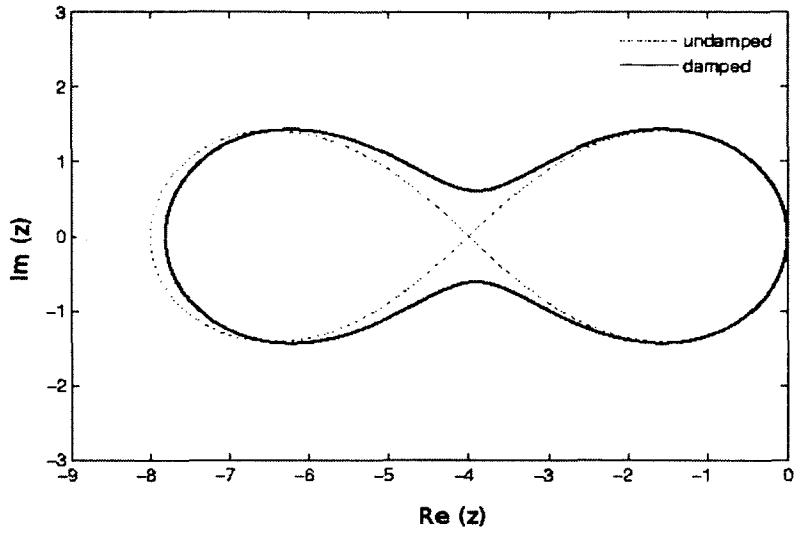

Figure 1.11: Stability regions of the first-order shifted Chebyshev polynomials P_2 in the damped and undamped cases.

where $w_1 = T_s'(w_0)/(T_s''(w_0))$. Now, using $T_s'(1) = s^2$, $T_s''(1) = \frac{1}{3}s^2(s^2-1)$, and $T_s'''(1) = \frac{1}{15}s^2(s^2-1)(s^2-4)$, then

$$\beta \approx \frac{(w_0+1)T_s''(w_0)}{T_s'(w_0)} \approx \frac{2}{3}(s^2-1)(1-\frac{2}{15}\epsilon).$$

By choosing $\epsilon = \frac{2}{13}$, we get approximately 5% damping in the interior of the stability interval and a reduction in the stability boundary of about 2% compared to the undamped case ($\epsilon = 0$).

Integration formulas Now by building a suitable stability polynomial, we can construct the Runge–Kutta formulas using these polynomials. But, one needs to be careful about what is happening internally. Consider the explicit RK

$$\begin{aligned} w_{n0} &= w_n, \\ w_{nj} &= w_n + \tau \sum_{k=0}^{j-1} \alpha_{jk} F(t_n + c_k\tau, w_{nk}), \quad j = 1, \ldots, s, \\ w_{n+1} &= w_{ns}, \end{aligned} \tag{1.3.9}$$

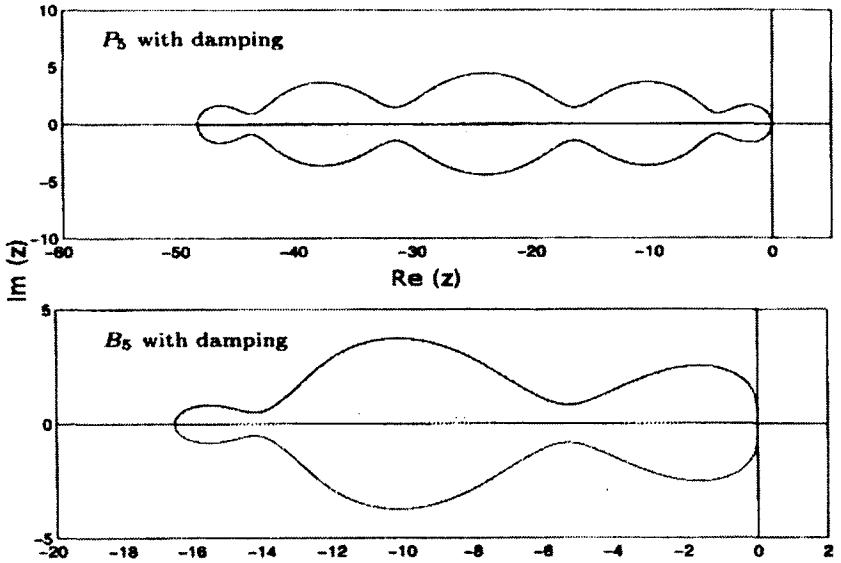

Figure 1.12: Stability regions of the first-order and the second-order shifted Chebyshev polynomials P_5 and B_5 damped cases.

and the perturbation

$$\begin{aligned} \tilde{w}_{n0} &= \tilde{w}_n, \\ \tilde{w}_{nj} &= \tilde{w}_n + \tau \sum_{k=0}^{j-1} \alpha_{jk} F(t_n + c_k\tau, \tilde{w}_{nk}) + r_j, \quad j = 1, \ldots, s, \\ \tilde{w}_{n+1} &= \tilde{w}_{ns}, \end{aligned} \tag{1.3.10}$$

where r_j is the local perturbation at the jth step, which can be the rounded-off error. Consider the linear system

$$w'(t) = Aw + g(t), \quad 0 \le t \le T, \quad w(0) = w_0,$$

and apply the perturbed scheme to this system. Using the notation $e_n = \tilde{w}_n - w_n$ and $e_{nj} = \tilde{w}_{nj} - w_{nj}$, we have

$$e_{nj} = R_j(\tau A)e_n + \sum_{k=1}^{j} Q_{jk}(\tau A)r_j, \quad 1 \le j \le s, \tag{1.3.11}$$

where $R_j(\tau A)$ is a matrix polynomial of degree j and $Q_{jk}(\tau A)$ is a matrix polynomial of degree $j-k$, and

$$\begin{aligned} e_{n+1} &= e_{ns} \\ &= R(\tau A)e_n + \sum_{j=1}^{s} Q_{sj}(\tau A)r_j. \end{aligned} \tag{1.3.12}$$

The polynomials Q_{sj} in (1.3.12) are called internal stability polynomials. If the matrix A is normal, then

$$\|e_{n+1}\| \le \max_{z=\tau\lambda} |R(z)|\|e_n\| + \sum_{j=1}^{s} \max_{z=\tau\lambda} |Q_{sj}(z)|\|r_j\|, \tag{1.3.13}$$

where λ represents the eigenvalues of A. Moreover, if $|R(z)| \le 1$ then we have stability, although these conditions will not guarantee internal stability. To avoid internal instability, a family of methods was derived in [36]. The idea is to control the stability polynomials Q_{sj} as z approaches the stability region and to determine the formulas such that all R_j in (1.3.11) are defined by the three-term Chebyshev recursion (1.3.2) and share the same stability region. So, the ansatz is made that all R_j $(1 \le j \le s)$ are of the form

$$R_j = a_j + b_j T_j(\omega_0 + \omega_1 z), \quad a_j = 1 - b_j T_j(\omega_0), \tag{1.3.14}$$

where $R_s(z) = R(z)$, which means that ω_0, ω_1, and b_s are defined as before. Therefore, b_j only needs to be defined for $1 \le j \le s$. Define $R_0(z) = a_0 + b_0 \equiv 1$ and make use of the three-term Chebyshev recursion (1.3.2) and $R_j(0) = 1$, to get

$$\begin{aligned} R_0(z) &= 1, \quad R_1(z) = 1 + \tilde{\mu}_1 z, \\ R_j(z) &= (1 - \mu_j - \nu_j)R_0 + \mu_j R_{j-1} + \nu_j R_{j-2} + \tilde{\mu}_j z R_{j-1} + \tilde{\gamma}_j z R_0, \quad \text{where } 2 \le j \le s, \\ \tilde{\mu}_1 &= b_1\omega_1, \quad \mu_j = \frac{2b_j\omega_0}{b_{j-1}}, \quad \nu_j = \frac{-b_j}{b_{j-2}}, \quad \tilde{\mu}_j = \frac{2b_j\omega_1}{b_{j-1}}, \quad \tilde{\gamma}_j = -a_{j-1}\tilde{\mu}_j. \end{aligned} \tag{1.3.15}$$

Based on these relations, we can introduce RKC for the nonlinear problem $w' = F(t, w(t))$ by associating R_j with the intermediate approximation w_{nj} and the occurrence of z with a function evaluation. For example, if $j = 1$, to extract the first step

of the scheme from the stability polynomial R_1, we put

$$w_{n+1} = R_1(z)w_n = (1 + \tilde{\mu}_1 z)w_n = w_n + \tilde{\mu}_1 z w_{n0} = w_n + \tilde{\mu}_1 \tau F_{n0}.$$

By doing the same for all j's, the RKC would be as follows

$$\begin{aligned} w_{n0} &= w_n, \\ w_{n1} &= w_n + \tilde{\mu}_1 \tau F_{n0}, \\ w_{nj} &= (1 - \mu_j - \nu_j)w_n + \mu_j w_{n,j-1} + \nu_j w_{n,j-2} + \tilde{\mu}_j \tau F_{n,j-1} + \tilde{\gamma}_j \tau F_{n0}, \\ w_{n+1} &= w_{ns}, \quad 2 \le j \le s. \end{aligned} \tag{1.3.16}$$

The function F_{nk} denotes $F(t_n + c_k\tau, w_{nk})$. Therefore, (1.3.16) is similar to (1.3.9). Note that, at this point, b_j $(1 \le j \le s)$ remain undefined. We will introduce selection criteria for b_j based on the desired order. For a first-order damped polynomial, to obtain (1.3.5) from the general form (1.3.14), a little work shows that b_j should be

$$b_j = \frac{1}{T_j(\omega_0)}, \quad j = 0, 1, \ldots, s. \tag{1.3.17}$$

We notice that $R_j(z) = e^{c_j z} + \mathcal{O}(z^2)$, with

$$c_j = b_j \omega_1 T_j'(\omega_0) = \frac{T_s(\omega_0)}{T_s'(\omega_0)} \frac{T_j'(\omega_0)}{T_j(\omega_0)} \approx \frac{j^2}{s^2},$$

where $1 \le j \le s-1$ and $c_s = 1$. For a second-order damped polynomial to obtain (1.3.8) from the general form (1.3.14), one of the choices for b_j would be

$$b_j = \frac{T_j''(\omega_0)}{(T_j'(\omega_0))^2}, \quad j = 2, \ldots, s, \tag{1.3.18}$$

and we pick $b_0 = b_1 = b_2$. We also notice that $R_j(z) = e^{c_j z} + \mathcal{O}(z^3)$ with

$$c_j = \frac{T_s'(\omega_0)}{T_s''(\omega_0)} \frac{T_j''(\omega_0)}{T_j(\omega_0)'} \approx \frac{j^2 - 1}{s^2 - 1},$$

where $2 \le j \le s-1$ and $c_s = 1$. These choices will define the parameters needed for applying stability polynomials of orders 1 and 2.

Internal stability As we said before, to avoid internal instability we have to pay attention to the polynomials Q_{jk}. By using (1.3.11), (1.3.15) and (1.3.16), one will find out that the polynomials Q_{jk} satisfy

$$\begin{aligned} Q_{jk}(z) &= 2\frac{b_j}{b_{j-1}}(\omega_0+\omega_1 z)Q_{j-1,k}(z) - \frac{b_j}{b_{j-2}}Q_{j-2,k}(z), \\ Q_{k,k}(z) &= 1, \\ Q_{k+1,k}(z) &= 2\frac{b_{k+1}}{b_k}(\omega_0+\omega_1 z), \quad 1 \le k \le s-2 \quad k+2 \le j \le s, \\ Q_{s-1,s-1}(z) &= Q_{s,s}(z) = 1, \\ Q_{s,s-1}(z) &= 2\frac{b_s}{b_{s-1}}(\omega_0+\omega_1 z). \end{aligned} \tag{1.3.19}$$

It follows that $b_j^{-1}Q_{jk}(z)$ satisfies the recursion for the shifted Chebyshev polynomial of the second kind, due to the factor 2 occurring in the definition of the second starting value. Then, we will have the following equation

$$Q_{sj} = \frac{b_s}{b_j}U_{s-j}(\omega_0+\omega_1 z), \quad j = 0,\ldots,s, \tag{1.3.20}$$

where $U_i(z)$ is the ith Chebyshev polynomial of the second kind. For $z \in [-\beta, 0]$, it can be shown that

$$|Q_{sj}(z)| \le \frac{b_s}{b_j}(s-j+1)(1+C\epsilon), \quad j = 1,\ldots,s, \tag{1.3.21}$$

where C is a constant independent of s. So, if A is negative definite and the stability condition $\tau\lambda \in [-\beta, 0]$ ($\lambda \in \sigma(A)$) is satisfied, where $\sigma(A)$ is the spectral radius of A, the error bound will be

$$\|e_{n+1}\| \le \|e_n\| + \sum_{j=1}^{s}\frac{b_s}{b_j}(s-j+1)(1+C\epsilon)\|r_j\|. \tag{1.3.22}$$

By considering both cases with and without damping for RKC1 and RKC2, the bound (1.3.22) can be simplified to

$$\|e_{n+1}\| \le \|e_n\| + \tilde{C}\sum_{j=1}^{s}(s-j+1)\|r_j\|,$$

$$\leq \|e_n\| + \frac{1}{2}s(s+1)\tilde{C}\max_j \|r_j\|, \tag{1.3.23}$$

where $\tilde{C}$ is again a constant independent of s, A, and τ. Therefore, within one integration step the accumulation of internal perturbations such as round off errors is independent of the spectrum of A as long as $\tau\lambda \in [-\beta, 0]$.

Convergence Consider a semi-discrete linear PDE problem of type

$$\frac{d}{dt}u_h(t) = F(t, u_h(t)) + \alpha_h(t), \quad 0 \leq t \leq T, \tag{1.3.24}$$

where $u_h(t_n)$ is the exact solution of the PDE restricted to a space grid and $\alpha_h(t)$ the local space truncation error on this grid. Let $\varepsilon_n = ((u_h(t_n) - u(t_n)) + (u(t_n) - w_n)) = u_h(t_n) - w_n$ be the global error on this grid and r_j $(1 \leq j \leq s)$ the stage truncation errors obtained by substituting u_h into the RKC method (1.3.16) with parameters from (1.3.15). That is,

$$\begin{aligned} u_h(t_n + c_1\tau) &= u_h(t_n) + \tilde{\mu}\tau F_0 + r_1, \\ u_h(t_n + c_j\tau) &= (1 - \mu_j - \nu_j u_h(t_n) + \mu_j u_h(t_n + c_{j-1}\tau) + \nu_j u_h(t_n + c_{j-2}\tau) \\ &\quad + \tilde{\mu}_j\tau F_{j-1} + \tilde{\gamma}_j\tau F_0 + r_j, \quad j = 2, \ldots, s. \end{aligned} \tag{1.3.25}$$

since this equation can be seen as a perturbed RKC method. Thus, bounds for the global error can be obtained by estimating r_j and using internal stability results. Let us illustrate this for the undamped $(\epsilon = 0)$ first-order scheme RKC1 defined by

$$\tilde{\mu}_1 = \frac{1}{s^2}, \quad \mu_j = 2, \quad \tilde{\mu}_j = \frac{2}{s^2}, \quad \tilde{\gamma}_j = 0, \quad 2 \leq j \leq s. \tag{1.3.26}$$

Let $u_h \in C^2[0, T]$. From (1.3.25) and the Taylor series expansion of u_h and u_h' at the intermediate step points $t_n + c_{j-1}\tau$ (see the details in [86]), it follows that

$$r_j = \tau^2\rho_j + \tau\tilde{\mu}_j\alpha_h(t_n + c_{j-1}\tau), \quad j = 1, \ldots, s, \tag{1.3.27}$$

where the remainder terms ρ_j are given by

$$\rho_1 = \frac{1}{2}c_1^2 u_h^{(2)}(t_n),$$

$$\rho_j = \frac{1}{2}(c_j - c_{j-1})^2 u_h^{(2)}(t_*) + \frac{1}{2}(c_{j-2} - c_{j-1})^2 u_h^{(2)}(t_*), \quad j = 2, \ldots, s, \tag{1.3.28}$$

with t_* denoting some point in $[t_n, t_{n+1}]$. From $c_j = j^2/s^2$, we have

$$c_j - c_{j-1} = s^{-2}(2j-1) \leq 2s^{-1}, \quad j = 1, \ldots, s. \tag{1.3.29}$$

Thus, we have the following bounds for r_j

$$\|r_j\| \leq \frac{4\tau}{s^2}\left(\tau \max_{t_n \leq t \leq t_{n+1}} \|u_h{}^{(2)}(t)\| + \frac{1}{2} \max_{t_n \leq t \leq t_{n+1}} \|\alpha_h(t)\|\right), \quad j = 1, \ldots, s. \tag{1.3.30}$$

Inserting these bounds into (1.3.23) and adding up yields

$$\|\varepsilon_n\| \leq C\left(\tau \max_{0 \leq t \leq T} \|u_h{}^{(2)}(t)\| + \max_{0 \leq t \leq T} \|\alpha_h(t)\|\right), \quad n = 1, 2, \ldots, n\tau \leq T, \tag{1.3.31}$$

where C is a constant independent of s, A, and τ. Therefore, irrespective of s or $\tau\sigma(A)$, the RKC1 scheme is convergent. A similar result for RKC2 can be found in [86].

The Orthogonal-Runge–Kutta–Chebyshev (ROCK) Family

The ROCK methods have been recently proposed and have a common property with RKC methods which use stable two-step (three-term) recurrence formulas for internal stability. On the other hand, the ROCK methods have an extended stability interval and are not given in a closed form. The construction of ROCK is very lengthy. Up to now, there are two types of ROCK methods of order two and order four. For details we refer to [2, 7]. However, some of the details of the analytical results are not known [39]. Thus, ROCK is not as established as RKC, and needs to be further developed.

1.3.1 Comparisons

Both RKC and ROCK methods are suitable for reaction diffusion equation types when dealing with negative real eigenvalues. Also, they make use of three-term recurrence formulas for the construction of stability polynomials. Moreover, the stability interval

in both schemes depends on the square of the number of stages. The ROCK methods are not given in closed form. The internal stability in the case of RKC is well studied. This is not the case with the ROCK due to its recent derivation. From the stability point view, β_s in RKC2 behaves like $\beta_s \approx 0.65s^2$. If we increase the number of stages s we get a larger stability domain. However, this will involve additional computation of extra variables. So in the end, determining which scheme is most efficient varies on a case by case basis.

1.4 ELP Schemes for Stiff Systems

We consider a number of time-discretization schemes for solving stiff reaction diffusion systems. These methods solve exactly the linear part (ELP schemes) of the resulting ODEs from spacial discretization of reaction diffusion systems. Due to the inefficiency of integration factor (IF) [22, 57] and exponential time differencing (ETD) [24, 26, 27] methods when applied to stiff reaction diffusion equations, many semi-implicit schemes were developed. In these schemes, the linear diffusion will be treated exactly and explicitly, and the nonlinear reactions implicitly. A remarkable feature is the decoupling between the exact evaluation of the diffusion terms and the implicit treatment of the nonlinear reaction terms. We will introduce an implicit integration factor Scheme (IIF), an exponential time differencing Scheme (ETD), and a few combinations between IIF and ETD. Also, modifications for each scheme are mentioned. Consider the reaction diffusion system

$$\frac{\partial u}{\partial t} = \bar{\alpha}\triangle u + F(u), \tag{1.4.1}$$

where $u \in \mathbb{R}^m$, the diffusion matrix $\bar{\alpha} \in \mathbb{R}^{m\times m}$, and the reaction term $F(u) \in \mathbb{R}^m$. By discretizing the space variables of (1.4.1), we have the following semi-discrete ODEs

$$u_t = \mathcal{C}u + \mathcal{F}(u), \tag{1.4.2}$$

and let N denote the number of spatial grid points for the approximation of the Laplacian. Then $u(t) \in \mathbb{R}^{N\times m}$ and $\mathcal{C}$ is an $Nm \times Nm$ matrix. For example, in a one-dimensional system with one diffusion term, $\mathcal{C}$ is a tri-diagonal matrix when a second-order central difference is used.

1.4.1 Implicit Integration Factor (IIF)

Let us consider a scalar case of the semi-discrete system (1.4.2) of the form

$$u_t = cu + f(u), \quad t \geq 0, \quad u(0) = u_0, \tag{1.4.3}$$

where c is a constant representing the diffusion and f is a nonlinear function representing the reaction. Multiplying (1.4.3) by the integration factor e^{-ct} and integrating the equation over one time step from t_n to $t_{n+1} = t_n + h$, we obtain

$$u(t_{n+1}) = u(t_n)e^{ch} + e^{ch}\int_0^h e^{-c\tau} f(u(t_n + \tau))d\tau. \tag{1.4.4}$$

Now we approximate the integrand $e^{-c\tau} f(u(t_n + \tau))$. In order to get r-th order accuracy, we use an $(r-1)$-th degree Lagrange polynomial, $p(\tau)$, with interpolation points at $t_{n+1}, t_n, \ldots, t_{n+2-r}$, where $n \geq r-2$. Then, we have

$$p(\tau) = \sum_{i=-1}^{r-2} e^{ich} f(u_{n-i}) \prod_{j=-1, i\neq j}^{r-2} \frac{\tau + jh}{(j-i)h}, \quad \text{where} \quad 0 \leq \tau \leq h. \tag{1.4.5}$$

Then integrate the polynomial $p(\tau)$ with respect to τ to get

$$u_{n+1} = e^{ch}u_n + h\left(\alpha_{n+1} f(u_{n+1}) + \sum_{i=0}^{r-2} \alpha_{n-i} f(u_{n-i})\right), \tag{1.4.6}$$

with α_{n+1}, α_n, α_{n-1}, $\ldots$, α_{n-r+2} defined as

$$\alpha_{n-i} = \frac{e^{(i+1)ch}}{h}\int_0^h \prod_{j=-1, i\neq j}^{r-2} \frac{\tau + jh}{(j-i)h} d\tau, \tag{1.4.7}$$

where $-1 \leq i \leq r-2$. Denoting u_n the approximation of $u(t_n)$ and f_n the approximation of $f(u(t_n))$, the second-order scheme IIF2 ($r = 2$) has the form

$$u_{n+1} = e^{ch}\left(u_n + \frac{h}{2}f(u_n)\right) + \frac{h}{2}f(u_{n+1}), \tag{1.4.8}$$

or simply

$$u_{n+1} = e^{ch}u_n + \frac{h}{2}f_{n+1} + \frac{he^{ch}}{2}f_n,$$

with local truncation error

$$-\frac{1}{12}(c^2 f_n - 2cf'_n + f''_n)h^3.$$

Here we see one of the important features of the scheme, that is, α_{n+1} is independent of e^{ch}, which will lead us to the following convenient form

$$u_{n+1} = h\alpha_{n+1}f(u_{n+1}) + \text{known quantities}.$$

Also, IIF schemes do not involve any calculation of c^{-1}, which in case of higher dimension represents the inverse of a matrix. On the other hand, one of the weaknesses of the scheme is the presence of c in the error. This means that if we have a large c, we will have a large error.

For stability, applying (1.4.8) to $u_t = -qu + du$ with $q \geq 0$ and then substituting $u_n = e^{in\theta}$ into the resulting equation, we obtain

$$e^{i\theta} = e^{-qh}\left(1 + \frac{1}{2}\lambda\right) + \frac{1}{2}\lambda e^{i\theta},$$

or simply

$$\lambda = 2\frac{e^{i\theta} - e^{-qh}}{e^{-qh} + e^{i\theta}},$$

where $\lambda = dh$ has a real part λ_r and imaginary part λ_i. Therefore, the equations for λ_r and λ_i are

$$\lambda_r = \frac{2(1 - e^{-2qh})}{(1 - e^{-qh})^2 + 2(1 + \cos\theta)e^{-qh}},$$

$$\lambda_i = \frac{4(\sin\theta)e^{-qh}}{(1 - e^{-qh})^2 + 2(1 + \cos\theta)e^{-qh}}.$$

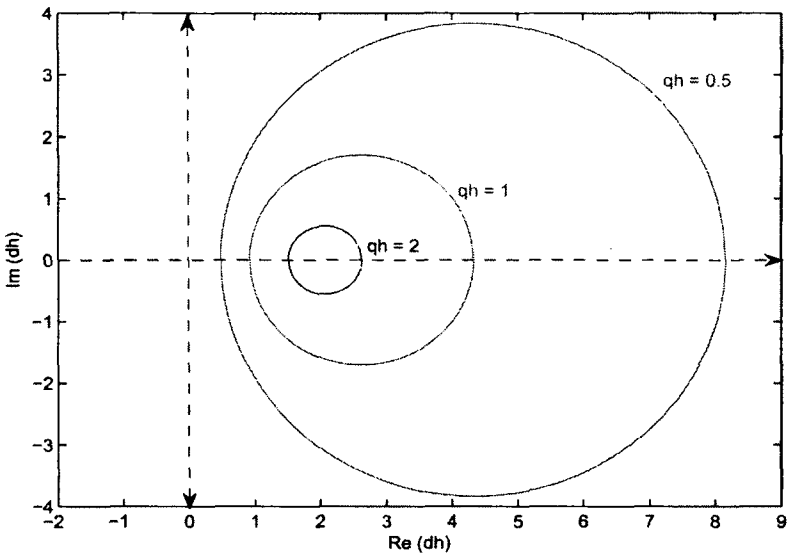

Figure 1.13: Stability regions (exterior of the closed curves) for the second order IIF with $qh = 0.5,\ 1,\ 2$.

Since $q \geq 0$, we have $\lambda_r \geq 0$ for $0 \leq \theta \leq 2\pi$. Then, IIF2 is A-stable since the stability region includes the complex plane for all λ with $\lambda_r \leq 0$. The stability regions were plotted in Figure 1.13 for different values of qh. It can be noticed that as q approaches 0 the stability region coincides with the domain $\lambda_r \leq 0$, whereas in the limit as q approaches ∞, the stability region approaches the entire complex plane excluding the point $(2, 0)$. For IIF3, as in [57], the scheme is no longer A-stable. Compared to EIF methods and other ETD schemes, the IIF method has excellent stability properties and the second-order IIF (IIF2) is unconditionally stable [22, 57].

1.4.2 Exponential Time Differencing (ETD) Scheme

ETD schemes are particularly well suited to Fourier spectral methods, which tend to have diagonal linear parts. For deriving the an ETD scheme, we consider as before (1.4.3) and then multiplying with an integration factor and integrating from t_n to

t_{n+1}, we will have

$$u(t_{n+1}) = u(t_n)e^{ch} + e^{ch}\int_0^h e^{-c\tau} f(u(t_n+\tau))d\tau. \tag{1.4.9}$$

In the derivation of ETD schemes, the integrand is approximated first through interpolation polynomials of the function $f(u(t_n+\tau))$ where we leave $e^{-c\tau}$ unchanged. The simplest approximation to the integral in (1.4.9) is that f is constant ($f = f_n + \mathcal{O}(h)$) between t_n and t_{n+1}. Then, (1.4.9) becomes the ETD1 scheme which is given by

$$u_{n+1} = e^{ch}u_n + \frac{f_n(e^{ch}-1)}{c}, \tag{1.4.10}$$

which has a local truncation error $h^2 f'/2$. It is seen that for small $|c|$ (1.4.10) will approach the forward Euler method. Now we use higher order approximation for f. Let us use

$$f = f_n + \tau\left(\frac{f_n - f_{n-1}}{h}\right) + \mathcal{O}(h^2).$$

So, the ETD2 scheme would be

$$u_{n+1} = e^{ch}u_n + f_n\frac{(1+hc)e^{ch} - 1 - 2hc}{hc^2} + f_{n-1}\frac{-e^{ch}+1+hc}{hc^2}, \tag{1.4.11}$$

which has a local truncation error of $5h^3 f''/12$. Also, we can use interpolation points to get an implicit scheme, which gives a much larger stability region for stiff reactions compared to the explicit ETD (IETD2) schemes. For example, as in [57], the second order implicit ETD scheme is of the form

$$u_{n+1} = e^{ch}u_n + f_{n+1}\frac{e^{ch}-1}{2c} + f_n\frac{e^{ch}-1}{2c}. \tag{1.4.12}$$

Explicit and implicit ETD schemes for high order have been derived in [11] and a more straightforward derivation of the explicit methods, based on a polynomial approximation of the integrand in (1.4.9), was introduced in [24]. Therefore, the general form of ETD of order m is

$$u_{n+1} = e^{ch}u_n + h\sum_{m=0}^{s-1} g_m \sum_{k=0}^{m}(-1)^k \binom{m}{k} f_{n-k}, \tag{1.4.13}$$

where $g_0 = \frac{e^{ch}-1}{ch}$ and

$$g_{m+1} = \frac{\sum_{k=0}^{m} \frac{g_k}{m+1-k} - 1}{ch}, \quad m \geq 0.$$

For the stability of ETD2, we consider the nonlinear autonomous ODE

$$u_t = cu + f(u),$$

and use the test equation

$$u_t = cu + \lambda u,$$

where both c and λ are complex. Thus, the stability region for these methods is four dimensional. Some cases have been discussed for different values of c and λ. For example, in [11], λ is complex and c is real or both c and λ are purely imaginary. Here, we introduce the case where both c and λ are real. Applying ETD2 (1.4.11) to the test equation, we get

$$y^2 r^2 - \{y^2 e^y + x[(1+y)e^y - 1 - 2y]\}r + x(e^y - 1 - y) = 0, \tag{1.4.14}$$

where $r = u_{n+1}/u_n$, $x = \lambda h$, and $y = ch$. The stability boundaries, when $|r| = 1$, are the lines

$$x + y = 0$$

and

$$x = \frac{-y^2(1+e^y)}{ye^y + 2e^y - 2 - 3y}.$$

The stability region is plotted in the x, y plane in Figure 1.14. Now, let us introduce an alternate way of presenting the stability regions that can be shown in the complex plane. Using the same notation as in the last section, where we apply the scheme to $u_t = -qu + du$ with $q \geq 0$, and then substituting $u_n = e^{in\theta}$ into the resulting equation, we obtain

$$\lambda = -qh\frac{e^{2i\theta}}{1 - 2e^{i\theta}},$$

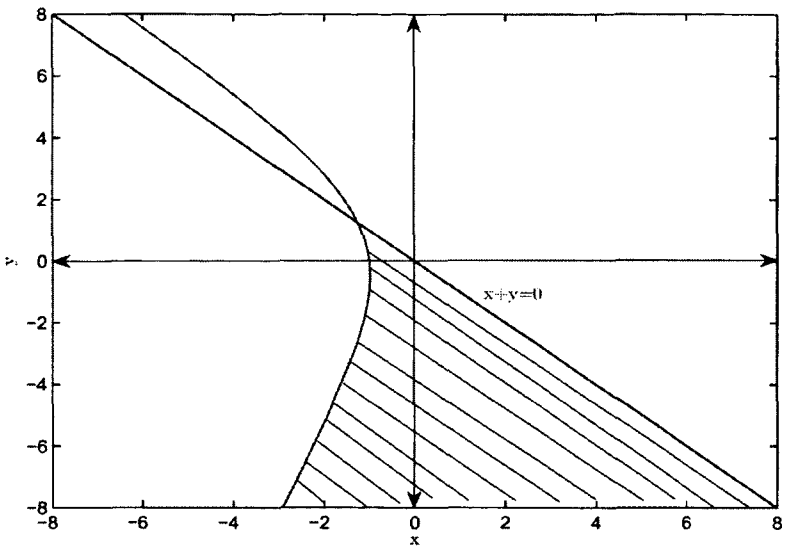

Figure 1.14: Stability regions (shaded) in the x, y plane for Exponential Time Differencing.

where $\lambda = dh$ has real part λ_r and imaginary part λ_i. Therefore, the equations for λ_r and λ_i are

$$\lambda_r = -qh\frac{\cos 2\theta - 2\cos\theta}{5 - 4\cos\theta},$$

$$\lambda_i = -qh\frac{\sin 2\theta - 2\sin\theta}{5 - 4\cos\theta}.$$

The stability regions are plotted in Figure 1.15 for different values of qh. More results for the stability of ETD are discussed in [24, 26, 27]. In addition, exponential time differencing Runge–Kutta (ETDRK) [24, 45] methods are a set of ETD methods based on Runge–Kutta time-stepping. ETDRK can be carried out efficiently if c (in the case of a matrix) is easily diagonalizable.

1.4.3 Comparisons

The difference between IIF and ETD schemes is that the ETD schemes are obtained when $f(u(t_n + \tau))$ is approximated, whereas the IIF method is obtained when the

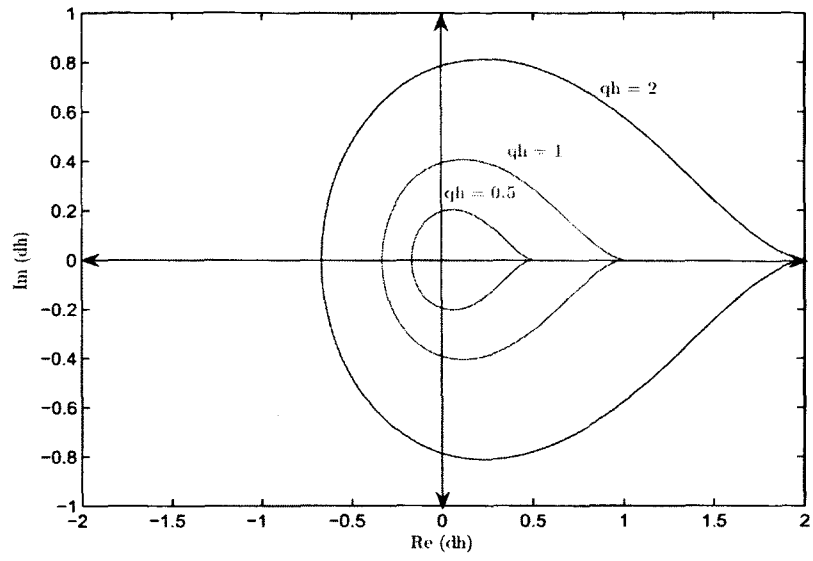

Figure 1.15: Stability regions (interior of the closed curves) for second-order ETD.

whole integrand $e^{-c\tau}f(u(t_n+\tau))$ is approximated. Both schemes share the feature of decoupling between the exact evaluation of the diffusion terms and implicit treatment of the nonlinear reaction terms. Moreover, they require the calculation of the exponential matrix e^{ch}, which is nontrivial when c is non-diagonal. However this calculation would only be required once. The IIF schemes do not involve any calculation of c^{-1}. On the other hand, when the matrix c has one or more zero eigenvalues, the method ETD can not be used. From a stability point of view, Figures 1.13 and 1.15 show that IIF2 is far superior to the ETD2 in both cases, whether $|c|$ is small or large. Moreover, the IIF2 scheme is unconditionally stable. Nevertheless, the errors in IIF are greater than those of ETD by a factor of order c^2. Finally, one can choose the suitable scheme based on the type of spacial discretization with effect on c, the stiffness of the system, and the required accuracy.

1.5 Numerical Examples of Nagumo Reaction Diffusion Equation

In this section, we test some of the schemes illustrated in previous sections and compare them from different perspectives. We introduce different schemes where each one represents a different class. Such classes include multistep methods, splitting methods, RK stabilized methods, and ELP methods of order two.

We consider a common nonlinear reaction diffusion example which is the Nagumo equation

$$u_t = \alpha u_{xx} + R(u - a_1)(a_2 - u)(u - a_3), \tag{1.5.1}$$

with a Gaussian initial condition. To provide more comparison details, different values of the diffusion and reaction coefficients are examined.

In the first case, non-stiff diffusion and non-stiff reaction coefficients are considered. As shown in Figure 1.16, Taylor2 and the RK class, RK2 and RKC2, demonstrated the best results from the error and computational time sides. Both AB2 and CNAB2 provided good performance, despite the fact that they had a high computational time.

The second case consists of the stiff diffusion case a with non-stiff reaction term which is presented in Figure 1.17. Figure 1.17 shows the extended stability for RKC2 compared to RK2. However RK2 is faster than RKC2. Also, CNAB2, AB2, and ETD2 performed similarly. From the computational side, ETD2 is the most expensive since it involves matrix inversion.

Finally, third case, where the diffusion coefficient is non-stiff and the reaction term is stiff, is shown in Figure 1.18. This case is similar to the non-stiff case. Thus similar conclusions can be stated here. Likewise, the resulting Figure 1.19, when both operators are stiff, is similar to Figure 1.17. These similarities in performance are due to the chosen parameter values in the example that make the reaction term

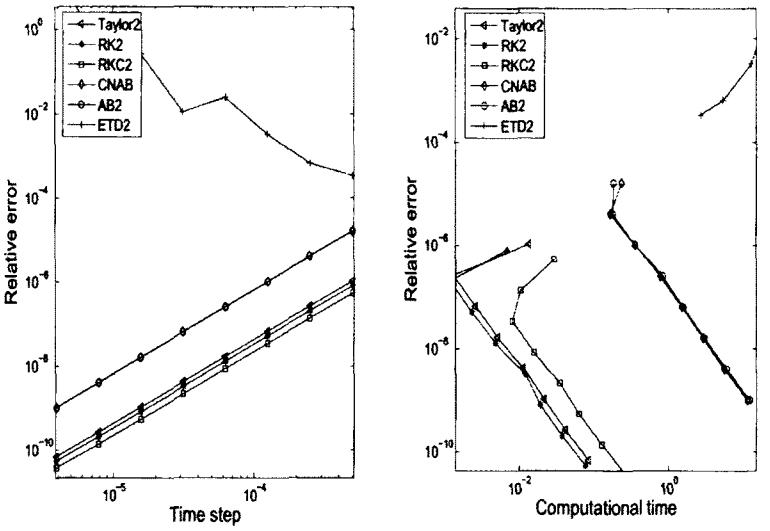

Figure 1.16: Solving Nagumo equation with a Gaussian initial condition and $\frac{\alpha}{dx^2} = 1$, $R = 1$, $t_0 = 0$, $t_f = 0.1$, $dx = 0.05$, $a_1 = 0$, $a_2 = 0.1$, $a_3 = 10$.

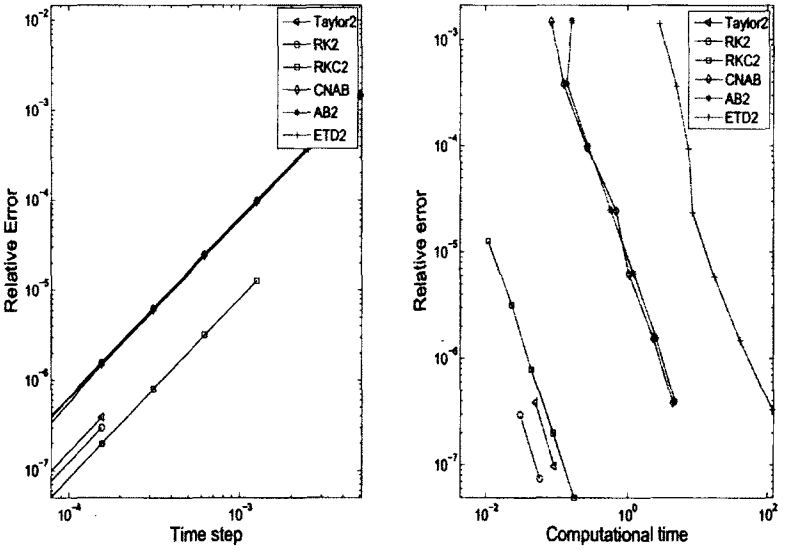

Figure 1.17: Solving Nagumo equation with a Gaussian initial condition and $\frac{\alpha}{dx^2} = 2000$, $R = 1$, $t_0 = 0$, $t_f = 0.1$, $dx = 0.05$, $a_1 = 0$, $a_2 = 0.1$, $a_3 = 10$.

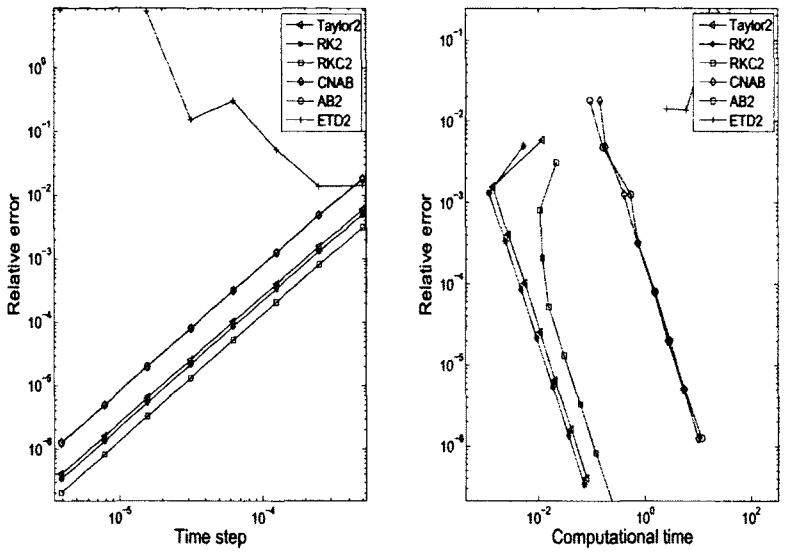

Figure 1.18: Solving Nagumo equation with a Gaussian initial condition and $\frac{\alpha}{dx^2} = 1$, $R = 10$, $t_0 = 0$, $t_f = 0.1$, $dx = 0.05$, $a_1 = 0$, $a_2 = 0.1$, $a_3 = 10$.

less dominant than the diffusion operator.

1.6 Summary and Conclusion

In this chapter, we studied numerical methods for reaction diffusion equations. We started by introducing basic definitions and preliminaries. Then, a large number of approaches were discussed.

In the first section of this chapter, the well-known RK family was discussed. Also, the linear multistep methods (LMM) were presented. Splitting methods, LOD, ADI, and IMEX were briefly introduced in Section 1.2. A recently developed method, the stabilized RK method, was introduced in Section 1.3. In Section 1.4, new modified schemes, ELP methods, were presented. These methods include exponential time differencing (ETD) and integration factor (IF) schemes. Lastly, in Section 1.5, a common model, the Nagumo equation, was used to test the performance of these numerical methods. Different types of stiffness for this equation were solved. For

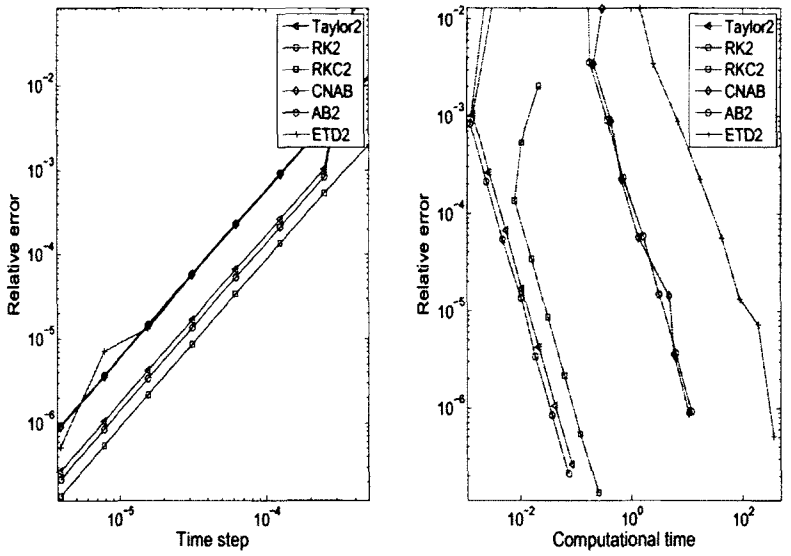

Figure 1.19: Solving Nagumo equation with a Gaussian initial condition and $\frac{\alpha}{dx^2} = 2000$, $R = 10$, $t_0 = 0$, $t_f = 0.1$, $dx = 0.05$, $a_1 = 0$, $a_2 = 0.1$, $a_3 = 10$.

cases with stiff diffusion, stiff reaction, or both, RK2 shows good performance.

Based on the analysis in this chapter and the numerical examples we suggest that in general, both RK and RKC show a better performance for both stiff or non-stiff regimes. Splitting methods are good, however, they have high computational cost. In addition, both ETD and IIF are very expensive since both involve matrix inversion. In conclusion, the RK family is the most suitable for solving reaction diffusion equations, combined possibly with splitting methods.

We now turn to a class of schemes that is simpler and less expensive per time-step than other explicit schemes, namely one-dimensional Padé approximations, which make use of rational functions. This approach was first developed in [5], where amongst other examples the Fisher equation was studied with different schemes. Two Padé time stepping (PTS) schemes performed much better than RK4 and RKC2 in the presence of Fourier spectral discretization. This suggests that studying our reaction diffusion equation with PTS schemes and then implementing these schemes in

the stiff regime may lead to efficient solvers. Also, we will search for modifications to PTS schemes that can provide better performance for our problem. Therefore, discussion, analysis, and modification of the PTS scheme will be investigated in the following chapters.

Finally, the main contributions of my thesis are summarized here:

- we demonstrated the need for control of PTS,
- we determined a local error control threshold for the PTS scheme,
- we established and compared three approaches based on PTS scheme, and
- we developed schemes based on splitting methods which provide a good balance of stability and efficiency.

Chapter 2

Padé Time Stepping (PTS) on the Diffusion Operator

As discussed in the previous chapter, it can be advantageous to solve the linear diffusion operator with an implicit scheme given the availability of unbounded stability region. However, such an approach is computationally costly. This factor also served as motivation to develop an explicit scheme which is highly stable for the purposes of the present experiment, and is associated with lower computational costs since it is explicit. Padé time stepping (PTS), described in [5], provides such a scheme. It is based on Padé approximation (see [9]) for time stepping each component of an ODE system.

This chapter introduces PTS, particularly for the diffusion operator. Some of the difficulties which arise when applying PTS as well as approaches to resolve them are also discussed. First, PTS is introduced and a small-dimension case is analyzed. Then, a number of PTS approaches are modified and tested. Finally, results are presented especially for strongly diffusive problems.

2.1 Padé Time Stepping (PTS)

This section provides a brief review of PTS scheme. For full details refer to [5]. The [N/M]-Padé approximation of a function $u = u(t)$ is defined as a rational function of polynomials in t with numerator degree N, denominator degree M, and which matches the Taylor series up to order $N + M$. Consider the following ODE system

$$u_t(t,x) = f(u(t,x), u_x(t,x), t, x), \tag{2.1.1}$$

which encompasses both ODEs and PDEs using the method of lines. Here $x \in [a,b]$; $t_i = ih$ is the i^{th} time-step and $x_j = a + jdx$ is the j^th space-step where $i,j = 0,1,2,\ldots$. Define $u^j(t_i) = u(x_j,t_i)$. Assuming we have the following form for the Taylor series approximation

$$u^j(t_i + h) \approx \sum_q c_q^j(t_i)h^q, \tag{2.1.2}$$

representing the j-th component of the solution vector u, where $c_0^j(t_i) = u^j(t_i)$, $c_1^j(t_i) = (u^j(t_i))_t$, $c_2^j(t_i) = \frac{(u^j(t_i))_{tt}}{2}$, ... Then the Padé approximation to the Taylor series (2.1.2) would be

$$[N/M]^{j,t_i}(h) = \frac{a_0^{j,t_i} + a_1^{j,t_i}h + a_2^{j,t_i}h^2 + \cdots + a_N^{j,t_i}h^N}{1 + b_1^{j,t_i}h + b_2^{j,t_i}h^2 + \cdots + b_M^{j,t_i}h^M}. \tag{2.1.3}$$

To find the coefficients $a_0^{j,t_i},\ldots,a_N^{j,t_i}$ and $b_1^{j,t_i},\ldots,b_M^{j,t_i}$, we need to solve a system of equations which results from equating (2.1.2) and (2.1.3). This can be done uniquely and directly. For full details, see [9]. Numerically, a major problem with this formula occurs when the denominator is close to zero because of round-off and loss of precision. However, from a theoretical point of view this is not a problem since the limit formally exists, i.e. $\lim_{|D|\to 0}\left(\frac{a_0^{j,t_i}+a_1^{j,t_i}h+\cdots+a_N^{j,t_i}h^N}{1+b_1^{j,t_i}h+\cdots+b_M^{j,t_i}h^M}\right)$ $= \lim_{h\to h^*(D(h^*)\to 0)}\left(c_0^j(t_i) + c_1^j(t_i)h + \cdots + c_{N+M}^j(t_i)h^{N+M} + \mathcal{O}(h^{N+M})\right)$, which leads to a finite number. This issue was discussed in [5] with certain ad-hoc controls being proposed.

We will study this issue in more detail and propose other more general remedies in Section 2.4. One of the proposed approaches is to use a Taylor series of the same order when the norm of the denominator is less than a certain number. The other approaches are based on PTS itself but with some modifications.

2.2 Applying PTS[1/1] to the Heat Equation

Consider the simple heat equation in one dimension

$$u_t = u_{xx}. \tag{2.2.1}$$

Applying the centred difference of order 2 on the spatial variable we obtain the system

$$u_t^j = \frac{1}{dx^2}(u^{j+1} - 2u^j + u^{j-1}). \tag{2.2.2}$$

The second-order Taylor recurrence to (2.2.2) will be

$$u_{i+1}^j = c_0^j(t_i) + c_1^j(t_i)\left(\frac{h}{dx^2}\right) + \left(c_2^j(t_i)\right)\left(\frac{h}{dx^2}\right)^2, \tag{2.2.3}$$

where

$$\begin{aligned} c_0^j(t_i) &= u_i^j, \\ c_1^j(t_i) &= u_i^{j+1} - 2u_i^j + u_i^{j-1}, \\ c_2^j(t_i) &= \frac{1}{2}\left(u_i^{j+2} - 4u_i^{j+1} + 6u_i^j - 4u_i^{j-1} + u_i^{j-2}\right). \end{aligned}$$

Using (2.1.3) with $N = M = 1$, we get PTS[1/1]

$$u_{i+1}^j = \frac{a_0^j(t_i) + a_1^j(t_i)L}{1 + b_1^j(t_i)L}, \tag{2.2.4}$$

where $L = \frac{h}{dx^2}$. Equating both equations (2.2.3) and (2.2.4), we obtain the values of the coefficients $a_0^j(t_i)$, $a_1^j(t_i)$ and $b_1^j(t_i)$. Then, the recurrence for PTS[1/1] has the following form

$$u_{i+1}^j = \frac{c_0^j(t_i) + \frac{-c_2^j(t_i)c_0^j(t_i) + (c_1^j(t_i))^2}{c_1^j(t_i)}L}{1 + \frac{-c_2^j(t_i)}{c_1^j(t_i)}L}, \tag{2.2.5}$$

which can be easily implemented numerically.

This equation will produce a non-linear time stepping scheme with, as we will see, good stability criteria compared to explicit schemes. From a theoretical standpoint, the denominator being close to zero will not pose any difficulties. However, for calculations in finite precision, there will be a potential for significant error.

2.3 Analysis of PTS[1/1] for Small Dimension

For the sake of simplicity, consider the three-component case $j = 3$. The PDE is discretized by 3 points in x with functions $u^1(t)$, $u^2(t)$, and $u^3(t)$. Clearly, in the context of the PDE, this will represent an overly coarse representation. However, it also offers a simple case to investigate the properties of PTS.

Instead of u^1, u^2, and u^3, we use the variables x, y, and z. Then, the first iteration (2.2.5) of PTS[1/1] can be written as

$$\mathbf{F}(x,y,z) = \begin{pmatrix} F^1(x,y,z) \\ F^2(x,y,z) \\ F^3(x,y,z) \end{pmatrix} = \begin{pmatrix} \frac{xy-2x^2+(x^2+y^2-2xy-\frac{1}{2}xz)L}{y-2x+(-\frac{1}{2}z+2y-3x)L} \\ \frac{zy-2y^2+xy+(x^2+y^2+z^2-2yz+2xz-2xy)L}{z-2y+x+(2z-3y+2x)L} \\ \frac{zy-2z^2+(z^2+y^2-2zy-\frac{1}{2}xz)L}{y-2z+(-\frac{1}{2}x+2y-3z)L} \end{pmatrix}. \tag{2.3.1}$$

Assume we have the initial condition $(x,y,z) = (\lambda\tilde{x}, \lambda\tilde{y}, \lambda\tilde{z})$ where $(\tilde{x}, \tilde{y}, \tilde{z}) \in \Omega$ for a scalar λ. Note that it can be easily seen that

$$\mathbf{F}(x,y,z) = \mathbf{F}(\lambda\tilde{x}, \lambda\tilde{y}, \lambda\tilde{z}) = \lambda\mathbf{F}(\tilde{x}, \tilde{y}, \tilde{z}).$$

The same will happen when multiple steps of PTS[1/1] are considered, i.e.,

$$\begin{aligned} \mathbf{F}\left(\ldots(\mathbf{F}(x,y,z))\right) &= \mathbf{F}\left(\ldots(\mathbf{F}(\lambda\tilde{x}, \lambda\tilde{y}, \lambda\tilde{z}))\right) = \mathbf{F}\left(\ldots\lambda\left(\mathbf{F}(\tilde{x}, \tilde{y}, \tilde{z})\right)\right) = \ldots \\ &= \lambda\mathbf{F}\left(\ldots(\mathbf{F}(\tilde{x}, \tilde{y}, \tilde{z}))\right). \end{aligned}$$

Therefore, due to this scale invariance, in the present test it is sufficient to consider the initial condition on the unit sphere. $\Omega = \{(x,y,z) : (x^2+y^2+z^2)^{\frac{1}{2}} = 1\}$.

To understand the behaviour of PTS[1/1] in a three-component case, the problem is transformed from a three dimensional to a spherical coordinate representation which helps study the schemes over a wider domain of initial conditions. By letting $u = (x, y, z) = (\cos\phi, \sin\phi\cos\theta, \sin\phi\sin\theta)$, where $0 \leq \phi \leq \pi$ and $0 \leq \theta \leq 2\pi$, we consider the initial condition on the unit sphere. Figure 2.1 confirms that $F(\theta, \phi)$ is bounded away from the points of singularity, but it also shows that there is some substantial numerical error which arises near these points. It will be demonstrated that these peaks are associated with the singularities of each component.

It is necessary to identify the denominator of each component:

$$\begin{aligned} D_1 &= y - 2x + \left(-\frac{1}{2}z + 2y - 3x\right) L, \\ D_2 &= z - 2y + x + (2z - 3y + 2x)L, \\ D_3 &= y - 2z + \left(-\frac{1}{2}x + 2y - 3z\right) L. \end{aligned} \tag{2.3.2}$$

Rewriting these formulas using polar representation and equating them to zero, and then solving for ϕ, we get

$$\begin{aligned} \phi_{D_1} &= \arctan\left(\frac{2 + 3L}{\cos\theta - (L/2)\sin\theta + (2L)\cos\theta}\right), \\ \phi_{D_2} &= \arctan\left(\frac{-2L}{\sin\theta - 2\cos\theta + 1 + (2L)\sin\theta - (3L)\cos\theta}\right), \\ \phi_{D_3} &= \arctan\left(\frac{(L/2)}{\cos\theta - 2\sin\theta - (3L)\sin\theta + (2L)\cos\theta}\right). \end{aligned} \tag{2.3.3}$$

Based on this, Figure 2.2 presents the predicted location of the zeros which coincide with the location of the peaks in Figure 2.1. Therefore, this confirms that the numerical artifact seen in Figure 2.1 is due to a vanishing denominator. This issue can be illustrated by paying direct attention to the floating point representation [1]. It simply arises when two close numbers are subtracted.

Without loss of generality, we focus on one component; the second component F^2. Figure 2.3 describes in more detail the singularities as a function of ϕ, when $\theta = 2$, and shows the number and width of these peaks.

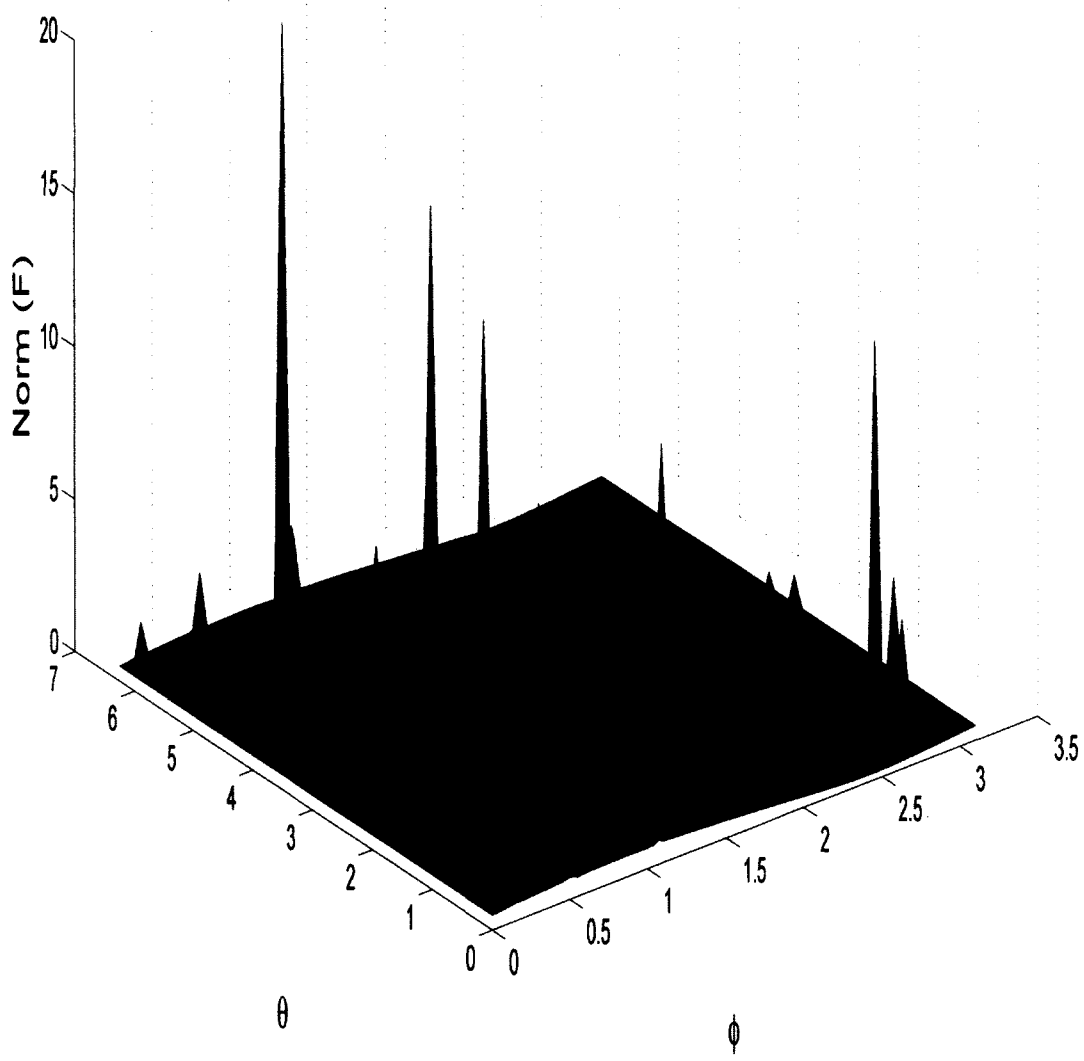

Figure 2.1: First iteration of the PTS[1/1] scheme in F, in the ϕ and θ-space when $L = 0.5$.

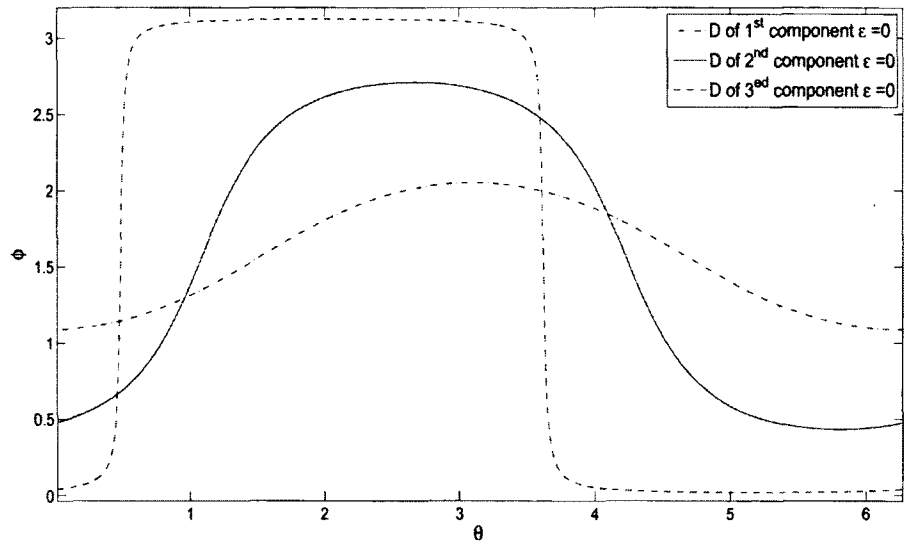

Figure 2.2: Singularities of each component in the ϕ, θ-space when $L = 0.5$.

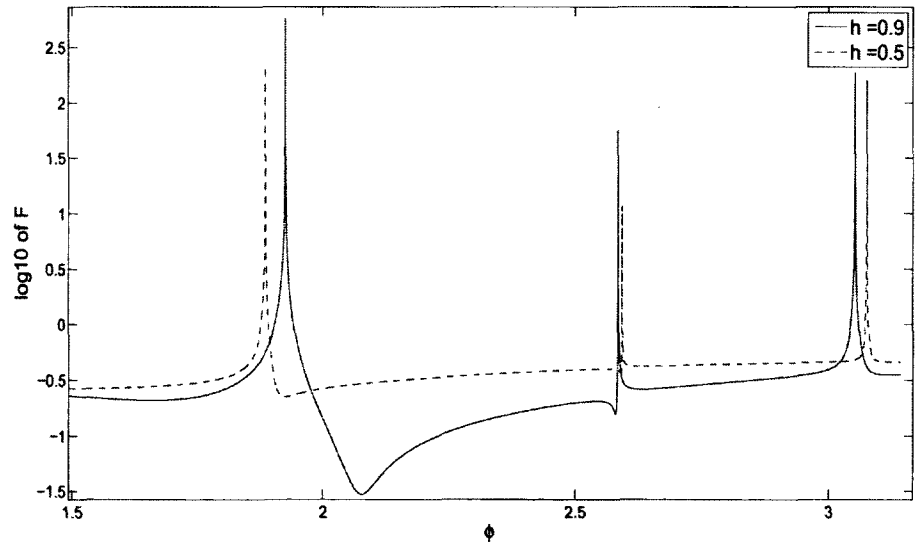

Figure 2.3: Norm of the second component of F as a function of ϕ when $\theta = 2$.

Thus, in order to have a convergent scheme, these peaks need to be truncated by using a different scheme, not just at the point of singularity, but in the neighbourhood around the singularity. Therefore, it is crucial to choose a threshold to remove these peaks. We will call this threshold ϵ. This is equivalent to considering a band of curves around the singularities of width proportional to ϵ as shown in Figure 2.5. Determining an appropriate value for ϵ is now a key question to consider.

An important property of these peaks is that as h gets smaller, their locations change and their widths get narrower. From (2.3.2),

$$D(x, y, z, h) = z - 2y + x + (2z - 3y + 2x)h.$$

Or using spherical coordinates,

$$D(\phi, \theta, h) = \sin\phi\sin\theta - 2\sin\phi\cos\theta + 2\cos\phi + (2\sin\phi\sin\theta - 3\sin\phi\cos\theta + 2\cos\phi)\,h.$$

Assume $\theta = 0$ for simplicity. A singularity exists at ϕ^* when $h = h^*$ that is,

$$D(\phi^*, 0, h^*) = 0.$$

If the width of this peak is 2δ, then the boundaries of the band are defined by

$$D(\phi^* - \delta, 0, h^*) = \epsilon,$$

which could be written as

$$-2\sin(\phi^* - \delta) + 2\cos(\phi^* - \delta) + (-3\sin(\phi^* - \delta) + 2\cos(\phi^* - \delta))\,h^* = \epsilon.$$

Solving for δ, and simplifying, leads to

$$\begin{aligned}\delta(h^*) &= \phi^* + \pi - \arcsin\left(\frac{\epsilon}{\sqrt{(-2-3h^*)^2 + (2+2h^*)^2}}\right) \\ &\quad - \arcsin\left(\frac{2+2h^*}{\sqrt{(-2-3h^*)^2 + (2+2h^*)^2}}\right),\end{aligned}$$

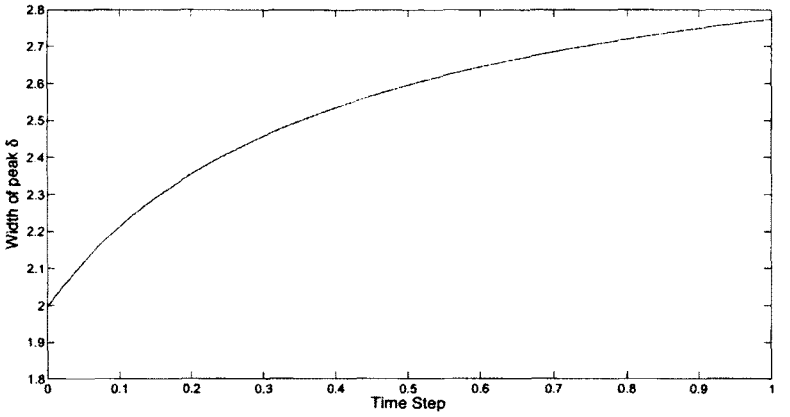

Figure 2.4: Width of peaks δ as function of time step when $\theta = 0$.

where $\phi^* = \arctan\left(\frac{2h}{1+3h}\right)$ from equation (2.3.3). The width of peaks, δ, is an increasing function with respect to the time step, h, i.e., δ gets smaller as h goes to zero. This property is shown in Figure 2.4.

In terms of changing singularity locations, consider a singularity at $\phi = \phi_1$ when $h = h_1$ and a singularity at $\phi = \phi_2$ when $h = h_2$. We want to show that $\phi_1 \neq \phi_2$ when $h_2 \neq h_1$. Assume the contrary, that $\phi_1 = \phi_2$. Using equation (2.3.3) leads to

$$\frac{2h_1}{1+3h_1} = \frac{2h_2}{1+3h_2}$$

which implies $h_2 = h_1$, which is a contradiction. Hence, the location of these peaks changes as h changes.

2.4 Approaches to Control PTS[1/1]

Given the existence of singularities, this section introduces a few controls. Some approaches are established in order to control the error(s) resulting from having a small denominator. For the bulk of the iterations these approaches use PTS[1/1] when the magnitude of the denominator is greater than a threshold. Otherwise, they apply some other approach such as Taylor, PTS[0/2], or PTS[1/1] with a smaller time

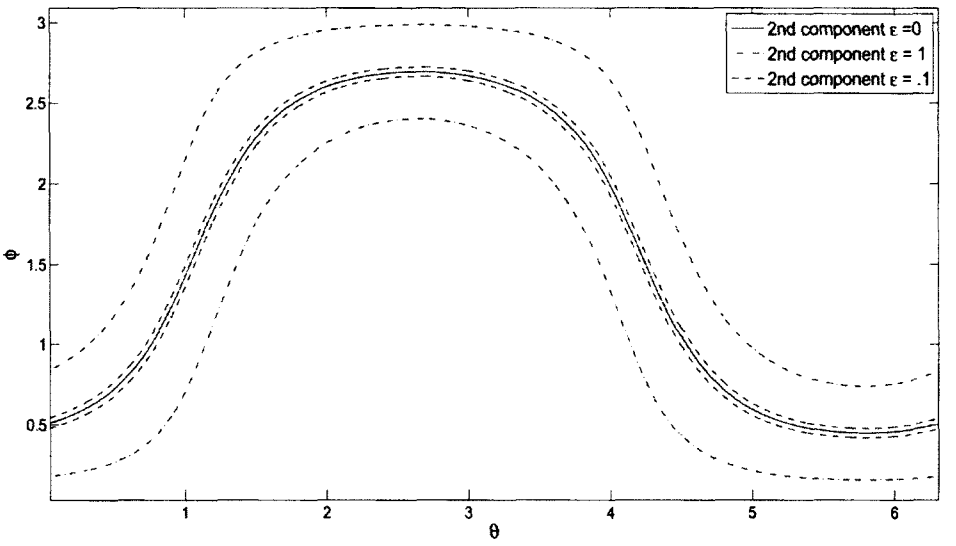

Figure 2.5: ϵ-bands around zeros of denominator of the second component in ϕ and the θ-space when $L = 0.25$.

step.

For simplicity consider the system of ODEs

$$\begin{aligned} x_t &= -2x + y, \\ y_t &= z - 2y + x, \\ z_t &= -z + y. \end{aligned} \tag{2.4.1}$$

In addition, assume (x_0, y_0, z_0) is of magnitude one and gives a zero in at least one denominator of the components when the first iteration of PTS[1/1] is performed. This system will be used as a model for the present study. These types of initial conditions guarantee that the worst case scenario is encountered during the computation. As we will see, the determination of the threshold depends on the stiffness of the operator. To begin, we will consider the non-stiff case. The case where the value of $\frac{\alpha}{dx^2}$ is large will be discussed in the last section.

Local Error Control Threshold (LECT) ϵ^*

A threshold ϵ^* is needed when the denominator is close to zero to prevent local error growth. Another scheme of order two can be used instead of PTS[1/1] when the condition

$$\left|1 - \frac{c_2}{c_1}L\right| \leq \epsilon^*$$

is met. This threshold, ϵ^*, will be called the local error control threshold (LECT).

From the theoretical formulation of local error for Taylor and PTS, we have

$$\begin{aligned} |\text{PTS - Taylor}| &= \frac{c_2^2 L^3}{c_1}\left(1 + \frac{c_2 L}{c_1} + \left(\frac{c_2 L}{c_1}\right)^2 + \ldots\right) \\ &= \frac{c_2^2 L^3}{c_1}\left(\frac{1}{1 - \frac{c_2}{c_1}L}\right). \end{aligned}$$

This justifies the need for the optimal LECT, ϵ^*, to be greater than or equals to one to insure that the local error will not grow. Then, the numerical runs may be used to define optimal values of thresholds that give accurate results.

2.4.1 PTS$[1/1]_{\text{Taylor}}$ Approach

This approach was initially used in [5] to solve some examples of ODEs and PDEs. However, in this section, additional information about LECT is introduced. We start with the first iteration of the PTS[1/1] scheme and then monitor the denominators of each component. Whenever the norm of the denominator is less than ϵ^*, then Taylor of order two is applied to that component. The optimal value of LECT, ϵ^*, will be discussed further in the chapter; for now, we use $\epsilon^* = 1.2$ unless otherwise stated. A major concern for this approach is the percentage of step where Taylor is used when Taylor is unstable. Additionally, it is relevant to consider how many such steps are in sequence?

Local Error (Consistency) of the PTS$[1/1]_{\text{Taylor}}$ Approach

First, recall the theoretical local error of second-order Taylor is

$$LE_{\text{Taylor}} = c_3 h^3 + \mathcal{O}(h^4).$$

The theoretical local error of PTS[1/1] can be extracted to be

$$LE_{[1/1]} = \left(\frac{c_2^2}{c_1} - c_3\right) h^3 + \mathcal{O}(h^4); \tag{2.4.2}$$

for each component. When we apply the first iteration of this approach, we use Taylor only once in the second component, since we chose an initial condition such that the second component has a singularity. Figure 2.6 shows the truncation of the spurious error resulting from using $\epsilon^* = 1.2$ for initial conditions generated by ϕ and $\theta^* = 2$ where all have initial singularities at the second component. Even though Taylor was used once only in one component, we see a significant improvement in the overall error. In addition, Figure 2.7 indicates that we are getting the right order of accuracy, which is order three. Therefore, in general, the theoretical local error formula for PTS$[1/1]_{\text{Taylor}}$ should be a combination of LE_{Taylor} and $LE_{[1/1]}$, depending on the number of singularities.

Global Error (Stability) of the PTS$[1/1]_{\text{Taylor}}$ Approach

The stability of the PTS$[1/1]_{\text{Taylor}}$ approach will depend on the choice of LECT ϵ^*. Thus, a few tests are required. First, the factors that affect the choice of epsilon are considered. Secondly, the value of the optimum LECT ϵ^* is discussed.

LECT, ϵ^*, for the PTS$[1/1]_{\text{Taylor}}$ Approach In the early stages of this work, we were expecting ϵ^* to be around 10^{-16} due to the standard double precision round-off errors. However, this turned out to be incorrect; the value of LECT ϵ^* needs to be much bigger than this number.

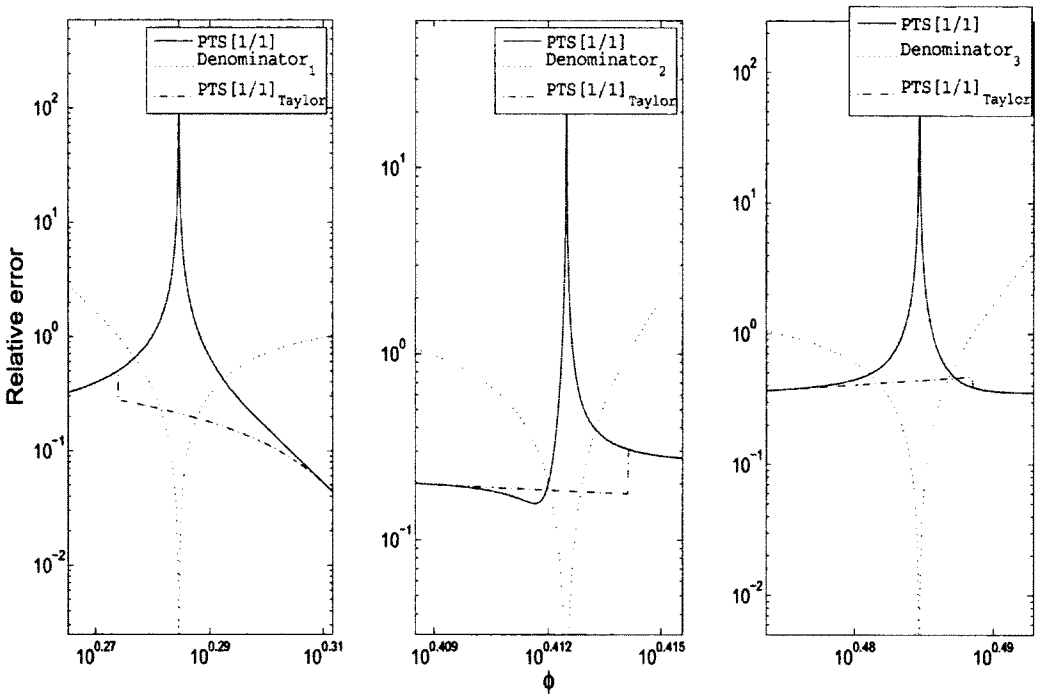

Figure 2.6: The first iteration of the PTS$[1/1]_{\text{Taylor}}$ approach in ϕ and error space, $h = 0.9$, $\epsilon = 1$.

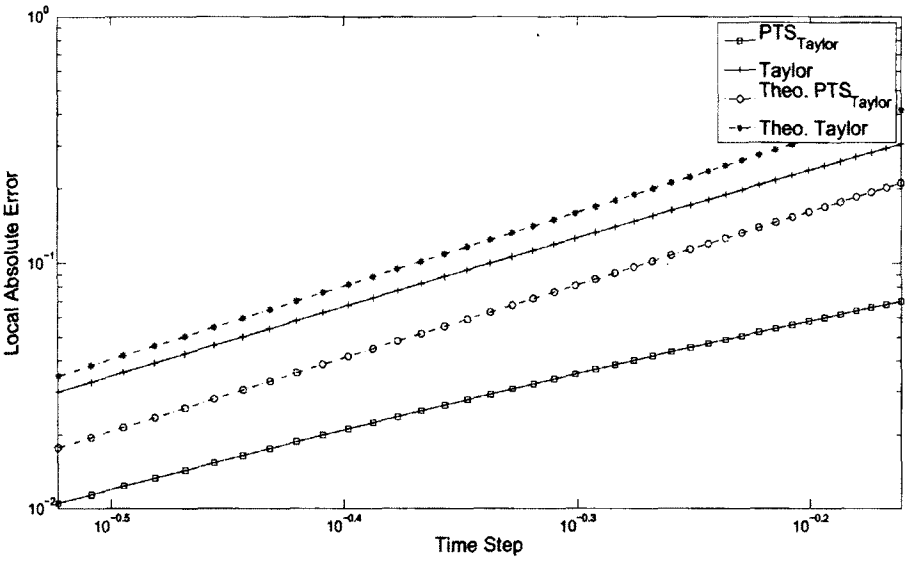

Figure 2.7: Theoretical and numerical local errors of one step of the $\text{PTS}[1/1]_{\text{Taylor}}$ approach versus time step h, $\epsilon = 1.2$.

Secondly, one of the properties of the optimal LECT is its independence on the length of the run. Figure 2.8 presents different plots with respect to different time steps. For example, when $h = 0.5$, all T_{final} plots behave qualitatively the same when $\epsilon^* \in (0.33, 1.094)$. Generally speaking, this suggests that for each h there is an interval of ϵ^* where the outcomes are the same.

The decision making process for the right value of LECT, ϵ^*, will be based on three steps. The first step is to find the lower bound for ϵ^* that will eventually reduce the effect of division by zero. The second step is to find the upper bound that will control and limit the use of Taylor. The last step is to study the interval of ϵ and then try to reduce that interval to obtain the optimal LECT ϵ^*. We started by testing small values of ϵ such as 10^{-16}. Figure 2.9 shows a lower bound of roughly 10^{-1} . Testing large values of ϵ will lead to an upper bound for ϵ^* which is 2, see Figure 2.10. So far, the optimal LECT $\epsilon^* \in (0.1, 2)$ is recognized, though it would be difficult to find it explicitly. For this purpose this study will focus on the interval where Taylor

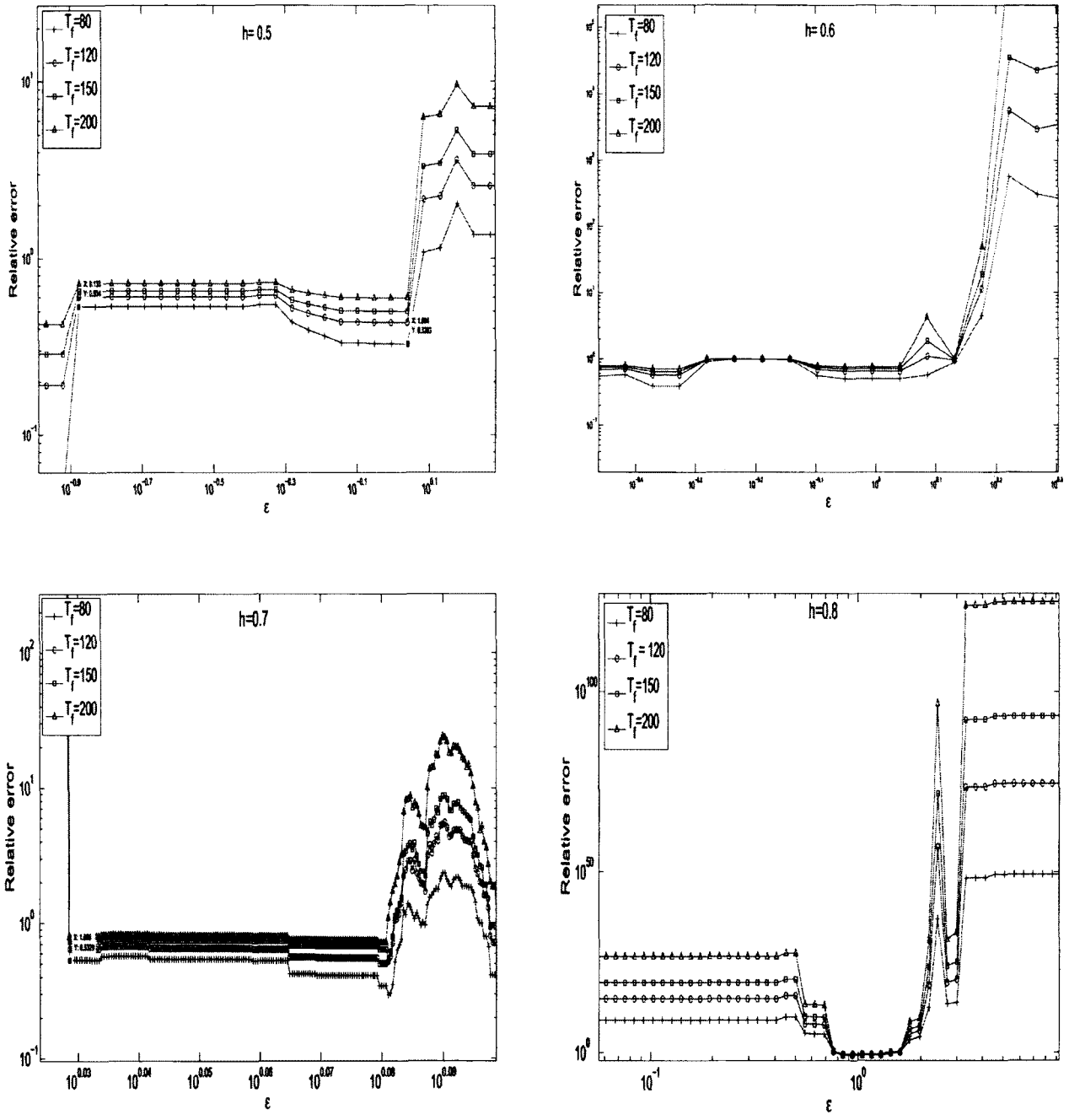

Figure 2.8: Plot of ϵ versus relative error for different time step h.

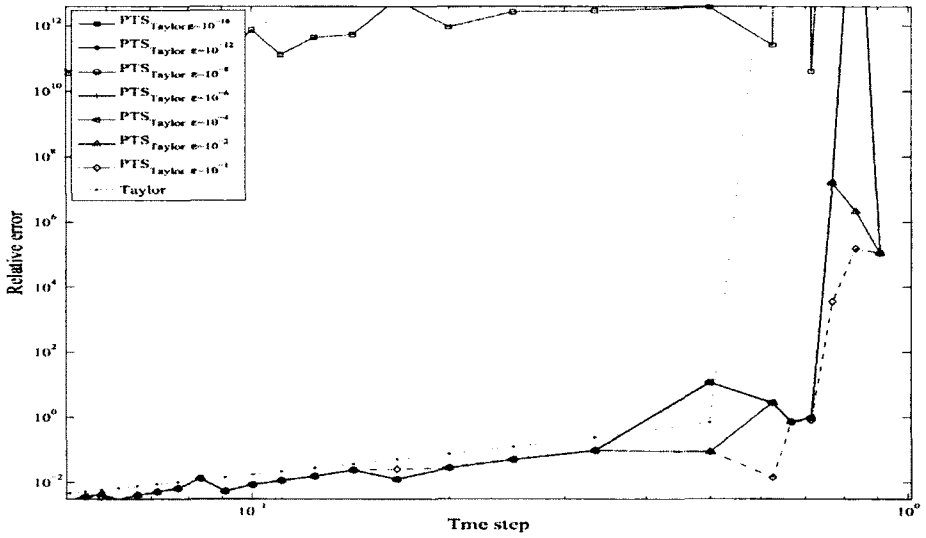

Figure 2.9: Lower bound of LECT, ϵ^*, for the PTS$[1/1]_{\text{Taylor}}$ approach when different values of ϵ are tested, $T_f = 50$.

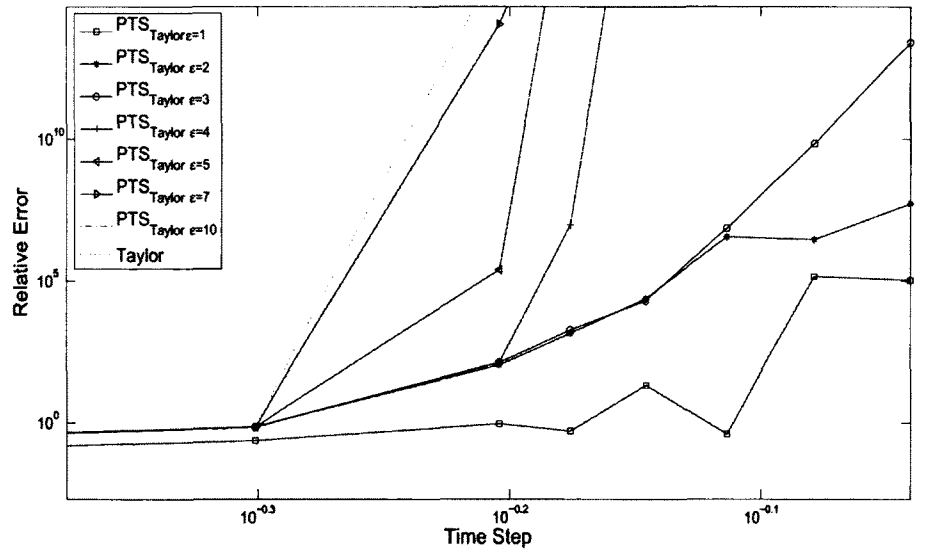

Figure 2.10: Upper bound of LECT, ϵ^*, for the PTS$[1/1]_{\text{Taylor}}$ approach when different values of ϵ are tested, $T_f = 50$.

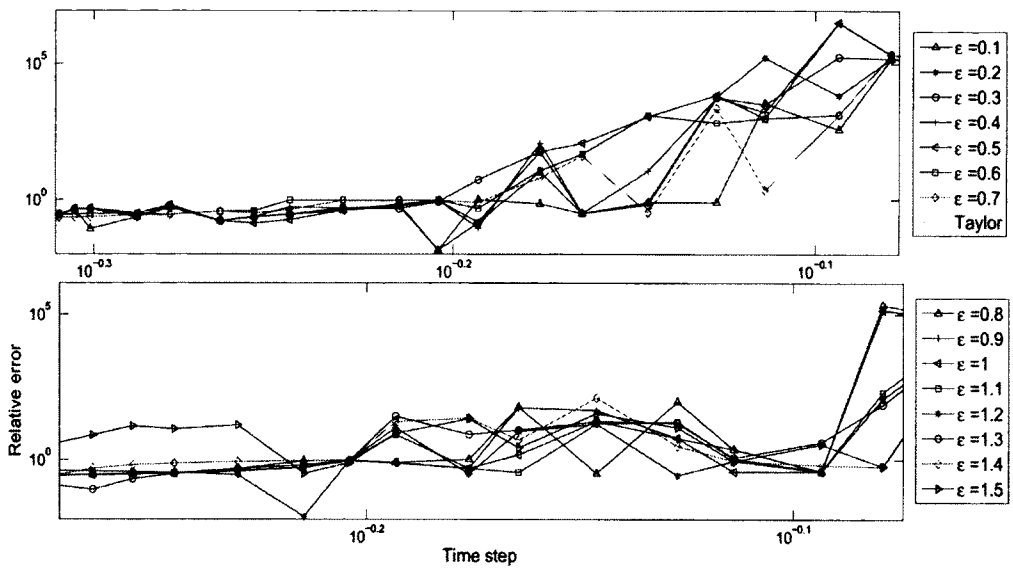

Figure 2.11: Plot of time step h versus relative error for various values of ϵ, $T_f = 50$.

is unstable, i.e., $h \in (0.5, 1)$ in our example.

As shown in Figure 2.11 and Figure 2.12, there are sharp transitions in ϵ and h, which made the search for optimal LECT, ϵ^*, difficult. However, in most runs we were using selections from neighbouring ϵ^*, which had excellent results. Therefore, we determined to set LECT, ϵ^*, for $\text{PTS}[1/1]_{\text{Taylor}}$ at 1.2. Figure 2.13 shows that there is a big improvement of the $\text{PTS}[1/1]_{\text{Taylor}}$ approach on the error side, especially when the Taylor scheme is unstable. However, some of the results in our scheme may lead to instability. For example, at the instant $h = 0.7264$, that is indicated in Figure 2.14, a bad result for the $\text{PTS}[1/1]_{\text{Taylor}}$ is observed. Table 2.1 presents one factor of that error, which is mostly due to the use of Taylor in sequence; in total, 27% of Taylor was used.

This sequential use of Taylor can be controlled in two ways, either by reducing the value of ϵ or by tracking the sequence of steps when Taylor is used. Reducing the threshold ϵ to solve the issue was briefly mentioned before. Afterwards, we introduced

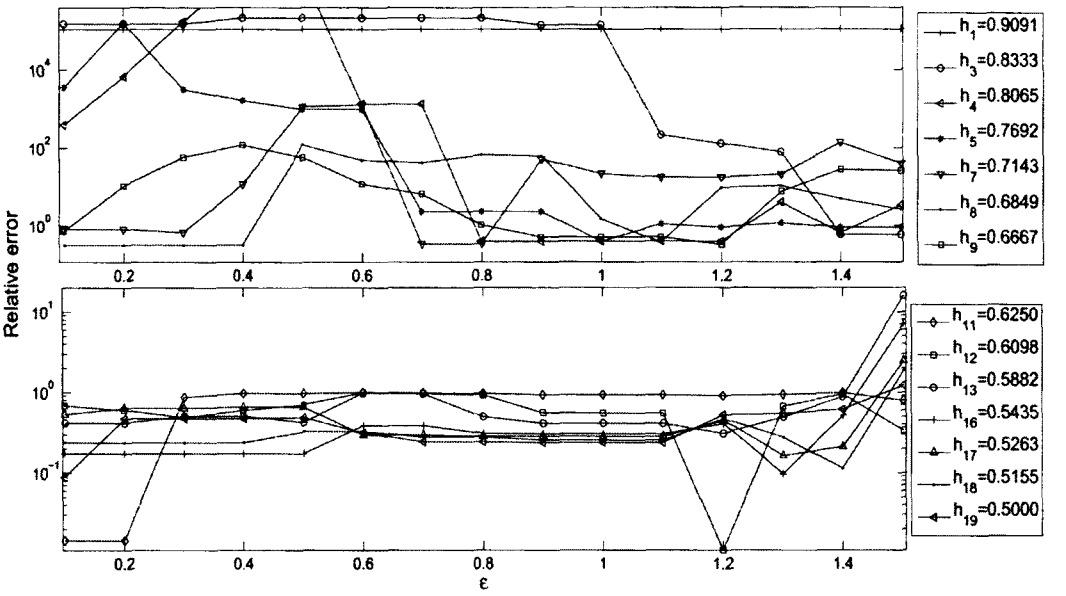

Figure 2.12: Plot of relative error versus ϵ for various values of time step h, $T_f = 50$.

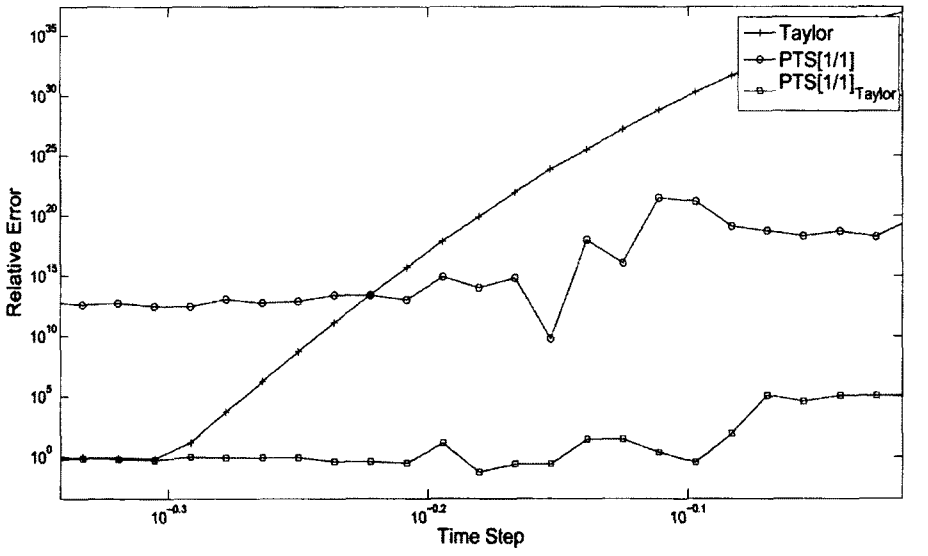

Figure 2.13: Plot of time step h versus relative error; full iteration, $T_f = 40$, $\theta = 2$ and $\epsilon^* = 1.2$.

	location (iteration number) of Taylor used
1^{st} component	3, 4, 5, 6, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 25, 29, 35, 49, 53, 59, 65
2^{nd} component	1, 9, 16, 21, 22, 23, 27, 31, 37, 45, 51, 57, 63
3^{rd} component	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 18, 19, 20, 21, 25, 29, 35, 49, 53, 59, 65

Table 2.1: The location (iteration number) of Taylor used of $\text{PTS}[1/1]_{\text{Taylor}}$ at $h = 0.7264$.

a modification to the $\text{PTS}[1/1]_{\text{Taylor}}$ approach to control for the use of Taylor.

Modified $\text{PTS}[1/1]_{\text{Taylor}}$ Approach

This part of the chapter presents a possible modification of our approach to prevent using Taylor in sequence. As shown in Algorithm 2.4.1, two thresholds, ϵ_1 and ϵ_2, are needed. ϵ_1 is applied when PTS is used in the previous step and ϵ_2 whenever Taylor is used in the previous iteration, and where $\epsilon_2 \ll \epsilon_1$.

Algorithm 2.4.1 Modified $\text{PTS}[1/1]_{\text{Taylor}}$

$\mathbf{c = 0}$

for $\mathbf{i = 1 : n}$

if $(|\mathbf{D_i}| \leq \epsilon_1 \;\; \& \;\; \mathbf{c = 0})$; **then** apply Taylor & $\mathbf{c = 1}$

elseif $(|\mathbf{D_i}| \leq \epsilon_2 \;\; \& \;\; \mathbf{c = 1})$; **then** apply Taylor & $\mathbf{c = 1}$

otherwise **apply** PTS & $\mathbf{c = 0}$

end

By using Algorithm 2.4.1 with thresholds $\epsilon_1 = 1.2$ and $\epsilon_2 = 10^{-2}$, we see changes of sequence use of Taylor in Table 2.2. In addition, only 9% of Taylor was used. Figure 2.14 shows an improvement by the modified scheme over the regular approach. The modified PTS change the jumps behaviour to a flat mode. Another advantage of this

	location (iteration number) of Taylor used
1^{st} component	1, 6, 8, 13, 15, 29, 31
2^{nd} component	1, 4, 28, 32
3^{rd} component	1, 4, 7, 10, 11, 30, 34

Table 2.2: The location (iteration number) of Taylor used of the modified $\text{PTS}[1/1]_{\text{Taylor}}$ at $h = 0.7264$.

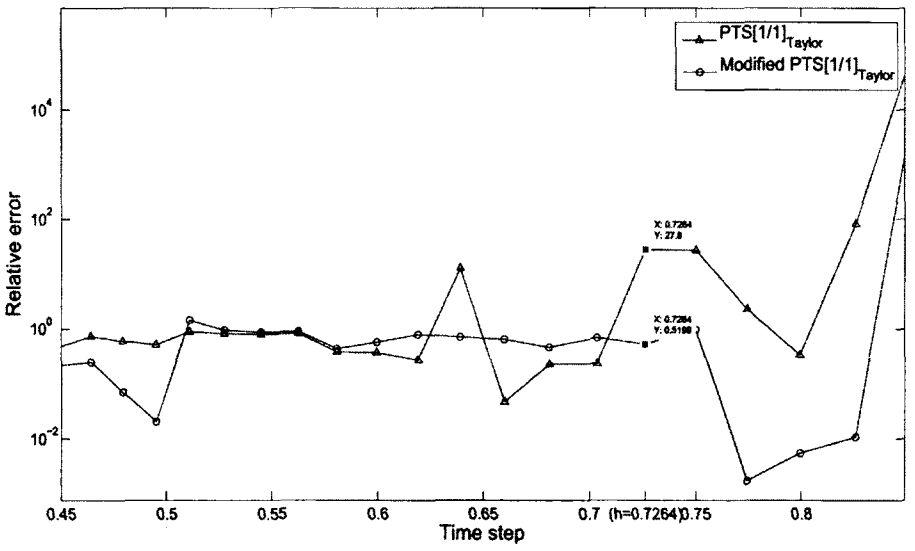

Figure 2.14: Plot of the relative error versus time step h of modified $\text{PTS}[1/1]_{\text{Taylor}}$.

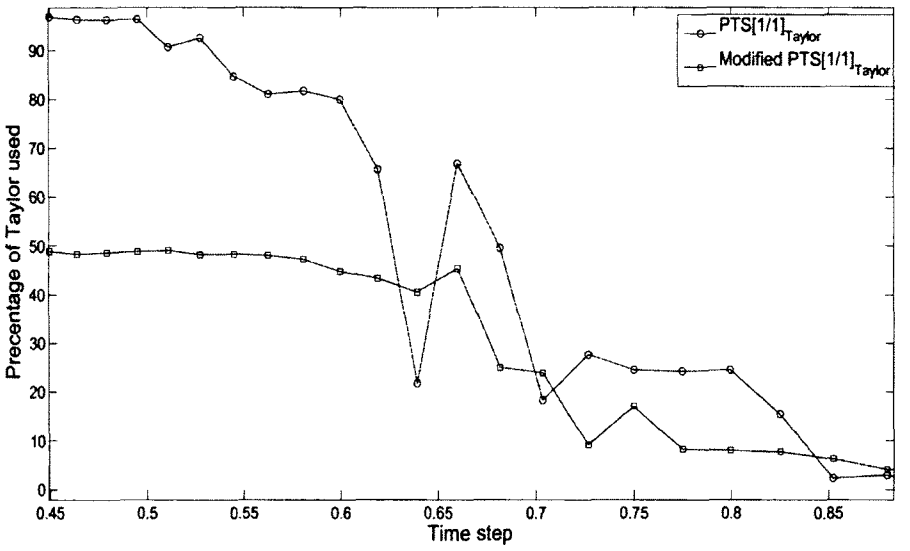

Figure 2.15: Plot of the percentage of Taylor used of the modified $\text{PTS}[1/1]_{\text{Taylor}}$ versus h.

modification is that the percentage of Taylor used is reduced approximately by half as shown in Figure 2.15.

2.4.2 $\text{PTS}[1/1]_{[0/2]}$ Approach

In the previous subsection, we demonstrated the use of Taylor to control the main scheme. In this subsection, we will apply a different approximation from the same second order PTS family, namely PTS[0/2].

PTS[0/2] can also be used whenever PTS[1/1] needs to be controlled. This approach will be named the $\text{PTS}[1/1]_{[0/2]}$ approach. First, from equation (2.1.3) with $N = 0$, $M = 2$, in terms of the Taylor coefficients c_0, c_1 and c_2, PTS[0/2] is

$$\text{PTS}[0/2](h) = \frac{c_0}{1 + \left(-\frac{c_1}{c_0}\right) h + \left(\frac{\frac{c_1^2}{c_0} - c_2}{c_0}\right) h^2}. \tag{2.4.3}$$

If, for example, $c_1 = 0$, PTS[1/1] suffers, since $u_{n+1} = c_0 = u_n$, while PTS[0/2]

does not suffer since we make use of c_2, i.e., $u_{n+1} = \frac{c_0^3}{c_0^2 - c_0 c_2 h^2}$. This is a compelling reason to use PTS[0/2]. Prior to applying this scheme, it is necessary to clarify the mathematical relation between the singularity of different types of PTS schemes. The denominators of PTS[1/1] could be written in term of c's and time step h as

$$D_{[1/1]}(h) = 1 + \left(-\frac{c_2}{c_1}\right) h. \tag{2.4.4}$$

Thus, a singularity exists at $h = \frac{c_1}{c_2}$. From equation (2.4.3), we have the formula of the denominator of PTS[2/0] which is simply

$$D_{[0/2]}(h) = 1 + \left(-\frac{c_1}{c_0}\right) h + \left(\frac{\frac{c_1^2}{c_0} - c_2}{c_0}\right) h^2. \tag{2.4.5}$$

Substitute $h = \frac{c_1}{c_2}$ into equation (2.4.5) to test for a common singularity. Consequently,

$$D_{[0/2]}\left(\frac{c_1}{c_2}\right) = 1 + \left(-\frac{c_1}{c_0}\right)\left(\frac{c_1}{c_2}\right) + \left(\frac{\frac{c_1^2}{c_0} - c_2}{c_0}\right)\left(\frac{c_1}{c_2}\right)^2 = 0.$$

This leads to the following condition

$$c_1^2 = c_0 c_2. \tag{2.4.6}$$

Two solutions: $(\theta_1, \phi_1) = \left(\frac{\pi}{2}, \frac{3\pi}{4}\right)$ and $(\theta_2, \phi_2) = \left(\frac{3\pi}{2}, \frac{\pi}{4}\right)$ are obtained. In fact, these two points represent two unimportant cases when $c_0 = c_1 = c_2 = 0$. These points correspond to fixed points of the ODE. As mentioned earlier, we demonstrated that there will not be an intersection between the singularities of PTS[1/1] and PTS[0/2] besides the trivial fixed points. Figure 2.16 presents this result for some types of PTS schemes.

Assuming we are not at a fixed point, for the PTS$[1/1]_{[0/2]}$ approach, if PTS[1/1] has a singularity, then PTS[0/2] does not have one. Hence, it is a good choice to control PTS[1/1]. In addition, we could use another mixture of PTS schemes depending on the order needed.

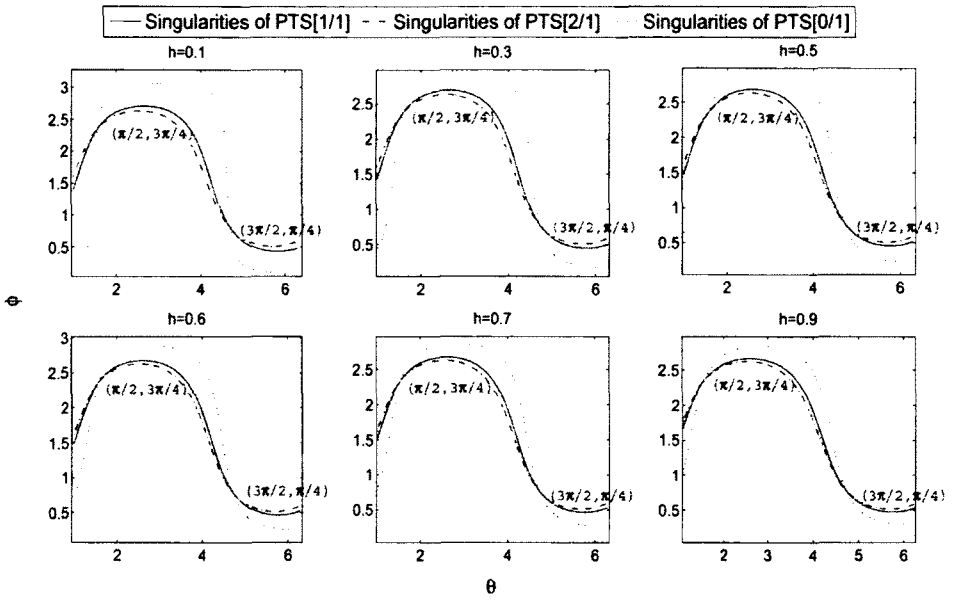

Figure 2.16: Plot of singularities of different classes of PTS in the θ, ϕ-space.

Local Error (Consistency) of the PTS$[1/1]_{[0/2]}$ Approach

We recall first $\mathrm{LE}_{\mathrm{PTS}[1/1]}$ in (2.4.2) and then will clarify the theoretical local error of PTS[0/2]. Expanding and simplifying the rational expression

$$\begin{aligned} PTS[0/2] &:= \frac{c_0}{1+\left(-\frac{c_1}{c_0}\right)h+\left(\frac{\frac{c_1^2}{c_0}-c_2}{c_0}\right)h^2} \\ &= c_0+c_1h+c_2h^2+\left(\frac{2c_0c_1c_2-c_1^3}{c_0^2}\right)h^3+\mathcal{O}(h^4); \end{aligned} \tag{2.4.7}$$

Therefore,

$$LE_{[0/2]} = \left(\frac{2c_0c_1c_2-c_1^3}{c_0^2}\right)h^3+\mathcal{O}(h^4). \tag{2.4.8}$$

Figure 2.17 shows the improvement as a result of applying PTS[0/2] on PTS[1/1]. In addition, Figure 2.18 indicates that we are getting close to the right order of accuracy, which is order three. Finally, the theoretical local error formula for $\mathrm{PTS}[1/1]_{[0/2]}$ would be a combination of $LE_{[1/1]}$ and $LE_{[0/2]}$, depending on the number of singularities.

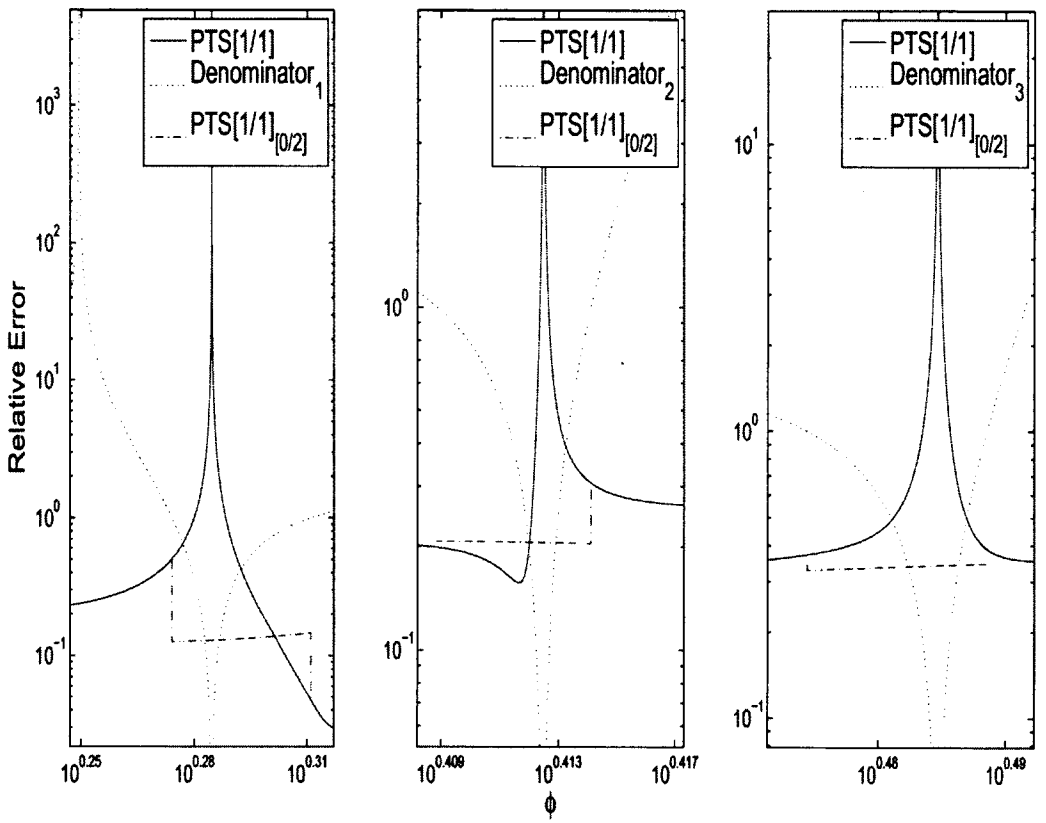

Figure 2.17: Plot of the first iteration of the PTS$[1/1]_{[0/2]}$ approach in the ϕ and error space, $\theta = 2$, $h = 0.9$, and $\epsilon^* = 1$.

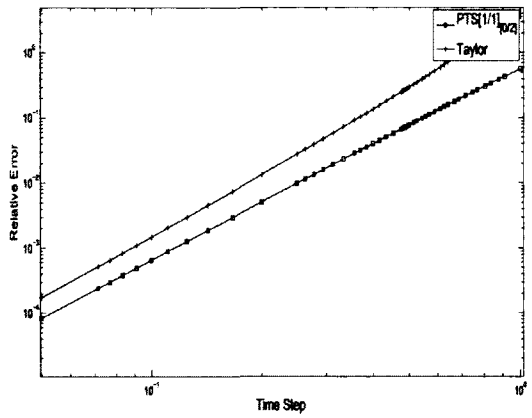

Figure 2.18: The first iteration of the PTS$[1/1]_{[0/2]}$ approach in the time step and error space.

Global Error (Stability) of the PTS$[1/1]_{[0/2]}$ Approach

For the purposes of the full scheme, we need to define the optimal LECT, ϵ^*, for the PTS$[1/1]_{[0/2]}$ approach, which will produce the new stable scheme for this study.

LECT, ϵ^* of the PTS$[1/1]_{[0/2]}$ Approach

As discussed previously, it is necessary to track the optimal LECT, ϵ^*, for PTS$[1/1]_{[0/2]}$. Figure 2.19 shows that the scheme for this study required ϵ^* to be in $(0.91, 1.0)$. Setting $\epsilon^* = 1$, as shown in Figure 2.19, produces a good result. It can be noticed the bad result for $\epsilon = 1.1$ as $h \to 0$.

2.4.3 PTS$[1/1](\frac{h}{s})$ Approach

Since the location of singularities depends on the time step h, this provides another way in which the error can be controlled. We apply a $\frac{1}{s}$ step of PTS[1/1] instead of applying the full time step where $s = 1, 2, \ldots$. We will call this PTS$[1/1](\frac{h}{s})$. First, the applicability of this approach is discussed. Recall the denominator of PTS[1/1]

$$D_{[1/1](h)} = 1 - \frac{c_2}{c_1}h.$$

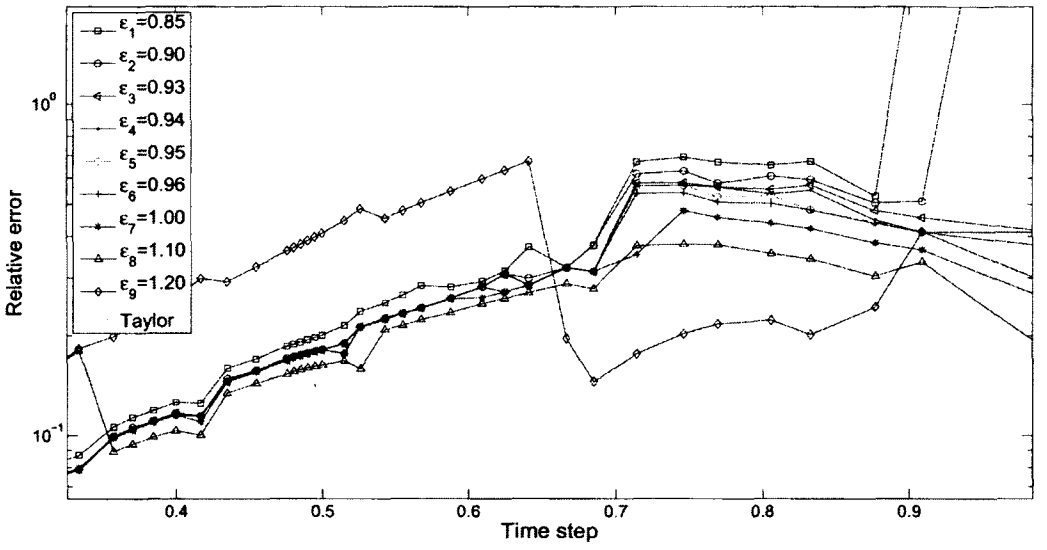

Figure 2.19: Searching for the optimal value of the LECT for PTS$[1/1]_{[0/2]}$ approach, $T_f = 50$ and $\theta = 2$.

If the condition, $|D_{[1/1](h)}| \leq \epsilon^*$ is satisfied, the denominator of the first iteration of PTS$[1/1](\frac{h}{s})$ can be written as

$$D_{[1/1](\frac{h}{s})} = 1 + \left(-\frac{c_2}{c_1}\right)\left(\frac{h}{s}\right).$$

The singularity of PTS[1/1] exists at $h^* = \frac{c_1}{c_2}$. The new approach denominator at h^* will be

$$\text{PTS}[1/1](\frac{h}{s})\Big|_{h=h^*} = 1 + \left(-\frac{c_2}{c_1}\right)\left(\frac{h^*}{s}\right) = \frac{1}{s} \neq 0.$$

For further clarification, see Figure 2.20 where we have only two points of intersection, namely $(\theta_1, \phi_1) = \left(\frac{\pi}{2}, \frac{3\pi}{4}\right)$ and $(\theta_2, \phi_2) = \left(\frac{3\pi}{2}, \frac{\pi}{4}\right)$, when $s = 2, 3, 4$ are considered, as in the previous approach.

Therefore, we are certain that the new approach does not have a singularity at the first iteration. Unfortunately, it may have one in the second iteration since c_0, c_1, and c_2 will be updated. To solve this issue, we suggest to use a bigger s instead. This will reduce the possibility of singularities at that time step.

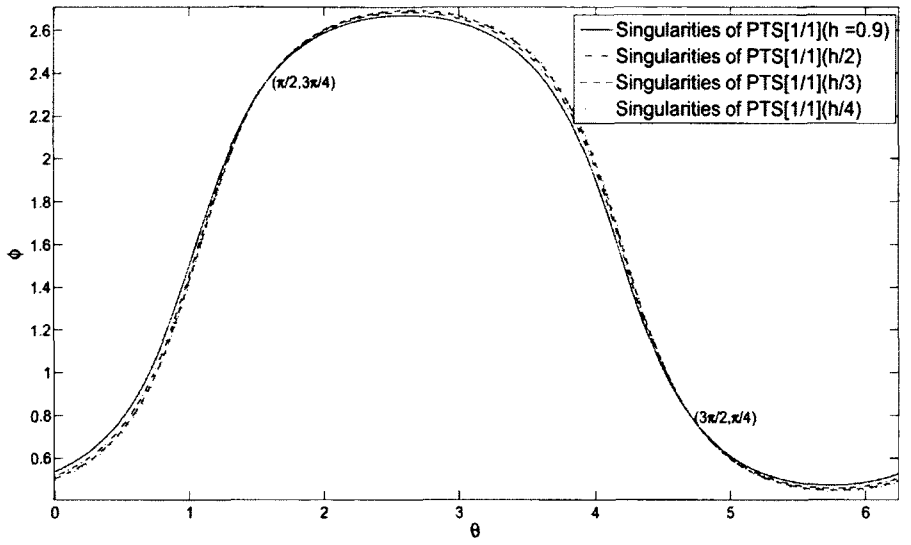

Figure 2.20: PTS[1/1] at different time steps, $\frac{h}{2}$, $\frac{h}{3}$ and $\frac{h}{4}$ in the ϕ and θ-space.

If $D\left(\frac{h}{2}\right) = \epsilon$, which implies $\frac{c_2}{c_1} = (1-\epsilon)\frac{2}{h}$. Therefore,

$$D\left(\frac{h}{4}\right) = \epsilon + \frac{1}{2}.$$

Hence, the case $s = 4$ corresponds to a bigger threshold than the case $s = 2$, i.e., $\epsilon + \frac{1}{2}$. According to our runs, using $\frac{h}{4}(s = 4)$ is sufficient to remove such singularity, see Figure 2.21, while using $\frac{h}{2}(s = 2)$ it still suffers from the singularities. For this reason, we may continue reducing the time step as needed, although this could be costly. Therefore, we set a limit, allowing it to use up to $\frac{h}{4}$. In the few cases where $\frac{h}{4}$ cannot be of benefit, other approaches can be used.

Local Error (Consistency) of the PTS[1/1]($\frac{h}{s}$) Approach

Recall, first, the $LE_{[1/1]}$ (2.4.2). The local error for the PTS[1/1]($\frac{h}{s}$) approach will be a result from substituting $\frac{h}{s}$ (2.4.2) s times

$$LE_{[1/1](\frac{h}{s})} = \sum_{j=1}^{s} \left(\frac{(c_{2j})^2}{c_{1j}} - c_{3j} \right) h^3 + \mathcal{O}(h^4); \tag{2.4.9}$$

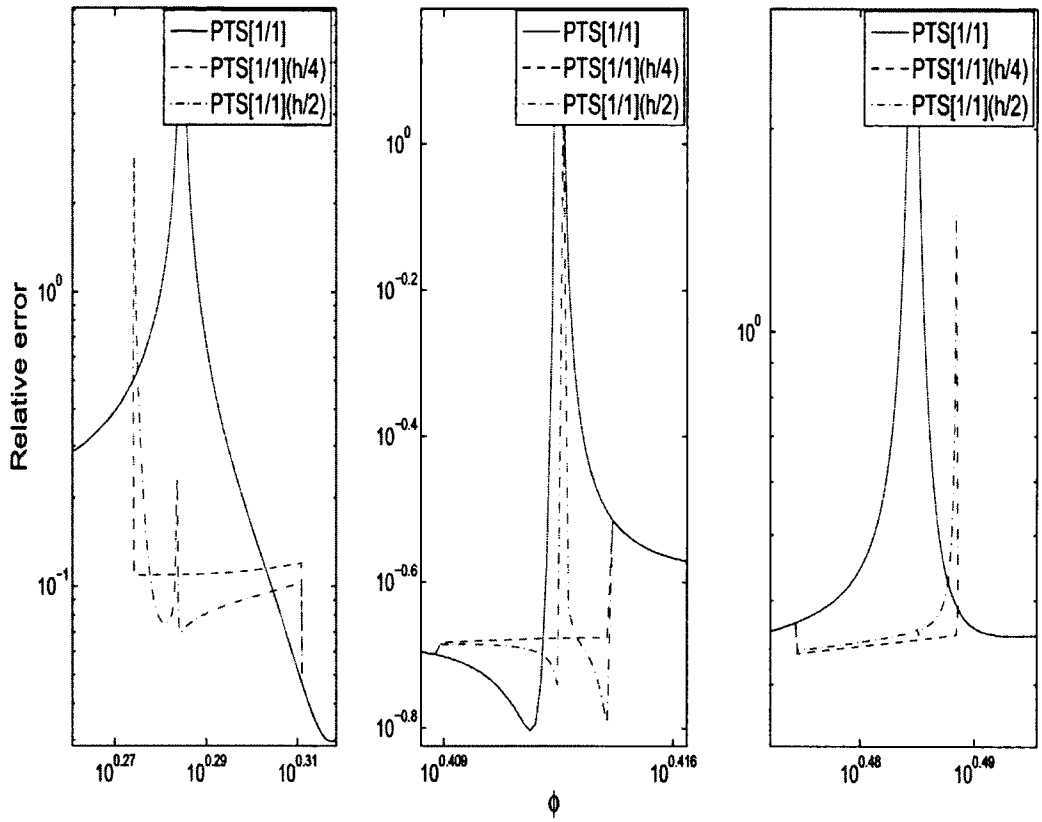

Figure 2.21: The first iteration of the PTS[1/1] approach at $\frac{h}{2}(s = 4)$ and $\frac{h}{4}(s = 4)$ in the ϕ and error space, $\theta = 2$, $\epsilon^* = 1$, and $h = 0.9$.

where c_{1j}, c_{2j}, c_{3j} are computed at time $t_{i+j/s}$. Then, the theoretical error formula for PTS[1/1]$(\frac{h}{s})$ will be a combination of $LE_{[1/1]}$ and $LE_{[1/1](\frac{h}{s})}$, depending on the number of singularities.

In addition, Figure 2.21 shows the local error result of the half step control on PTS[1/1]. It was not that very useful since parts of the peaks still exist. Using $\frac{h}{4}$, on the other hand, solves the issue for this specific case of h and θ. Figure 2.22 also indicates that the right order of accuracy, i.e., order three, is achieved.

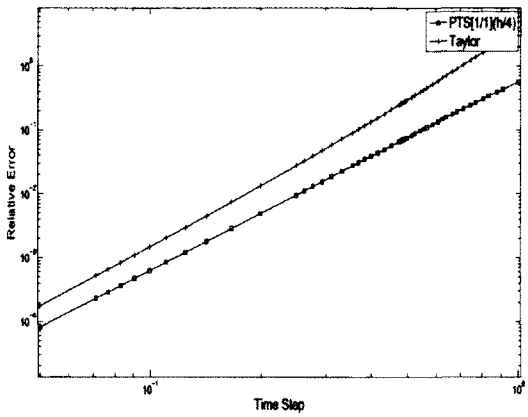

Figure 2.22: The first iteration of the PTS[1/1]($\frac{h}{2}$) approach in ϕ and error space.

Global Error (Stability) of the PTS[1/1]($\frac{h}{s}$) Approach

This part of the chapter explains the use of the full scheme to obtain an optimal LECT, ϵ^*, for the PTS[1/1]($\frac{h}{s}$) approach which, in turn, will establish our new stable scheme. It is difficult to study the value of the threshold for general s, therefore we limit ourself to the cases $s = 2$ and $s = 4$.

LECT, ϵ^*, of the PTS[1/1]($\frac{h}{s}$) Approach

As previously discussed in relation to ϵ, the optimal LECT, ϵ^*, is close to one but each approach leads to slightly different values. When searching for the optimal LECT, ϵ^*, for PTS[1/1]($\frac{h}{2}$), as shown in Figure 2.23, we can say that our scheme requires ϵ^* to be in $(0.95, 1.3)$. If we use a small time step control, say $\frac{h}{4}$, we find LECT gets smaller, which can be seen in Figure 2.24. However, there is an important factor that needs to be considered; the percentage of $\frac{h}{4}$ substep used in the approach. This is mainly a computational issue because using $\frac{h}{4}$ substeps involves a higher time cost, which needs to be accounted for. Taking this into consideration, the interval for optimality is found to be $\epsilon \in (0.95, 1)$. Figure 2.25 presents percentages of used substeps for each epsilon. Indeed, we set $\epsilon^* = 1$ for PTS[1/1]($\frac{h}{4}$) which, as seen in Figure 2.24,

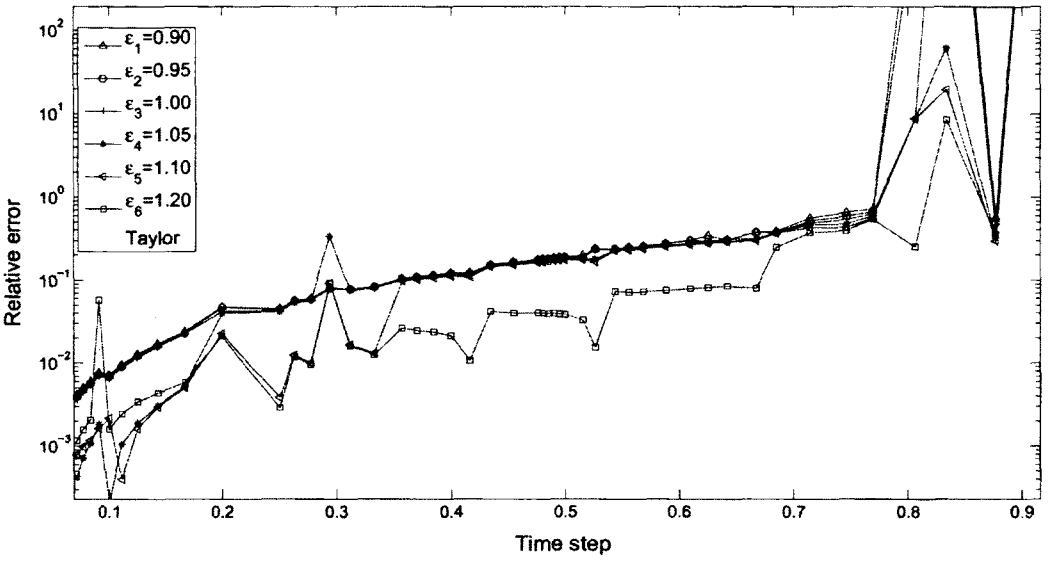

Figure 2.23: Different values of ϵ for the PTS[1/1]($\frac{h}{2}$) approach, with $\theta = 2$, $T_f = 50$, and $s = 2$.

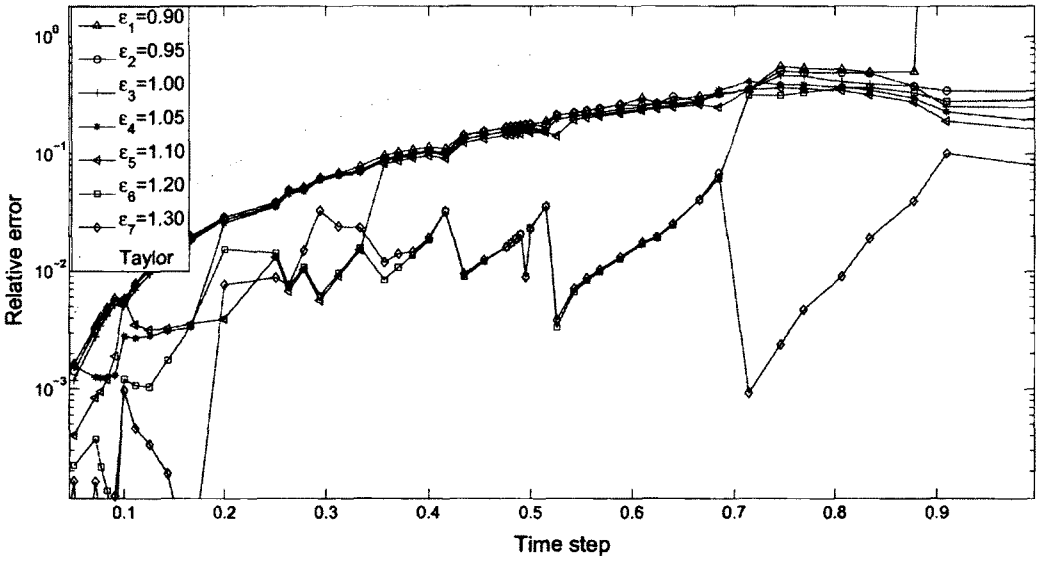

Figure 2.24: Different values of ϵ for the PTS[1/1]($\frac{h}{4}$) approach, with $\theta = 2$, $T_f = 50$, and $s = 4$.

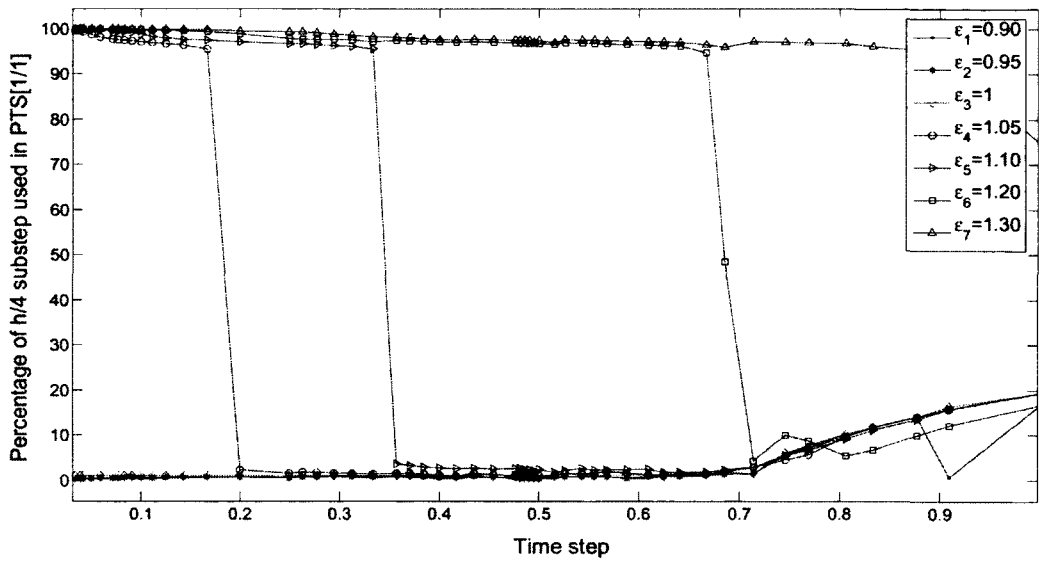

Figure 2.25: Plot of relative error versus percentage of $h/4$ substep used for different values of ϵ for PTS[1/1]($\frac{h}{4}$) approach, $\theta = 2$, $T_f = 50$, and $s = 4$.

yields the best performance amongst the ones considered.

2.5 Large Systems of ODEs

A small dimension case was discussed previously, we now turn to the the performance of different approaches of PTS[1/1] when a heat operator with a specific initial condition is considered. The comprehensive comparison for PTS[1/1] approaches with other competitive ones will be discussed in Chapter 4 when the reaction diffusion equation is solved. First, solving a large system of ODEs with PTS[1/1] approaches is studied. Secondly, the diffusion problem is considered and a modification for LECT is discussed. Finally, supporting examples are presented.

2.5.1 Solving ODEs with PTS[1/1] Approaches

We test the PTS[1/1] approaches on the full spatial discretization of a heat equation with zero boundary condition. Consider this ODE system

$$\mathbf{u}(t)_t = A\mathbf{u}(t), \qquad \mathbf{u}(0) = \mathbf{u}_0; \tag{2.5.1}$$

where $\mathbf{u} \in \mathbb{R}^m$, and

$$A = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -2 \end{bmatrix}_{m\times m}.$$

We test the performance of PTS[1/1] with the three new approaches on (2.5.1) with a large number of components and random initial conditions. We pick $m = 5000$ and 75 random initial conditions with $\|\mathbf{u}_0\| \leq 1$. Figure 2.26 shows good results for PTS[1/1] approaches, similar to the three component case solved in the previous section. It can also be noticed that the pure PTS[1/1] has a stability problem at $h = 0.7$ where a singularity exists.

On the other hand, we found that overall, the PTS$[1/1]_{\text{Taylor}}$, PTS$[1/1]_{[0/2]}$ and, PTS$[1/1](\frac{h}{s})$ $(s = 4)$ approaches have an excellent performance particularly for bigger time steps. This valuable property is greatly needed in many applications, i.e., with stiff problems. They also provide reasonable results for small step size. Consequently, this example supports the application of the new PTS approaches for the diffusion equation.

2.5.2 Solving the Full Diffusion Operator by PTS

Moving to the full PDE problem, the new technique discussed above will be applied to solve the diffusion equation in one dimension. We consider problem (1.0.1), when

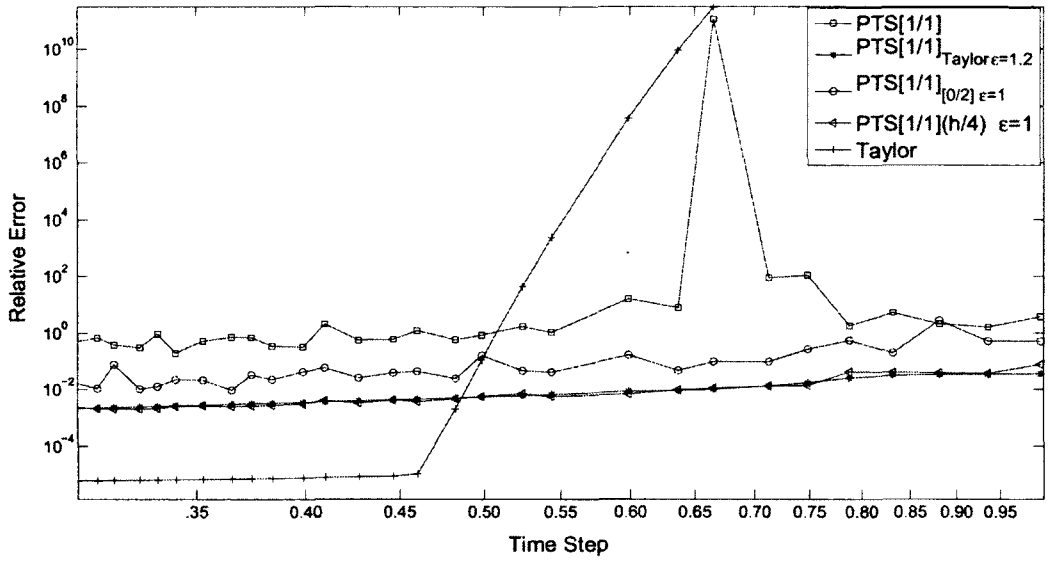

Figure 2.26: Plot of relative error versus time step h for ODEs (2.5.1), 5000 component and 75 random I.Cs.

$f(u) = 0$, that is

$$u_t = \alpha u_{xx}(x,t) \tag{2.5.2}$$

in the region $-l \leq x \leq l$, subject to the initial condition

$$u(x,t_0) = \exp\left(-x^2\right)$$

where $t_0 = 0$ with zero spatial boundary conditions. As is well known, the analytic solution is

$$u(x,t) = \frac{1}{\sqrt{1+4\alpha t}} \exp\left(\frac{-x^2}{\sqrt{1+4\alpha t}}\right).$$

Using spatial discretization, (2.5.2) becomes similar to (2.5.1) with coefficient matrix, $\left(\frac{\alpha}{dx^2}\right) A$, instead of A. We study a wide range of stiff problems by varying α and fixing both dx and h.

Due to the good performance of $\text{PTS}[1/1]_{\text{Taylor}}$ over $\text{PTS}[1/1]_{[0/2]}$, shown in Figure 2.26, and the expected computational cost for the PTS[1/1] $\left(\frac{h}{s}\right)$ approach, we limit

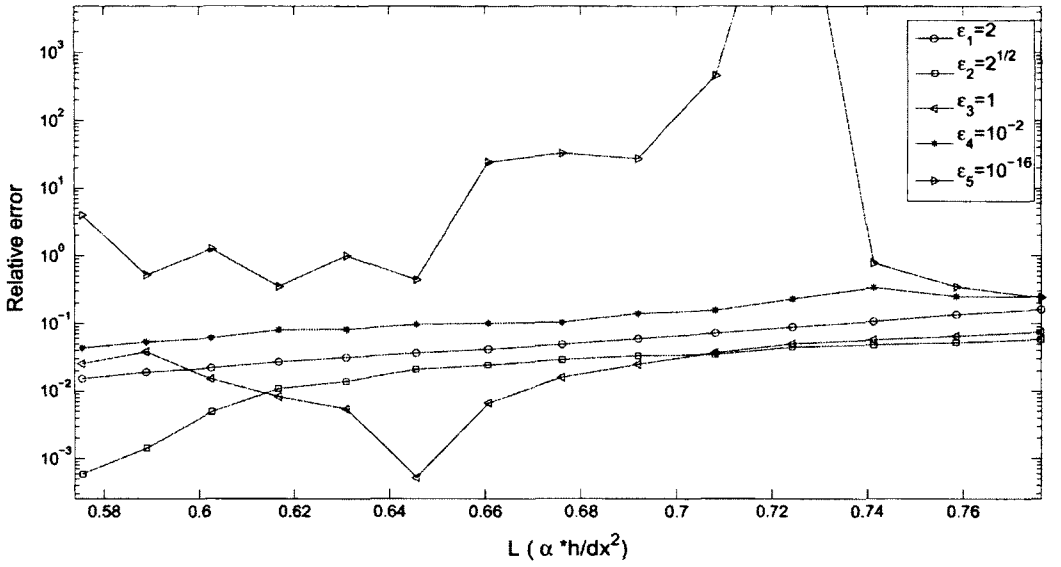

Figure 2.27: Plot of relative error versus stiff factor L for diffusion equation, $T_f = 0.002$.

ourselves here to PTS$[1/1]_{\text{Taylor}}$. One critical criteria of PTS[1/1] exists when $c_1 = 0$ and $c_2 \neq 0$. Recalling the PTS[1/1] scheme, we have

$$u_{n+1} = \frac{c_0 c_1 + (c_1^2 - c_0 c_2)\, L}{c_1 - c_2 L}.$$

This means that when $c_1 = 0$ and $c_2 \neq 0$ either we do not have an improvement or simply a fixed point exist, i.e., $u_{n+1} = c_0 = u_n$. Therefore, our numerical scheme should avoid this instance by applying Taylor instead of PTS. The appropriate condition is discussed below.

A LECT Modification for PTS$[1/1]_{\text{Taylor}}$

Here we search for the optimal value for LECT, ϵ^*, for the full diffusion problem (2.5.2). Figure 2.27 shows the performance of our scheme with different choices of ϵ. It is clear that $\epsilon^* = \sqrt{2}$ is the best choice. Indeed, the transition between two numbers close to $\sqrt{2}$ appears in Figure 2.28, which shows how sensitive is the dependence. In

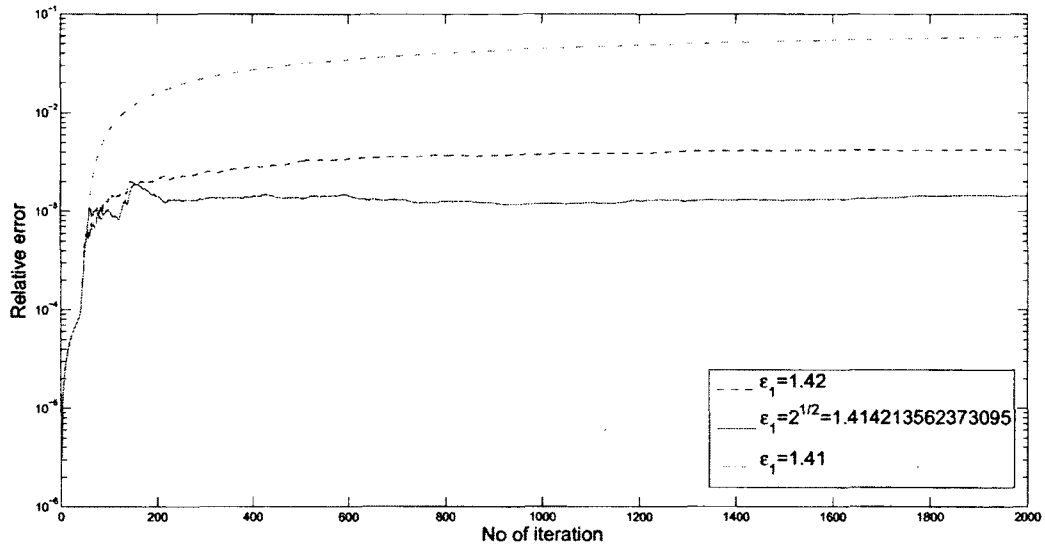

Figure 2.28: Plot of relative error versus number of iteration N for diffusion equation, $T_f = .002$ and $L = 0.5888$.

the previous section, when we set ϵ^* to be 1.2, we were unable to notice the fixed point behaviour as we did in the present example. Returning to our previous discussion, assuming a fixed point u^* exists at component j and iteration number n, then $R^* = \frac{c_2}{c_1}$ is a fixed ratio that leads to slow movement. This behaviour is demonstrated in Figure 2.29 by a dashed line. Following a number of trials, a second condition is required: if $\left|\frac{c_1}{c_0}\right| \leq 10^{-6}$, apply Taylor; this is clarified in Algorithm 2.5.1. The chosen threshold, 10^{-6}, was selected based on many trials. The second condition is important when $|D| > \epsilon$ and we are close to a fixed point.

Algorithm 2.5.1 PTS$[1/1]_{\text{Taylor}}$ for the diffusion part.

for $\mathbf{i = 1 : n}$

if $(|\mathbf{D_i}| \leq \epsilon_1)$ **OR** $\left(\left|\frac{\mathbf{c_{1i}}}{\mathbf{c_{0i}}}\right| \leq \epsilon_2\right)$; then apply Taylor

otherwise apply PTS

end

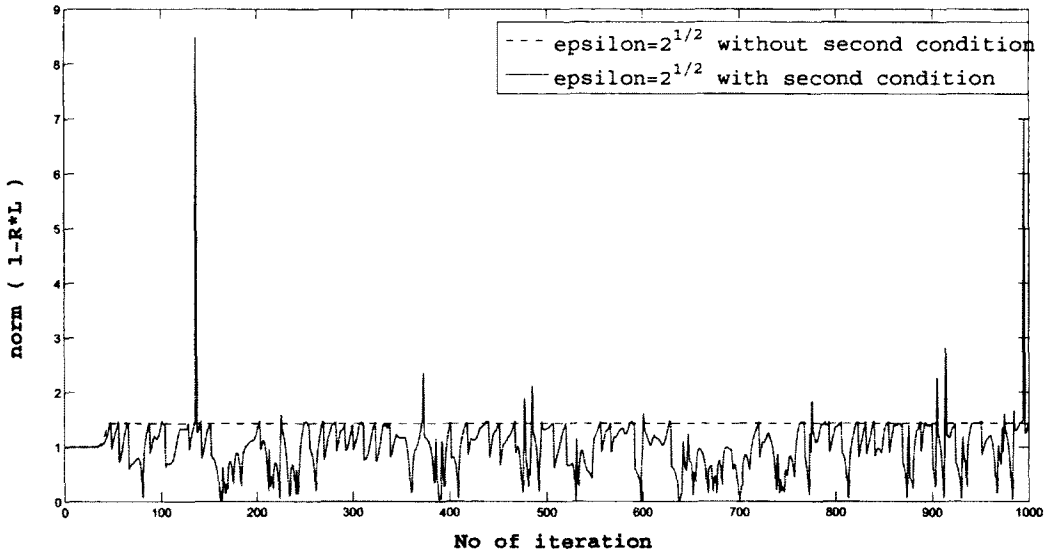

Figure 2.29: Plot of $|D|$ versus number of iteration N for diffusion equation, $T_f = 0.002$.

Algorithm 2.4.1 and Algorithm 2.5.1 differ mainly in the second conditions. The second condition in Algorithm 2.4.1 was based on reducing the number of Taylor used, while the second condition, in Algorithm 2.5.1, avoids the fixed point behaviour. In general, having a fixed point means we will not move, and thus, Taylor is used in sequence. Therefore, the second condition in both algorithms addresses the same issue. However, due to the number of conditions on each algorithm, the computational cost of Algorithm 2.5.1 is less than the computational cost of Algorithm 2.4.1. Good performance is observed when Algorithm 2.5.1 is applied with $\epsilon_1 = \sqrt{2}$ and $\epsilon_2 = 10^{-6}$.

2.5.3 Examples and Comparison

We numerically solve both stiff and non-stiff heat equations with well-chosen schemes of second order, such as Crank–Nicholson–Adam–Bashforth (CNAB), Adam-Bashforth $\gamma = 3/4$ (ABg), RK2, RKC2, and the new PTS$[1/1]_{\text{Taylor}}$ approach.

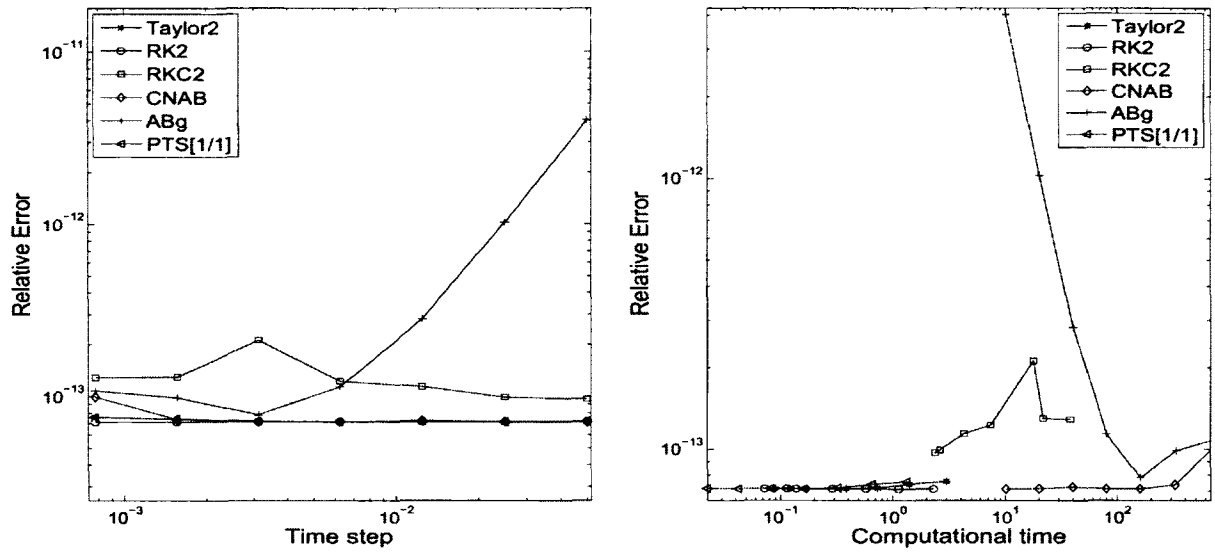

Figure 2.30: Plot of relative error versus computational time for diffusion equation with a Gaussian initial condition and $t_0 = 0$, $t_f = 1$; $x \in [-20, 20]$, $dx = 0.005$, $\frac{\alpha}{dx^2} = 1$, non-stiff case.

Non-stiff Case of Diffusion Equation

First, the non-stiff case when $\frac{\alpha}{dx^2} = 1$, is solved. Figure 2.30 shows that $\text{PTS}[1/1]_{\text{Taylor}}$ provides a good level of accuracy with much less computational time. This suggests an advantage over other approaches in such cases.

Stiff Case of Diffusion Equation

Now the stiff case, when $\frac{\alpha}{dx^2}$ is very large, is introduced. Figure 2.31 illustrates good performance for the new scheme. The $\text{PTS}[1/1]_{\text{Taylor}}$ approach produces a relative error of $10^{-1.5}$ in 10^{-2} seconds where no other scheme gives a comparable result. In addition, for higher accuracy, the $\text{PTS}[1/1]_{\text{Taylor}}$ approach provides a relative error of 10^{-11} in half of the computation time of the explicit RK2 and it is also 230 times faster than the implicit Crank–Nicholson–Adams–Bashforth scheme. In conclusion, the proposed scheme, PTS[1/1], shows good performance especially for strongly diffusive equations.

2.6 Summary

In this chapter, the diffusion operator was solved by the PTS[1/1] scheme. A large number of initial conditions was considered in the analysis. The PTS[1/1] showed better results, although some controls need to be imposed in order to account for possible singularities. We introduced the local error control threshold (LECT) and then different approaches were studied, namely Taylor, PTS[0/2] and PTS[1/1] $\left(\frac{h}{s}\right)$.

First, the cases when $c_0 = c_1 = c_2 = 0$ were discussed. The $\text{PTS}[1/1]_{\text{Taylor}}$, $\text{PTS}[1/1]_{[0/2]}$ and PTS[1/1] $\left(\frac{h}{s}\right)$ approaches are not applicable in these instances, i.e., fixed points. The value of LECT, ϵ^*, for $\text{PTS}[1/1]_{[0/2]}$ and PTS[1/1] $\left(\frac{h}{s}\right)$ is set up to be 1. Both the $\text{PTS}[1/1]_{[0/2]}$ and PTS[1/1] $\left(\frac{h}{s}\right)$ approaches are not completely understood and they need more detailed analysis. However, based on some examples, $\text{PTS}[1/1]_{\text{Taylor}}$ perform better than $\text{PTS}[1/1]_{[0/2]}$ and it is computationally faster than PTS[1/1] $\left(\frac{h}{s}\right)$.

The $\text{PTS}[1/1]_{\text{Taylor}}$ is studied and a modified approach is considered with new condition shown in Algorithm 2.4.1. Thirdly, the case when $\frac{\alpha}{dx^2}$ is very large number, is investigated. The optimal LECT is $\epsilon^* = \sqrt{2}$ for the $\text{PTS}[1/1]_{\text{Taylor}}$ approach. Since the value of c_1 needed to be checked accordingly, a criteria is updated in the approach as in Algorithm 2.5.1.

Finally, the $\text{PTS}[1/1]_{\text{Taylor}}$ approach is well-studied here for diffusion equations. The $\text{PTS}[1/1]_{\text{Taylor}}$ approach has good performance, especially when bigger time steps are used. Lastly, the PTS[1/1] approach in general offers a favourable alternative to other explicit and implicit schemes in the context of a linear diffusion problem. The performance for general nonlinear and reaction diffusion type problems will be discussed in the following chapters.

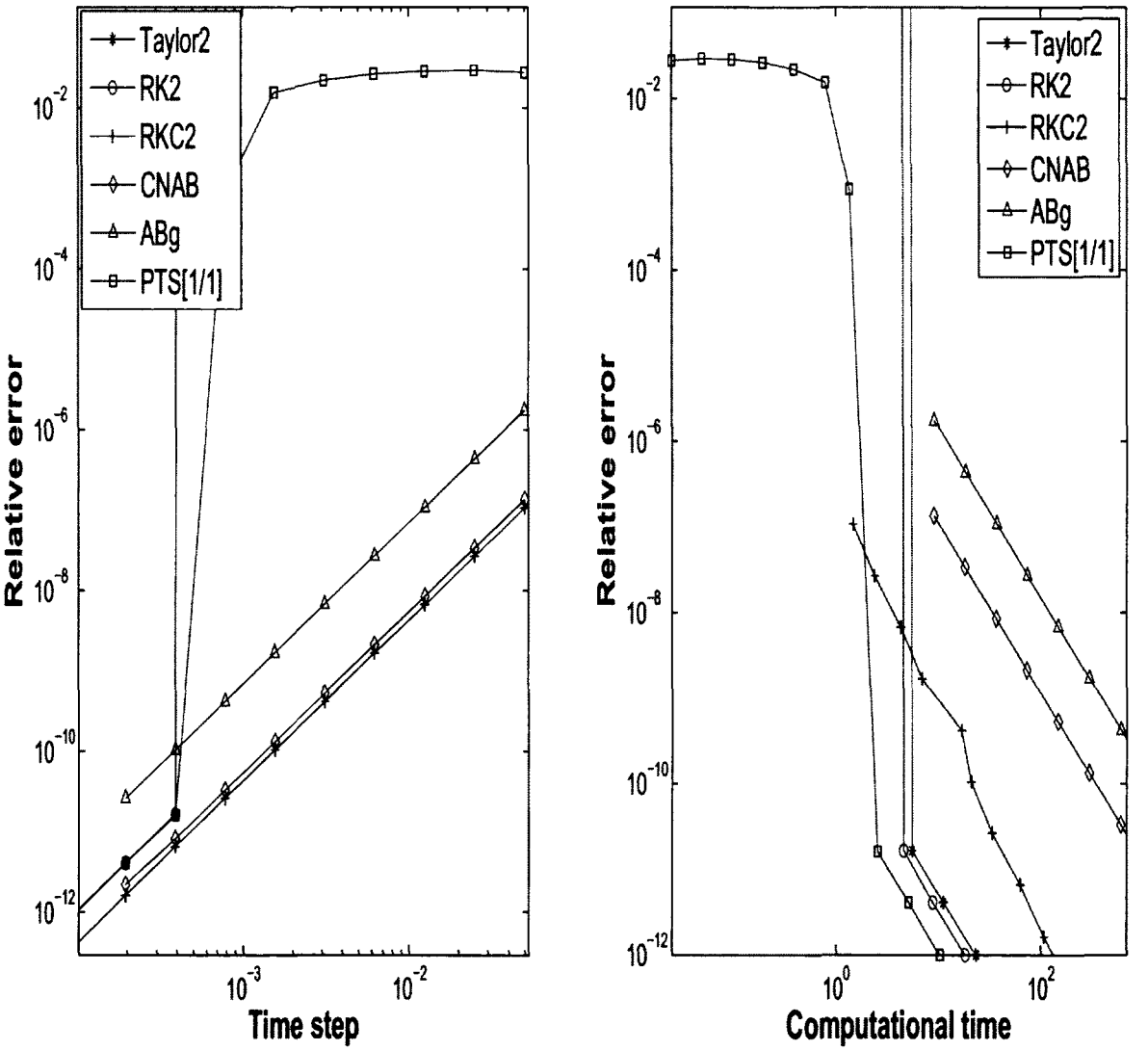

Figure 2.31: Plot of relative error versus computational time for the diffusion equation with a Gaussian initial condition and $t_0 = 0$, $t_f = 1$; $x \in [-20, 20]$, $dx = 0.005$, $\frac{\alpha}{dx^2} = 800$, stiff case.

Chapter 3

Padé Time Stepping (PTS) on Scalar Autonomous Differential Equations

In the previous chapter, the new PTS[1/1] approaches showed excellent performances when the heat operator was considered. Now, we consider their performance on nonlinear ODEs. In most applications, the reaction term in the studied model (1.0.1) is simply a polynomial in u [54, 55]. Therefore, we consider scalar autonomous differential equations. This chapter discusses solving such equations with the PTS[1/1] scheme. First, the application of PTS[1/1] to these equations is introduced, paying particular attention to the existence of singularities. A general stability analysis for the PTS[1/1] scheme is provided for logistic and more general equations. Both stiff and non-stiff cases are considered. Also, the use of the previous controls for PTS[1/1] is investigated. The presented approaches, $\text{PTS}[1/1]_{\text{Taylor}}$, $\text{PTS}[1/1]_{[0/2]}$, and $\text{PTS}[1/1](\frac{h}{s})$, behave very well when non-stiff logistic equations are considered. The case of stiff logistic equation is also analyzed. Supporting examples are presented to show the performance of PTS[1/1] compared to explicit schemes.

3.1 PTS[1/1] and its Singularities

Consider this differential equation

$$u_t = f(u); \tag{3.1.1}$$

where $u(t_0)$ is a given initial condition. By considering a polynomial $f(u) = a_1u + a_2u^2 + \cdots + a_nu^n$, a wide range of nonlinear reaction terms are covered, such as Fisher [20], Nagumo [21], etc. It is important to identify the type and locations of singularities when PTS[1/1] is applied, as well as the ways to control them.

The solution to equation (3.1.1) by PTS[1/1] can be written in terms of the Taylor coefficients as

$$u_{i+1} = \frac{u_i + [-(1/2)f_u(u_i)u_i + f(u_i)]\,h}{1 + [-(1/2)f_u(u_i)]\,h}. \tag{3.1.2}$$

A singularity exists at $h = \frac{2}{f_u(u_i)}$, where $f_u(u) = a_1 + 2a_2u + \cdots + na_nu^{n-1}$. Therefore a necessary condition is $f_u(u_i) > 0$, in order for a singularity to exist.

From an application point of view [55], most models are of degree three or less. Consider this general term as in [30],

$$f(u) = u\left(\mu + \beta u - \gamma u^2\right); \tag{3.1.3}$$

for real numbers μ, β, and γ. For example, for $\mu = 1$, $\beta = -1$, and $\gamma = 0$ the reaction term of the Fisher equation would be obtained. For $\mu = 1$, $\beta = 0$, and $\gamma = 1$ the reaction term of Nagumo equation is found. Using PTS[1/1] to solve (3.1.1) with $f(u)$ as in (3.1.3), the iterated solution would be

$$u_{i+1} = \frac{u_i - (1/2)u_i\left(\mu + \gamma u_i^2\right)h}{1 - (1/2)\left(\mu + 2\beta u_i - 3\gamma u_i^2\right)h}. \tag{3.1.4}$$

A singularity of (3.1.4) exists at $h = \frac{2}{\mu + 2\beta u_i - 3\gamma u_i^2}$. In the case of ($\mu = 1$, $\beta = -1$, and $\gamma = 0$), $h = \frac{2}{1-2u_i}$, no singularity exists if $u_i > \frac{1}{2}$. Some cases of (3.1.4) will be

introduced here to clarify the expected deficiency of the pure PTS[1/1]. For example, for $\gamma = 0$ in (3.1.3), there exists a singularity at some h for some real numbers μ and β. Another example, when $\beta = 0$ in (3.1.3), there exists a singularity at some point of h if both μ and γ have the same sign. Therefore, the same approaches discussed previously in Chapter 2 are needed to control these singularities.

In order to illustrate the application of the PTS[1/1]$_{\text{Taylor}}$, PTS[1/1]$_{[0/2]}$, and PTS[1/1]$(\frac{h}{4})$ approaches, the simplest form of the Ricatti ODE [90], i.e., the logistic ODE (3.1.1)

$$f(u) = Ru(1-u) \tag{3.1.5}$$

is considered ((3.1.3) with $\mu = \gamma = 1$). For this ODE, the exact solution is

$$u(t) = \frac{u_0}{u_0 + (1-u_0)\exp(-Rt)}, \tag{3.1.6}$$

where $u(0) = u_0$ is the initial condition. Applying the values of μ and γ to (3.1.4), the solution using PTS[1/1] would be

$$u_{i+1} = \frac{u_i + \frac{1}{2}Rhu_i}{1 - \frac{1}{2}(1-2u_i)\,Rh}. \tag{3.1.7}$$

The singularity exists at $h = \frac{1}{R(\frac{1}{2}-u_i)}$ (or $u_i = \frac{Rh-2}{2Rh}$). This example is used for the stability analysis in the next section.

3.2 Stability Analysis

Consider the logistic equation (3.1.5) with $R = 1$. The first iteration of PTS[1/1] will be

$$u_1 = \frac{(2+h)\,u_0}{(2-h)+2hu_0} = A_1(u_0, h)u_0.$$

The n^{th} iteration will be

$$u_1 = \frac{(2+h)\,u_0}{(2-h)+2hu_0}$$

$$\begin{aligned} u_2 &= \frac{(2+h)^2 u_0}{(2-h)^2 + 8hu_0} \\ u_3 &= \frac{(2+h)^3 u_0}{(2-h)^3 + (24h + 2h^3) u_0} \\ &\vdots \\ u_n &= \frac{(2+h)^n u_0}{(2-h)^n + n2^n h u_0 + \cdots + 2h^n u_0} \\ &= A_n(u_0; h) u_0, \end{aligned} \tag{3.2.1}$$

where $A_n(u_0; h) = \frac{(2+h)^n}{(2-h)^n + n2^n h u_0 + \cdots + 2h^n u_0}$ is the growth factor.

First, for $0 \leq h \leq 1$, we notice that each $A_i(u_{i-1}; h)$ is bounded by a constant, i.e., $\left|\frac{(2+h)}{(2-h)+2hu_0}\right| \leq 3$ when no singularities exist. If there is a singularity, $u_i = \frac{h-2}{2h}$, then $A_i = 1 + h + \frac{h^2}{2}$, which is bounded. We also need to check the previous growth factor when the PTS is applied, A_{i-1}, which can be written as

$$A_{i-1} = \frac{u_i}{u_{i-1}} = \frac{\left(\frac{h-2}{2h}\right)}{u_{i-1}}.$$

So,

$$|A_{i-1}| = \left|\left(\frac{h-2}{2h}\right)\right| \left|\frac{1}{u_{i-1}}\right| \leq \frac{1}{|2u_{i-1}|},$$

which is bounded since $u_{i-1} \neq 0$.

Secondly, we show the boundedness of $A_n(u_0; h)$ as $n \to \infty$. The critical points of $A_n(u_0; h)$ are $h = -2$ and 2. By checking the sign of $A_u(u)$, we find that $A_n(u_0; h)$ is increasing in $h \in [0, 2]$ and $A_i(u_0; 2)$ leads to the maximum for all $i = 1, \ldots, n$.

$$\lim_{n\to\infty} A_n(u_0; 1) = \lim_{n\to\infty} \left(\frac{3^n}{1 + (3^n - 1)u_0}\right) = \frac{1}{u_0}. \tag{3.2.2}$$

Therefore, u_n approaches one as $n \to \infty$ which verifies the stability property of the PTS[1/1] scheme. Figure 3.1 shows a good behaviour for PTS[1/1]. Figure 3.2 introduces a case where Taylor and RK2 become unstable while PTS[1/1] maintains good stability criteria.

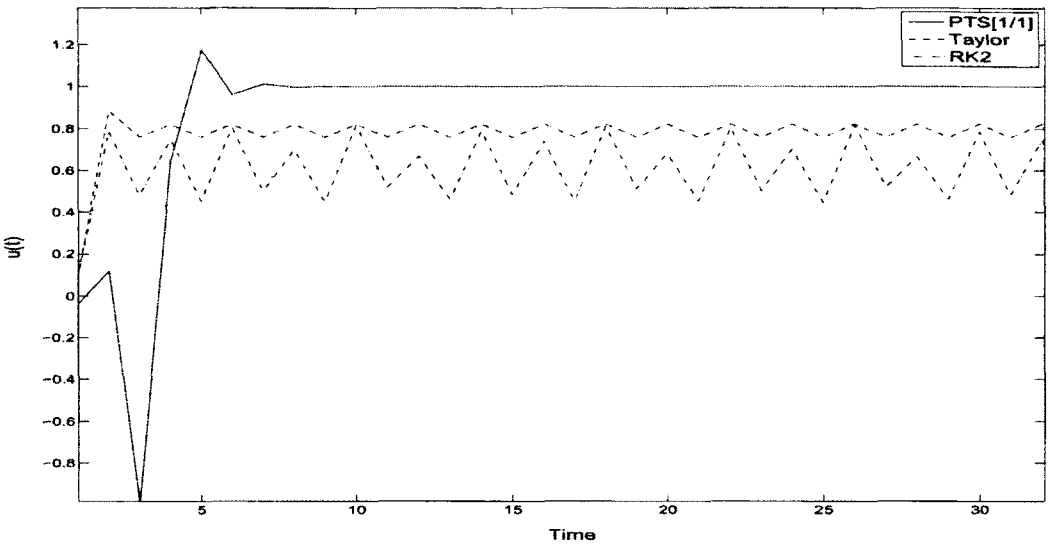

Figure 3.1: Plot of $u(t)$ versus time for non-stiff logistic equation when $R = 350$, $t_f = 10$, and $u_0 = 0.01$ using PTS[1/1], Taylor and RK2, $h = 0.01$.

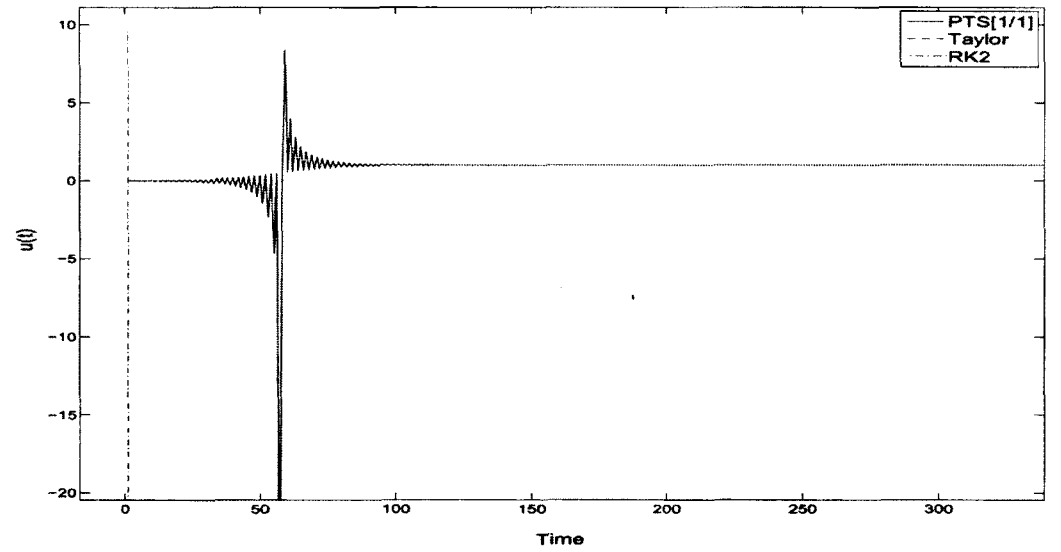

Figure 3.2: Plot of $u(t)$ versus time for stiff logistic equation when $R = 5000$, $t_f = 10$, and $u_0 = 0.01$ using PTS[1/1], Taylor and RK2, $h = 0.01$.

This work can be generalized to a polynomial $f(u)$ of degree n with real zeros. By considering polynomials, a large class of functions will be covered, which can be approximated by polynomials. A generalization of the previous work will be stated in the following theorem.

Theorem 3.2.1 Let $f(u) = (u - a_1)(a_2 - u)(a_3 - u)\dots(a_n - u)$ with $0 \leq a_1 < a_2 < \cdots < a_n$, for the nonlinear ODEs

$$u_t = f(u) \tag{3.2.3}$$

and

$$u_t = -f(u), \tag{3.2.4}$$

with $u_0 \in (a_1, a_2)$ as initial condition. Each iteration of the PTS[1/1] scheme for (3.2.3 and 3.2.4) is bounded and the numerical solution u_n for (3.2.3 and 3.2.4) tend to a_2 and a_1, respectively, as $n \to \infty$.

Proof 3.2.2 Consider first equation (3.2.3). Without loss of generality, we set $a_1 = 0$. For $u \in (a_1, a_2)$, we rewrite

$$f(u) = R(u)(u - a_1)(a_2 - u),$$

where $R(u) = (a_3 - u)\dots(a_n - u) > 0$. First, each $A_i(u_0; h)$ is bounded away from the singularities. Secondly, we show the limit goes to a_2 as $n \to \infty$.

The maximum value for $R(u)$ when $u \in (a_1, a_2)$ is $R_{\max} = R(a_1)$ and the minimum value is $R_{\min} = R(a_2)$. Set

$$f_{\max}(u) = R_{\max} u(a_2 - u)$$

and

$$f_{\min}(u) = R_{\min} u(a_2 - u).$$

We clearly have

$$f_{\min}(u) \leq f(u) \leq f_{\max}(u),$$

for all $u \in (a_1, a_2)$. A lower bound solution by PTS[1/1], solution of $u_t = f_{\min}(u)$, will be

$$u_n = \frac{(2 + R_{\min} h a_2)^n u_0}{(2 - R_{\min} h a_2)^n + n 2^n R_{\min} h u_0 + \cdots + 2 R_{\min}^n h^n a_2^{n-1} u_0}.$$

Here, the limit of $A_n(u_0; h)$ will be

$$\lim_{n\to\infty} A_n(u_0; \frac{2}{R_{\min} a_2}) = \lim_{n\to\infty} \left(\frac{4^n}{0 + \frac{(4^n - c_n)}{a_2} u_0} \right) = \frac{a_2}{u_0},$$

where $c_n \to 0$ as $n \to \infty$.

Similarly, an upper bound solution by PTS[1/1], solution of $u_t = f_{\max}(u)$, will be written as

$$u_n = \frac{(2 + R_{\max} h a_2)^n u_0}{(2 - R_{\max} h a_2)^n + n 2^n R_{\max} h u_0 + \cdots + 2 R_{\max}^n h^n a_2^{n-1} u_0}.$$

Studying the limit of A_n, we have

$$\lim_{n\to\infty} A_n(u_0; \frac{2}{R_{\max} a_2}) = \lim_{n\to\infty} \left(\frac{4^n}{0 + \frac{(4^n - c_n)}{a_2} u_0} \right) = \frac{a_2}{u_0},$$

where $c_n \to 0$ as $n \to \infty$. In both cases, the numerical solution, u_n, is bounded by a_2 as $n \to \infty$. By the sandwich theorem, the numerical solution, u_n, of $u_t = f(u)$ using PTS[1/1], is bounded by a_2 as $n \to \infty$. This verifies the stability property of the PTS[1/1] scheme applied to equation (3.2.3) and completes the proof of the first part.

Now, consider the second equation (3.2.4) with u_0 as initial condition and $0 = a_1 < a_2 < \cdots < a_n$. For $u_0 \in (a_1, a_2)$, equation (3.2.4) will be

$$f(u) = R(u)(u - a_1)(a_2 - u),$$

where $R(u) < 0$. By repeating the same analysis as we did for equation (3.2.3), we found that the numerical solution, u_n, of $u_t = -f(u)$ using PTS[1/1] is bounded by a_1 as $n \to \infty$. This completes the proof of the theorem. □

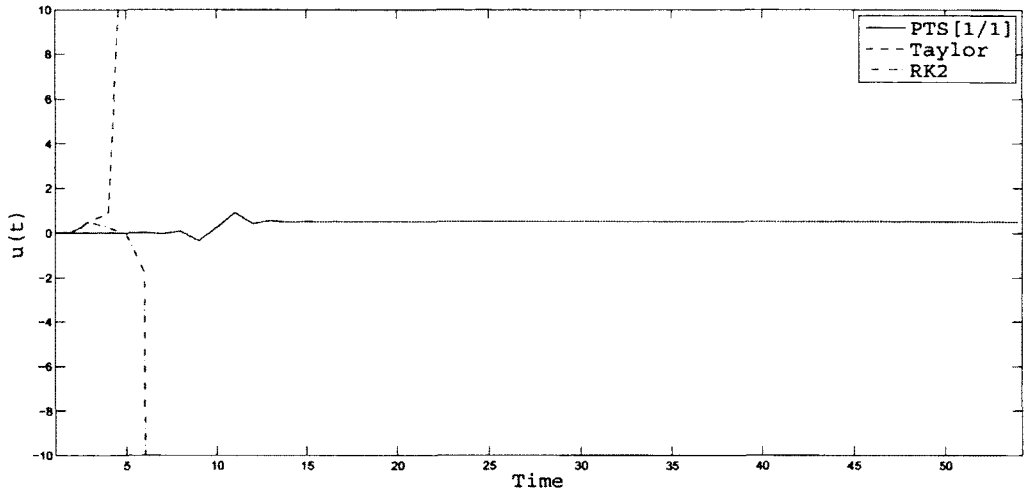

Figure 3.3: Plot of $u(t)$ versus time for nonlinear ODE using PTS[1/1], Taylor and RK2, $a_1 = 0$, $a_2 = 0.5$, $a_3 = 1000$, $h = 0.01$.

One example of a nonlinear ODE is this polynomial $f(u) = (u-a_1)(a_2-u)(a_3-u)$ with appropriate choices of a_1, a_2 and a_3. Figure 3.3, shows a good stability result for PTS[1/1]. This completes the discussion of the stability of the PTS[1/1] scheme for nonlinear ODEs (3.2.3) and (3.2.4).

3.3 Local Error Control Threshold; ϵ^*

The logistic equation (3.1.5) is considered to study PTS[1/1] controllers. For analysis purposes, $u_0 = \frac{Rh-2}{2Rh}$ is selected to ensure that there is at least one singularity for each time step h.

First, the non-stiff case, $R = 1$, was studied. No strict condition on LECT, ϵ^*, is noticed, i.e., $10^{-16} \leq \epsilon < \infty$, which is expected since Taylor is working. Figure 3.4 shows good results for the new approaches. Using the same values of the chosen ϵ^* previously used in Chapter 2, satisfactory outputs were obtained. That supports our

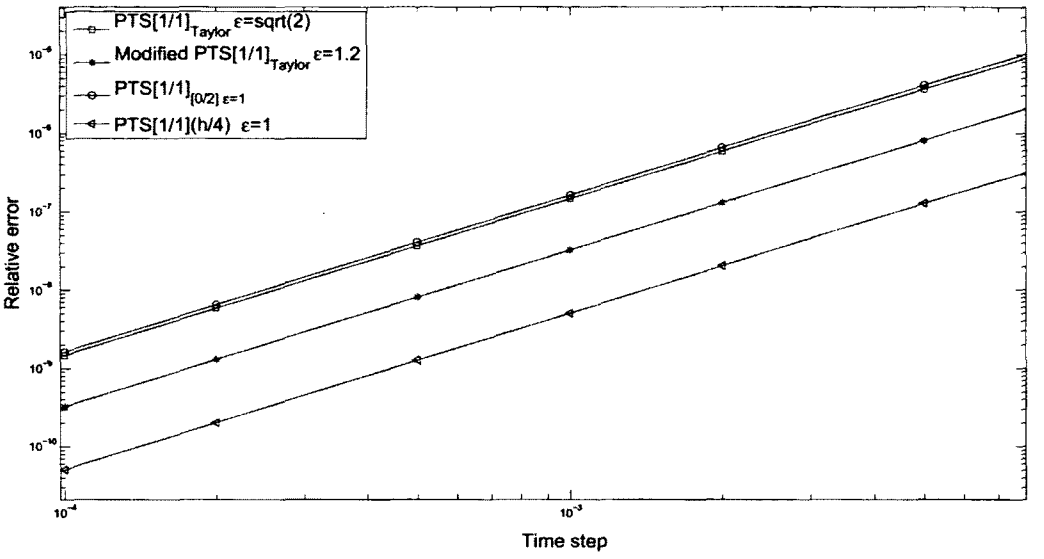

Figure 3.4: Plot of relative error versus time step h for the logistic equation, non-stiff case, $t_0 = 0$, $t_f = 1$, $R = 1$, $u_0 = .01$.

choices for optimal LECT when the heat operator is analyzed.

Secondly, the use of both $\text{PTS}[1/1]_{\text{Taylor}}$ and $\text{PTS}[1/1]_{[0/2]}$ approaches for a stiff case is introduced. The $\text{PTS}[1/1](\frac{h}{s})$ approach would also be expected to behave well with stiff problems, however, it would require a high computational cost. For this reason, we will consider the $\text{PTS}[1/1]_{\text{Taylor}}$ and $\text{PTS}[1/1]_{[0/2]}$ approaches in the following discussion.

3.3.1 $\text{PTS}[1/1]_{\text{Taylor}}$ Approach for Stiff Logistic Equation

Let us start by studying the effect of stiffness on the PTS[1/1] scheme. Consider the first iteration of PTS[1/1] applied to solve equation (3.1.1) with (3.1.5)

$$u_1 = \frac{(2 + Rh)u_0}{(2 - Rh) + 2Rhu_0}.$$

Some initial conditions $u_0 \in (0, 1)$ will be picked. Figure 3.5 demonstrates the expected convergence of PTS[1/1] scheme on a stiff problem. Basically, u_i approaches

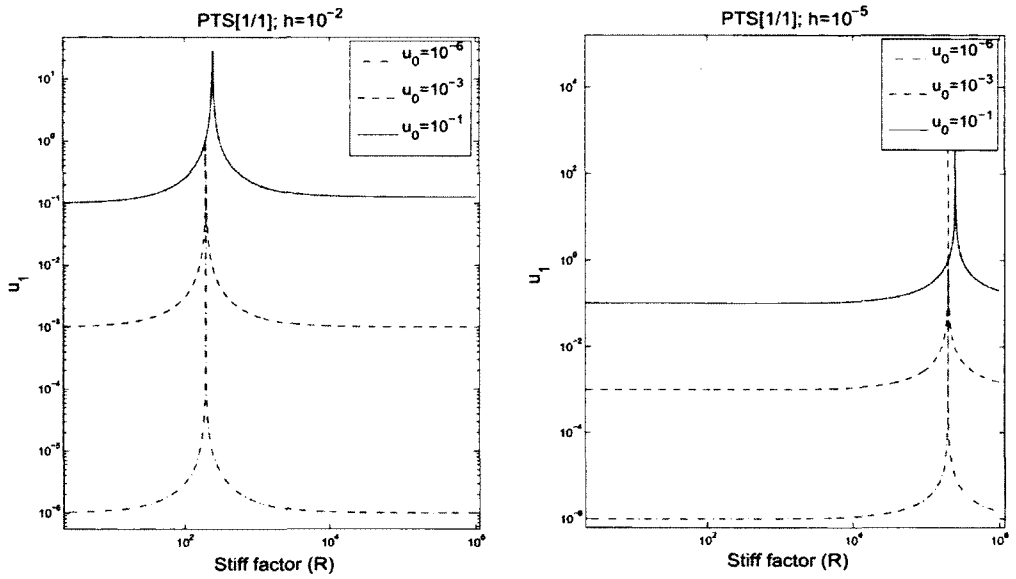

Figure 3.5: First iteration of PTS[1/1] on the logistic equation with various initial conditions.

1 for both stiff and non-stiff cases. It also illustrates the need for a controller for the PTS[1/1] scheme at the singularities.

The $\text{PTS}[1/1]_{\text{Taylor}}$ approach will be tested for different values of thresholds to determine the best choice for stiff logistic equation (large R). In order to ensure a worst case scenario, the initial condition will be forced to start at singularities, i.e., $u_0 = \frac{Rh-2}{2Rh}$. Then, Taylor is used at least once to control PTS[1/1] at the time of the singularity. The following parameters are chosen: $t_0 = 0$, $t_f = 0.1$ and $R = 10^6$. A range of threshold between 0 and 10 was implemented. The resulting plot, Figure 3.6, suggests $\epsilon^* \in [10^{-10}, 2]$. This choice matches ϵ^* in Chapter 2.

Figure 3.7 provides us with a case where Taylor is not stable, while $\text{PTS}[1/1]_{\text{Taylor}}$ provides a bounded and convergent approximation.

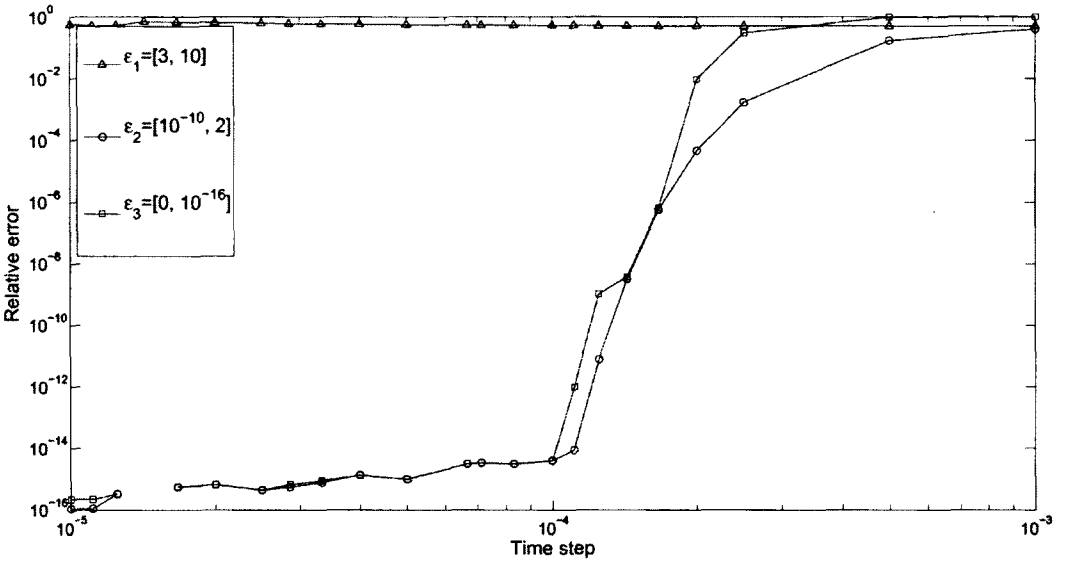

Figure 3.6: Plot of of relative error versus time step h for solving logistic equation with various epsilons using $\text{PTS}[1/1]_{\text{Taylor}}$ when $t_0 = 0$, $t_f = 0.1$, and $R = 10^6$.

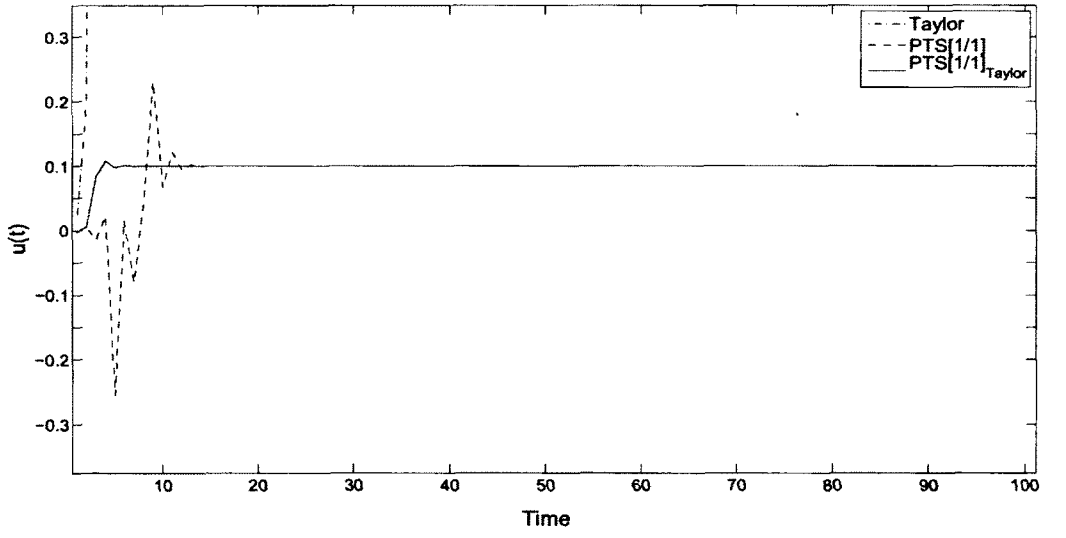

Figure 3.7: Plot of $u(t)$ versus time for nonlinear ODE using PTS[1/1], Taylor and $\text{PTS}[1/1]_{\text{Taylor}}$, , $a_1 = 0$, $a_2 = 0.1$, $a_3 = 1$, $a_4 = 5$, $a_5 = 100$, $u_0 = 0.001$.

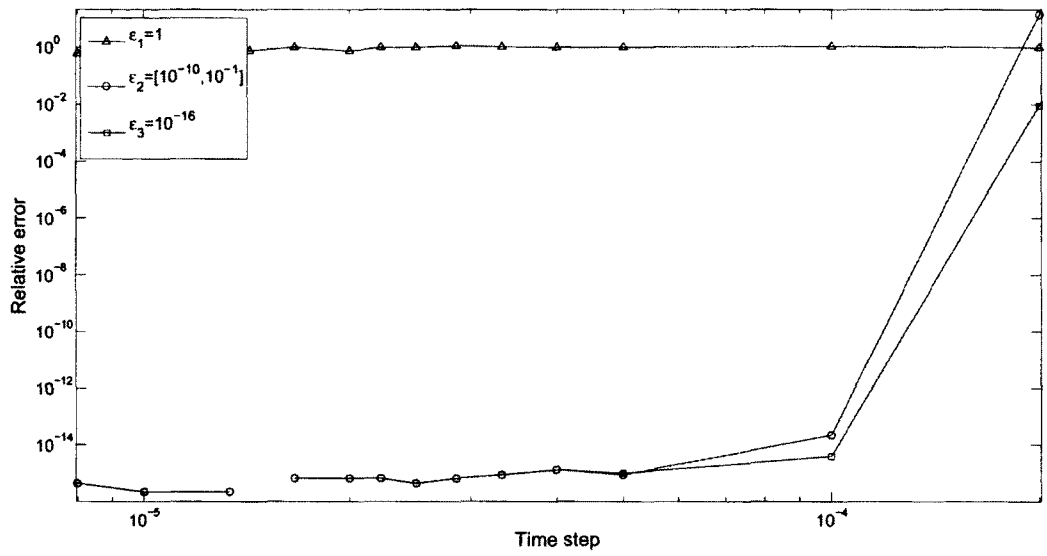

Figure 3.8: Plot of relative error versus time step h for solving stiff logistic equation with different epsilons using the PTS$[1/1]_{[0/2]}$ approach when $t_0 = 0$, $t_f = 0.1$, $a_2 = 1$, and $R = 10^6$.

3.3.2 PTS$[1/1]_{[0/2]}$ Approach for Stiff Logistic Equation

We will repeat the same test as in the last subsection for PTS$[1/1]_{[0/2]}$. The same initial condition, $u_0 = \frac{Rh-2}{2Rh}$, is imposed with the same stiffness factor $R = 10^6$. Figure 3.8 indicates PTS$[1/1]_{[0/2]}$ is not as effective as a controller. It can be noticed how ϵ^* is much smaller and in particular close to 10^{-16}. This value of LECT suggests that PTS[0/2] is unstable for stiff problems.

To illustrate this further, the first iteration of PTS[0/2] scheme can be written as

$$u_1 = \frac{2\,u_0}{2 - 2\,Rh + h^2R^2 + (2\,Rh - h^2R^2)u_0}. \tag{3.3.1}$$

This map will be tested over an interval of stiffness factors. Figure 3.9 shows that the iterated solutions by the PTS[0/2] scheme become inaccurate as R gets larger, i.e., the denominator approaches infinity while the nominator does not. This is a

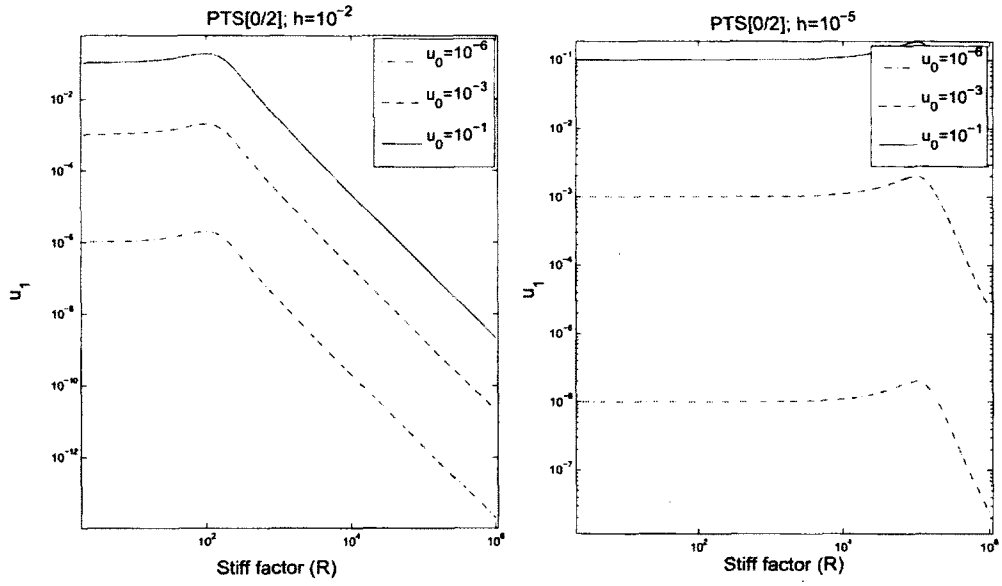

Figure 3.9: First iteration of PTS[0/2] on the logistic equation with various initial conditions.

disadvantage of using PTS[0/2] for solving stiff ODEs by a big time step. However, the PTS$[1/1]_{[0/2]}$ approach works well for mild-stiff differential equations. Figure 3.10 clarifies the values of the LECT $\epsilon^* \in (10^{-14}, 1)$. Therefore, a similar value for ϵ^* as in Chapter 2 will be set for the PTS$[1/1]_{[0/2]}$ approach in these cases.

Comparison

Both PTS$[1/1]_{\text{Taylor}}$ and PTS$[1/1]_{[0/2]}$ approaches are good solvers for non-stiff and mild-stiff ODEs. For this case, the LECTs are similar to the ones selected in Chapter 2. On the other hand, for stiff ODEs, the PTS$[1/1]_{[0/2]}$ approach develops instability for large time step. Thus, PTS$[1/1]_{\text{Taylor}}$ is recommended for stiff problem with LECT $\epsilon^* = \sqrt{2}$.

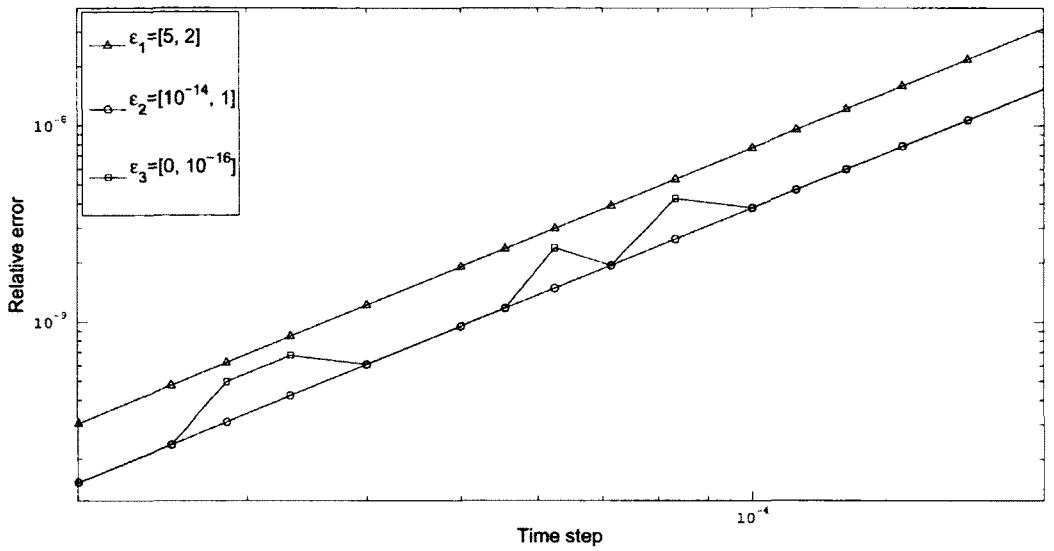

Figure 3.10: Plot of relative error versus time step h for solving mild logistic equation with various epsilons using the PTS$[1/1]_{[0/2]}$ approach when $t_0 = 0$, $t_f = 0.1$, $a_2 = 1$, and $R = 10^3$.

3.4 Summary

In this chapter, solving the reaction of model (1.0.1) with PTS[1/1] is discussed. We focused mainly on the case where the reaction term, $f(u)$, is a polynomial; however, the conclusions are applicable to more general choices for $f(u)$.

This nonlinear scheme, PTS[1/1], turns out to have good stability criteria when reaction ODEs are solved. Some singularities exist and may be controlled by either Taylor, PTS[0/2] or PTS$[1/1](\frac{h}{s})$.

The PTS$[1/1]_{\text{Taylor}}$ approach is recommended for the stiff case while PTS$[1/1]_{[0/2]}$ is not. The optimum LECT, ϵ^*, for all approaches are found to be $\epsilon^* = O(1)$, similar to that found for the diffusion operator in Chapter 2.

Chapter 4

Padé Time Stepping (PTS) on Reaction Diffusion (RD) Equation

From the introductory part, we know that all explicit methods have restricted boundary stability regions. Hence, explicit methods are generally inefficient for solving stiff problems. Implicit schemes offer greater stability but they are costly in computational time. One possible solution in the case of the reaction diffusion equation is to split the operators as discussed previously.

In this chapter, we solve the full reaction diffusion problem with two approaches. The first scheme is the direct PTS approach which shows good behaviour when both heat and logistic equations are considered. The second scheme uses a combination of PTS[1/1] and other explicit methods on each of the reaction and diffusion parts. This approach is especially powerful when the explicit scheme, PTS[1/1], is used to solve the stiff part of the reaction diffusion equation instead of an implicit scheme. The PTS[1/1] scheme and its singularities are analyzed before discussing the local error control threshold for the full reaction diffusion equation. Then, new splitting schemes are constructed and their stability criteria are discussed. Finally, supporting examples are shown in order to compare our proposed schemes with other competitive

schemes.

4.1 PTS[1/1] Scheme

First, the application of PTS[1/1] to the general reaction diffusion equation is introduced. We then discuss singularities. Finally, the optimum value for LECT will be decided once the full reaction diffusion equation is solved.

Let us first introduce some well-studied examples for the reaction diffusion equation. First, we present Fitzhugh–Nagumo equation [54] with general reaction term

$$u_t = \alpha u_{xx} + R(u - a_1)(a_2 - u)(u - a_3). \tag{4.1.1}$$

A travelling wave type solution exists corresponding to $u(x \to \infty) = a_1$ and $u(x \to -\infty) = a_3$. When $a_1 = -1$, $a_2 = 1$, and $a_3 = 0$ the Fitzhugh–Nagumo equation reduces to the real Newell–Whitehead equation [21]

$$u_t = \alpha u_{xx} + Ru(1 - u^2).$$

If $a_1 = 0$, $a_2 = 1$, and $a_3 = 0$ the Fitzhugh–Nagumo equation becomes the Huxley equation [21]

$$u_t = \alpha u_{xx} + Ru^2(1 - u).$$

The Fisher equation [20]

$$u_t = \alpha u_{xx} + R\,u(1 - u), \tag{4.1.2}$$

is another common example. Both models (4.1.1) and (4.1.2) are implemented in Section 4.3.2, with various initial conditions.

4.1.1 Singularities of the PTS[1/1] Scheme

Consider the general form

$$u_t = \alpha u_{xx} + f(u), \tag{4.1.3}$$

with arbitrary initial and boundary conditions. Solving (4.1.3) by PTS[1/1], will lead to

$$u_{i+1}^j = \frac{u_i^j + \left[\frac{-(1/2)\left(\alpha^2 u_{xxxx}^j + \alpha f_{uu}(u_i^j)(u_x^j)^2 + 2\alpha f_u(u_i^j)u_{xx}^j + f_u(u_i^j)f(u_i^j)\right)u_i^j + \left(\alpha u_{xx}^j + f(u_i^j)\right)^2}{\alpha u_{xx}^j + f(u_i^j)}\right] h}{1 - (1/2)\left[\frac{\alpha^2 u_{xxxx}^j + \alpha f_{uu}(u_i^j)u_x^2 + 2\alpha f_u(u_i^j)u_{xx} + f_u(u_i^j)f(u_i^j)}{\alpha u_{xx}^j + f(u_i^j)}\right] h}, \tag{4.1.4}$$

where u_{xxxx}^j, u_{xx}^j and u_x^j are calculated at t_i for the j^{th} component. Thus we see that a singularity exists for the step size

$$h = \frac{2\left(\alpha u_{xx}^j + f(u_i^j)\right)}{\alpha^2 u_{xxxx}^j + \alpha f_{uu}(u_i^j)(u_x^j)^2 + 2\alpha f_u(u_i^j)u_{xx} + f_u(u_i^j)f(u_i^j)}.$$

Therefore, as discussed in Chapter 2, a LECT is required where alternative schemes are used such as PTS[1/1]$_{\text{Taylor}}$, PTS[1/1]$_{[0/2]}$, and PTS[1/1]$(\frac{h}{s})$. As in Chapter 2, a condition applicable in the neighbourhood of these singularities should be considered. This condition will prevent the loss of accuracy due to numerical round off. We refer to Chapter 2 for the location of singularities as well as for the formulation of local error.

The established approaches, PTS[1/1]$_{\text{Taylor}}$, PTS[1/1]$_{[0/2]}$, and PTS[1/1]$(\frac{h}{s})$, are used to solve the Fitzhugh–Nagumo equation (4.1.1). The choice of thresholds are the same as those found in Chapter 2 and, in particular, $\epsilon^* = \sqrt{2}$. Figure 4.1 shows superior results for the new approaches compared to Taylor. The same behaviour should be expected when comparing PTS to other explicit schemes. In Chapters 2 and 3, we found that PTS[1/1]$_{[0/2]}$ does not behave well with stiff equations. In addition, PTS[1/1]$(\frac{h}{s})$ approaches require high computational time. As a result of this, we will focus on the PTS[1/1]$_{\text{Taylor}}$ approach in the remaining discussion.

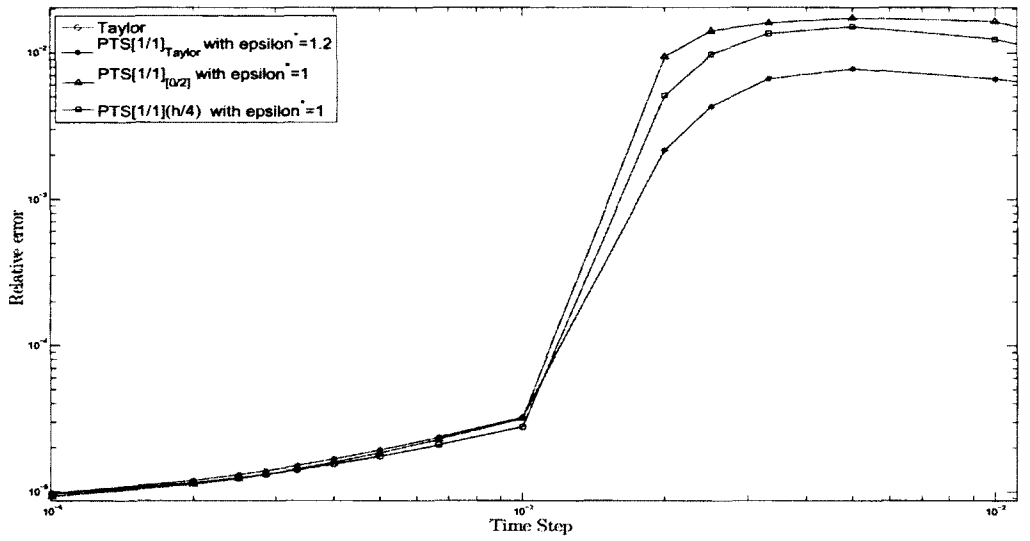

Figure 4.1: Solving the FN equation (4.1.1) by $\text{PTS}[1/1]_{\text{Taylor}}$, $\text{PTS}[1/1]_{[0/2]}$ and $\text{PTS}[1/1](\frac{h}{4})$ with a Gaussian initial condition and $t_0 = 0$, $t_f = 0.1$, $x \in [-5, 5]$, $dx = 0.05$, $\alpha = 1$, $R = 1$, $K = 1$, $a_1 = 0$, $a_2 = 1$, $a_3 = 1$.

4.1.2 The Local Error Control Threshold for the $\text{PTS}[1/1]_{\text{Taylor}}$ Approach

The optimum epsilon was previously discussed for both the heat equation (Chapter 2) and the nonlinear ODEs (Chapter 3). The value of the optimum LECT, ϵ^*, for $\text{PTS}[1/1]_{\text{Taylor}}$ can be tested here for the full reaction diffusion equation using Algorithm (2.5.1). The first condition that needs to be applied to the denominator is

$$\left|1 - \frac{c_2}{c_1} h\right| \leq \epsilon^*.$$

Recall from Algorithm 2.5.1 the use of another condition that helps avoid fixed point behaviour.

A wide range of thresholds $[10^{-16}, 10]$ is used here to search for ϵ^*. Figure 4.2 shows the best choice for ϵ^* when equation (4.1.2) is solved by the $\text{PTS}[1/1]_{\text{Taylor}}$

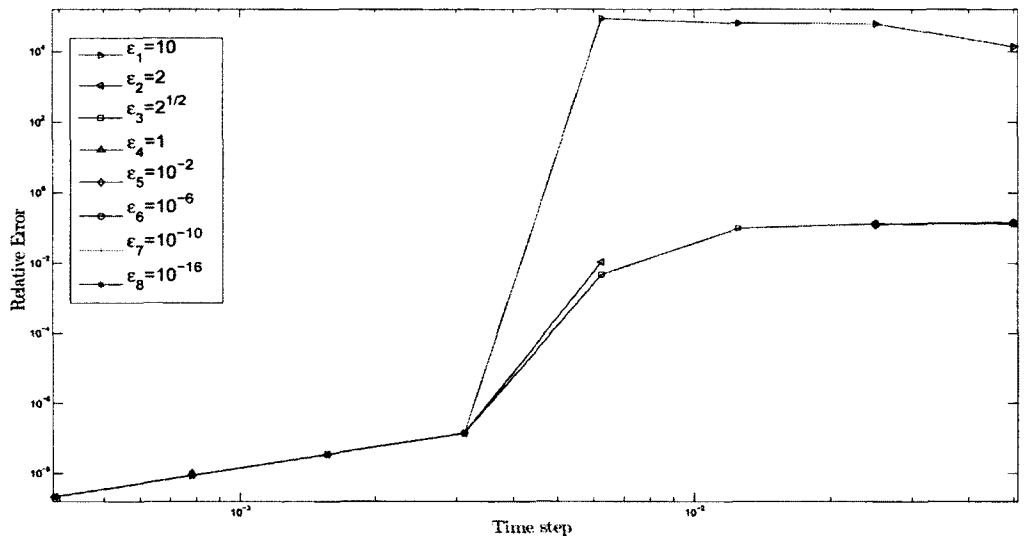

Figure 4.2: PTS$[1/1]_{\text{Taylor}}$ approach with different thresholds when we solve Fisher with the Gaussian initial condition, $t_0 = 0$; $t_f = 1$; $x \in [-20, 20]$; $dx = 0.01$; $\frac{\alpha}{dx^2} = 100$; $R = 1$.

approach. It coincides with the previous selection of LECT, i.e., $\epsilon^* = \sqrt{2}$. This is justifiable since the diffusion part is generally the dominant part.

The tested examples for this scheme and comparison will be presented in Section 4.3 when both Fitzhugh–Nagumo and Fisher equations are solved with various initial conditions.

4.2 Splitting Schemes

Operator splitting is a well-known method for time integration, a technique used to reduce dimension or to solve physical operators separately. As noted in Chapter 1, splitting methods are well suited for reaction diffusion equations. Rewriting our reaction diffusion equation (4.1.3) as two terms, we get

$$u_t = G(u) + F(u), \tag{4.2.1}$$

where $F(u)$ corresponds to the non-stiff part and $G(u)$ corresponds to the stiff operator which is mostly the diffusion operator, i.e $G(u) = \alpha u_{xx}$. Then, it is expected that splitting the two operators using PTS[1/1] and an explicit method will result an efficient scheme especially when one operator is stiff while the the other is not. In order to get a second order accurate scheme in time, a Strang splitting method [73] is used. In many applications of splitting methods, an implicit scheme is used to solve the stiff part, which leads to a huge computational time while our proposed scheme does not require an implicit scheme.

Splitting with PTS[1/1] and RK2 will lead to RK2-PTS scheme and splitting with PTS[1/1] and RKC2 will be called RKC2-PTS scheme. In addition, a variable step size Strang splitting approach is introduced which shows suitability for the reaction diffusion equations.

Lastly, if the reaction term is the stiff part and the diffusion is the non-stiff term, we will solve the reaction with the PTS and the diffusion with either RK2 or RKC2.

4.2.1 Splitting with PTS[1/1] and RK2

For simplicity we make the following notation: A one step of PTS[1/1] scheme, u_{i+1}, that applies to the stiff operator, $G(u)$, is denoted by

$$PTS(u_i, h) = \frac{u_i + \left[\frac{-(1/2)G_t(u_i)u_i + G(u_i)^2}{G(u_i)}\right] h}{1 + \left[\frac{-(1/2)G_t(u_i)}{G(u_i)}\right] h}.$$

A step of RK2 scheme, u_{i+1}, to solve the non-stiff operator, $F(u)$, is denoted by

$$RK(u_i, h) = u_i + hF\left[u_i + \frac{h}{2}F(u_i)\right].$$

RK2-PTS Scheme

Considering equation (4.2.1) with Strang splitting way, we solve half step of $u_t = G(u)$ by PTS[1/1], full step of $u_t = F(u)$ by RK2, and another half step of $u_t = G(u)$ by PTS[1/1]. Combining similar terms leads to the following scheme

$$
\begin{aligned}
u_{\frac{1}{2}} &= PTS\left(u_0, \frac{h}{2}\right), \\
u^*_{i-\frac{1}{2}} &= RK\left(u_{i-\frac{1}{2}}, h\right), \\
u_{i+\frac{1}{2}} &= PTS\left(u^*_{i-\frac{1}{2}}, h\right), \quad i = 1, \ldots, n-1, \\
u^*_{n-\frac{1}{2}} &= RK\left(u_{n-\frac{1}{2}}, h\right), \\
u_n &= PTS\left(u^*_{n-\frac{1}{2}}, \frac{h}{2}\right),
\end{aligned}
\tag{4.2.2}
$$

where n is number of iteration and scheme (4.2.2) will be called RK2-PTS scheme.

RK2-PTS Variable Step (RK2-PTSVS) Scheme

Considering variable step sizes and leaving similar terms untouched can produce a different scheme. The iteration of RK2-PTSVS will be

$$
\begin{aligned}
u^*_i &= PTS\left(u_{i-1}, \frac{h}{2}\right), \\
u^{**}_i &= RK\left(u^*_i, h\right), \\
u_i &= PTS\left(u^{**}_i, \frac{h}{2}\right), \quad i = 1, \ldots, n.
\end{aligned}
\tag{4.2.3}
$$

The difference between RK2-PTS and RK2-PTSVS is that for RK2-PTSVS the $\frac{1}{2}$-PTS steps are not combined, which leads to the variable step size. Since the basis is Strang method for both schemes, the resulted scheme is of second order accuracy.

RK2-PTS versus RK2-PTSVS The direct Strang way, RK2-PTS, has less computational work than the new one, RK2-PTSVS. However the variables step size one is expected to be more accurate, particularly for stiff diffusion term. Figure 4.3 shows better results for the new approach (4.2.3) compared to both the Strang approach (4.2.2) and the pure PTS. Therefore, the new approach (4.2.3) is used instead of (4.2.2) in the remainder of the chapter since in most cases the diffusion is stiff.

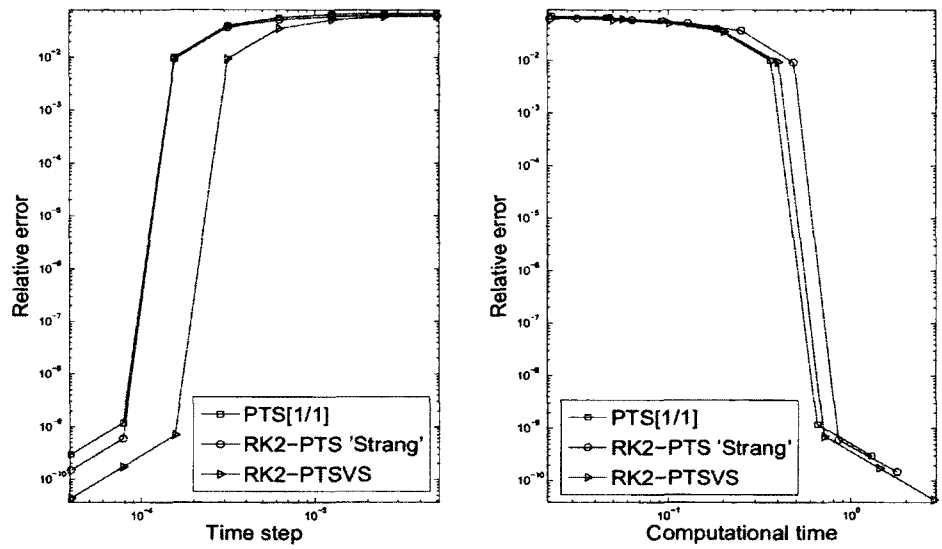

Figure 4.3: Comparison between the Strang approach (4.2.2) and the variable step approach (4.2.3) for solving Fisher with the Gaussian initial condition, $t_0 = 0$, $t_f = 0.1$, $x \in [-20, 20]$, $dx = 0.01$, $\frac{\alpha}{dx^2} = 5000$, $R = 1$.

Stability Analysis of the RK2-PTSVS Scheme

For the purpose of stability analysis, consider the linear scalar test equation

$$u_t(t) = Au(t) + Bu(t), \quad u(0) = u_0, \tag{4.2.4}$$

where A and B are two real numbers. RK2-PTSVS (4.2.3), when applied to (4.2.4), will be

$$\begin{aligned} u_1 &= \left[\left(\frac{1+\frac{Ah}{4}}{1-\frac{Ah}{4}}\right)^2\left(1+Bh+\frac{(Bh)^2}{2}\right)\right]u_0. \\ &\vdots \\ u_n &= \left[\left(\frac{1+\frac{Ah}{4}}{1-\frac{Ah}{4}}\right)^2\left(1+Bh+\frac{(Bh)^2}{2}\right)\right]^n u_0. \end{aligned}$$

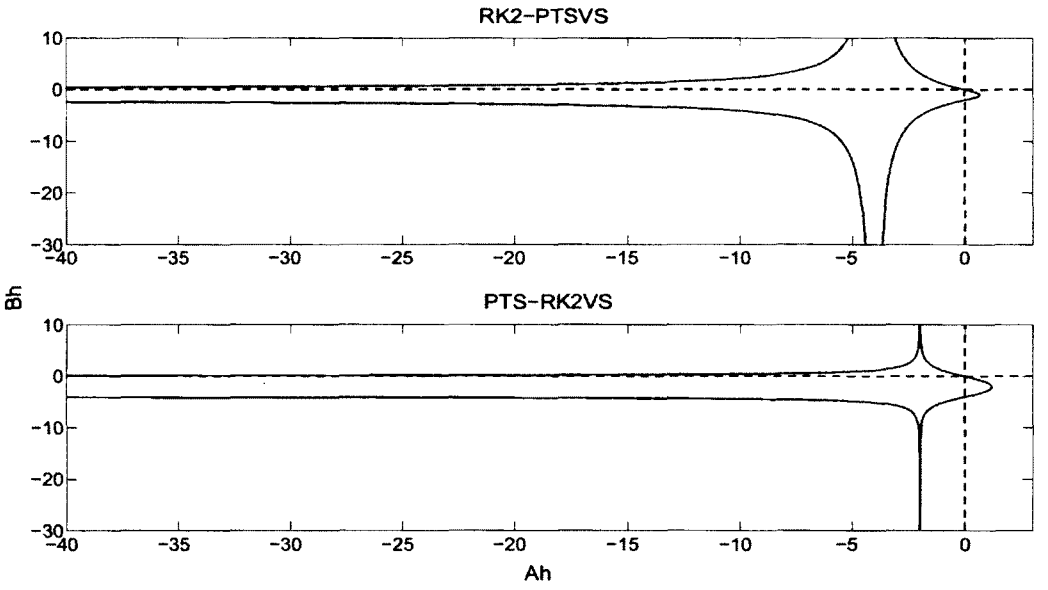

Figure 4.4: Plot of the boundary of the stability region of the RK2-PTSVS and PTS-RK2VS schemes.

Therefore, the growth factor of the RK2-PTSVS scheme is

$$R(Ah, Bh) = \left(\frac{1+\frac{Ah}{4}}{1-\frac{Ah}{4}}\right)^2 \left(1 + Bh + \frac{(Bh)^2}{2}\right), \tag{4.2.5}$$

whereas, the growth factor of the PTS-RK2VS scheme, that is RK applies first, is

$$R(Ah, Bh) = \left(\frac{1+\frac{Ah}{2}}{1-\frac{Ah}{2}}\right) \left(1 + \frac{Bh}{2} + \frac{(Bh)^2}{8}\right)^2. \tag{4.2.6}$$

Figure 4.4 shows a good stability boundary for the RK2-PTSVS and PTS-RK2VS schemes. It is clear from that stability region that it is unbounded from the left. We can simply say that the stability region is the left open region which is bounded by zero from the top and by -2 from the bottom in addition to some extra region. The stability region for the second case is the same except it is bounded from the bottom by -4. These extra parts are found when Ah gets closer to -4 and -2, respectively. Consequently, for $Ah \approx -4$, our scheme RK2-PTSVS constitutes an appropriate solver for a problem with a stiff reaction term.

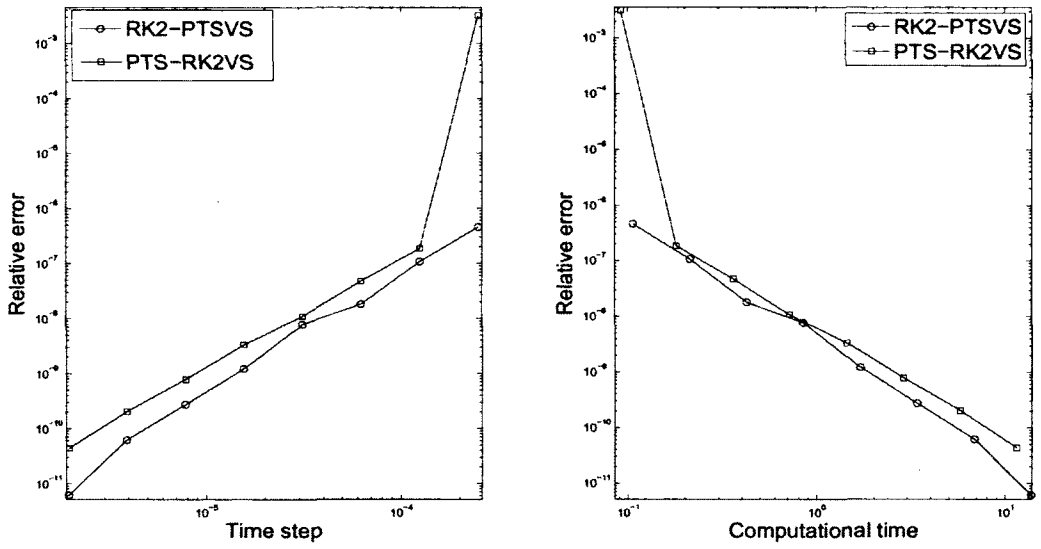

Figure 4.5: Plot of performance of RK2-PTSVS and PTS-RK2VS schemes when solving Fisher with the Gaussian initial condition, $t_0 = 0$; $t_f = 0.1$; $x \in [-20, 20]$; $dx = 0.01$; $\frac{\alpha}{dx^2} = 2400$; $R = 1$.

RK2-PTSVS versus PTS-RK2VS In general, solving the stiff term first will improve the result. This means that, if the stiff term is the heat operator, as is most often the case, we use RK2-PTSVS and vice versa. The improvement, when using RK2-PTSVS, is due to solving the stiff operator with a finer time step. This is shown in Figure 4.5 where a stiff heat operator and non-stiff reaction term are considered. If both operators are mildly stiff, we can say that using the PTS-RK2VS scheme will give a quicker result.

4.2.2 Splitting with PTS[1/1] and RKC2

First, we use the same notation as in the last subsection. A step of a RKC2 scheme, u_{i+1}, to solve a non-stiff operator is denoted by

$$u_{i0} = u_i,$$

$$
\begin{aligned}
u_{i1} &= u_i + \tilde{\mu}_1 h F_{i0}, \\
u_{ij} &= (1-\mu_j-\nu_j)u_i + \mu_j u_{i,j-1} + \nu_j u_{i,j-2} + \tilde{\mu}_j h F_{i,j-1} + \tilde{\gamma}_j h F_{i0}, \\
RKC(u_i,h) &= u_{is}, \quad 2 \le j \le s,
\end{aligned}
$$

where all parameters $\tilde{\mu}_j$, μ_j, ν_j, $\tilde{\gamma}_j$ are defined in Chapter 1 and the function F_{ik} denotes $F(t_i + c_k\tau, u_{ik})$.

RKC2-PTSVS Scheme

Considering equation (4.2.1), we solve a half step of $u_t = G(u)$ by PTS[1/1], a full step of $u_t = F(u)$ by RKC2, and another half step of $u_t = G(u)$ by PTS[1/1]. The following is the formulation of RKC2-PTSVS when variable step size is used

$$
\begin{aligned}
u_i^* &= PTS\left(u_{i-1}, \frac{h}{2}\right), \\
u_i^{**} &= RKC\left(u_i^*, h\right), \\
u_i &= PTS\left(u_i^{**}, \frac{h}{2}\right), \quad i = 1, \ldots, n.
\end{aligned} \tag{4.2.7}
$$

The scheme is of second order. The same comparison between RK2-PTS and RK2-PTSVS can be repeated here for the RKC2-PTS and RKC2-PTSVS schemes.

Stability Analysis of the RKC2-PTSVS Scheme

Consider the linear scalar test equation (4.2.4) and apply RKC2-PTSVS. Similar RK2-PTSVS case, a single iteration of RKC2-PTSVS (4.2.7) with $s = 4$, will be

$$
\begin{aligned}
u_1 &= \left[\left(\frac{1+(\frac{Ah}{4})}{1-\frac{Ah}{4}}\right)^2\left(1+Bh+\frac{(Bh)^2}{2}+\frac{2(Bh)^3}{25}+\frac{(Bh)^4}{250}\right)\right]u_0. \\
&\vdots \\
u_n &= \left[\left(\frac{1+\frac{Ah}{4}}{1-\frac{Ah}{4}}\right)^2\left(1+Bh+\frac{(Bh)^2}{2}+\frac{2(Bh)^3}{25}+\frac{(Bh)^4}{250}\right)\right]^n u_0.
\end{aligned}
$$

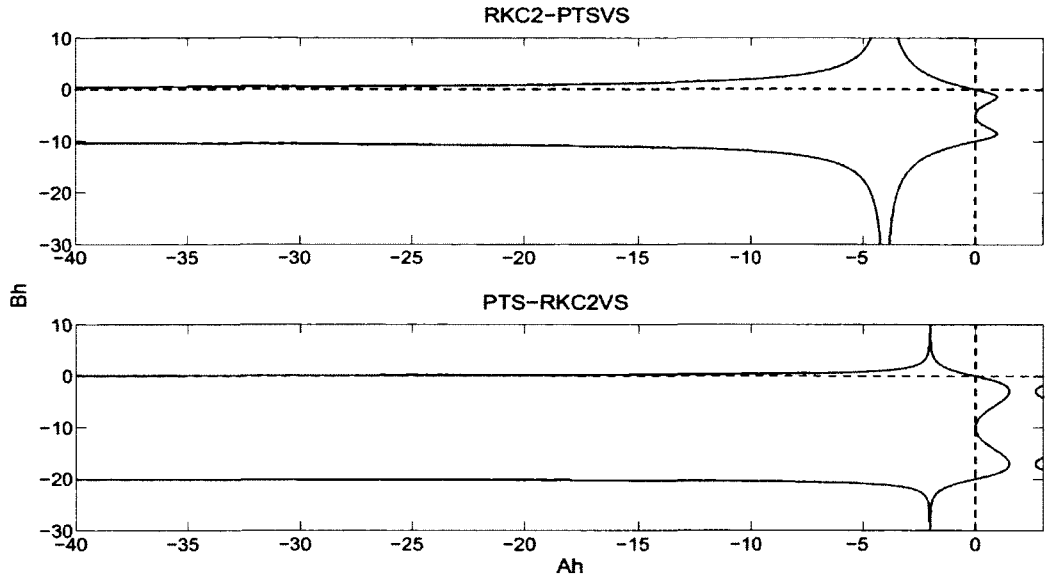

Figure 4.6: The boundary of stability region of the RKC2-PTSVS and PTS-RKC2VS schemes.

Therefore, the growth factor of RKC2-PTSVS scheme will be

$$R(Ah, Bh) = \left(\frac{1+\frac{Ah}{4}}{1-\frac{Ah}{4}}\right)^2 \left(1 + Bh + \frac{(Bh)^2}{2} + \frac{2(Bh)^3}{25} + \frac{(Bh)^4}{250}\right), \tag{4.2.8}$$

while the growth factor of PTS-RKC2VS is

$$R(Ah, Bh) = \left(\frac{1+\frac{Ah}{2}}{1-\frac{Ah}{2}}\right) \left(1 + \frac{Bh}{2} + \frac{(Bh)^2}{8} + \frac{(Bh)^3}{100} + \frac{(Bh)^4}{4600}\right)^2. \tag{4.2.9}$$

Figure 4.6 shows a good stability boundary for the RKC2-PTSVS scheme which is unbounded from the left. The stability region is essentially the open region that is bounded by zero from the top and by -10 from below. The stability region for the reverse case is the same but is bounded by -20 from the bottom. These extra parts exist when Ah approaches -4 and -2, respectively. This means that for $Ah \approx -4$, our scheme RKC2-PTSVS will constitute an appropriate solver for a problem with a stiff reaction term. The stability region of RKC2-PTSVS scheme shows a wider region compared to the stability region of RK2-PTSVS scheme for both cases. This justifies

the use of RKC2-PTSVS scheme to solve the reaction diffusion equation when the reaction term is mildly stiff.

Lastly, based on previous discussions, good performance is expected, when a reaction diffusion equation with stiff diffusion part is solved. The behaviour of both RK2-PTSVS and RKC2-PTSVS schemes are presented in the following section.

4.3 Examples and Final Remarks

In this section, the performance of the pure PTS[1/1], RK2-PTSVS, and RKC2-PTSVS schemes are examined. The Fisher equation and the Fitzhugh–Nagumo equations are the most common examples for the reaction diffusion equation. These two examples have qualitatively covered a broad class of reaction diffusion equations. In addition, the behaviour of the schemes with different initial conditions will be studied. As expected the relative stiffness of the terms will be an important consideration.

In the following subsection, the Fisher equation will be used to study the new splitting schemes. This will be followed by a study of the Fitzhugh–Nagumo equation and a detailed discussion.

4.3.1 Fisher Equation

Consider the Fisher equation

$$u_t = \alpha u_{xx} + R\, u(1-u); \tag{4.3.1}$$

with the Gaussian initial condition

$$u_0 = \sqrt{\frac{1}{1+4t_0}} \exp\left(\frac{-x^2}{1+4t_0}\right), \quad x \in [-20, 20], \tag{4.3.2}$$

with a resolution of 8000 components. We numerically solve this equation with a number of competitive schemes of second order, such as Crank–Nicholson-Adam–

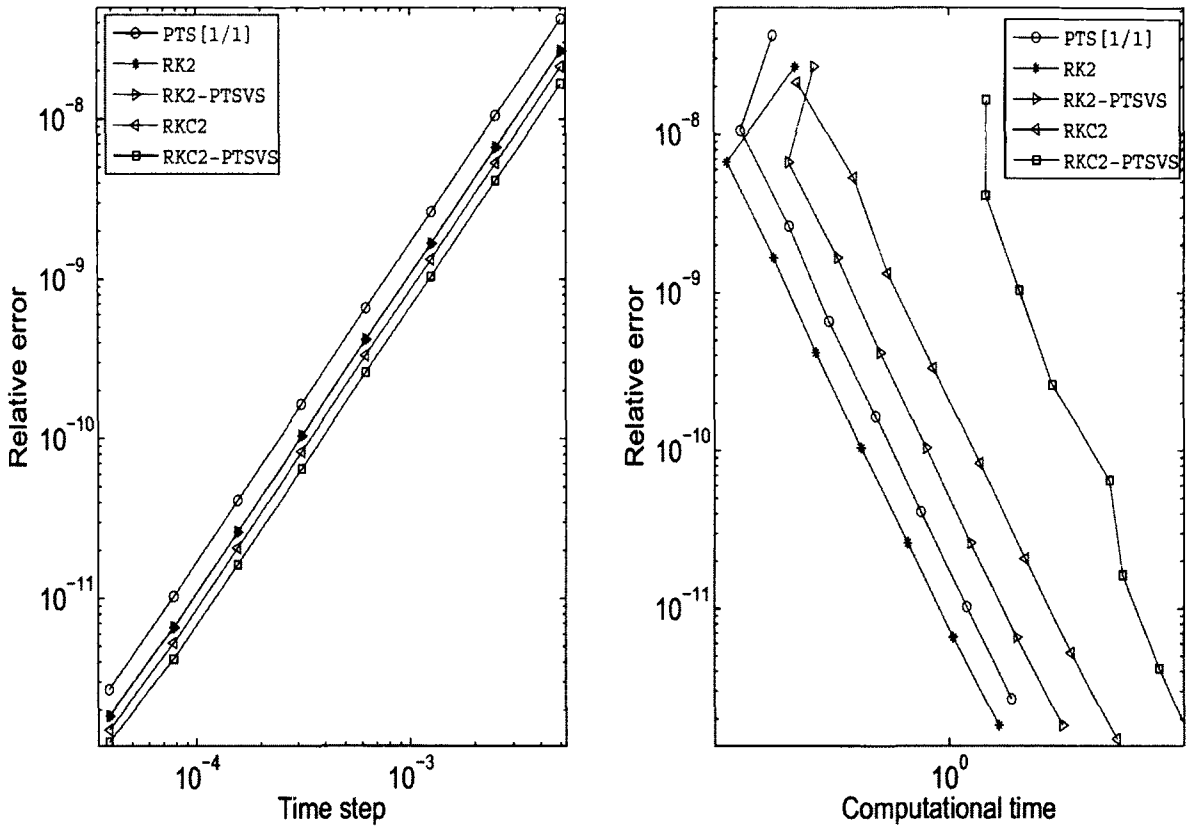

Figure 4.7: Solving Fisher equation using RK2-PTSVS and RKC2-PTSVS with a Gaussian initial condition and $t_0 = 0$, $t_f = 0.1$; $x \in [-20, 20]$, $dx = 0.005$, $\frac{\alpha}{dx^2} = 1$, $R = 1$, non-stiff case.

Bashforth (CNAB), Adam-Bashforth γ (ABg), RK2, and RKC2. In addition, the new schemes, PTS[1/1], RK2-PTSVS, and RKC2-PTSVS will be implemented.

First, assume $\frac{\alpha}{dx^2} = 1$, which corresponds to a non-stiff case. Figure 4.7 indicates good performance for all schemes and in particular the simple explicit RK2 provides the best overall performance. Now, consider $\frac{\alpha}{dx^2} = 5000$, which means $\alpha = 0.125$. Figure 4.8 demonstrates that all schemes are converging with order two, as expected. We see that CNAB2, ABg2, and RKC2 schemes perform better for large time step while RK2, RK2-PTSVS, and RKC2-PTSVS provides better performance for small

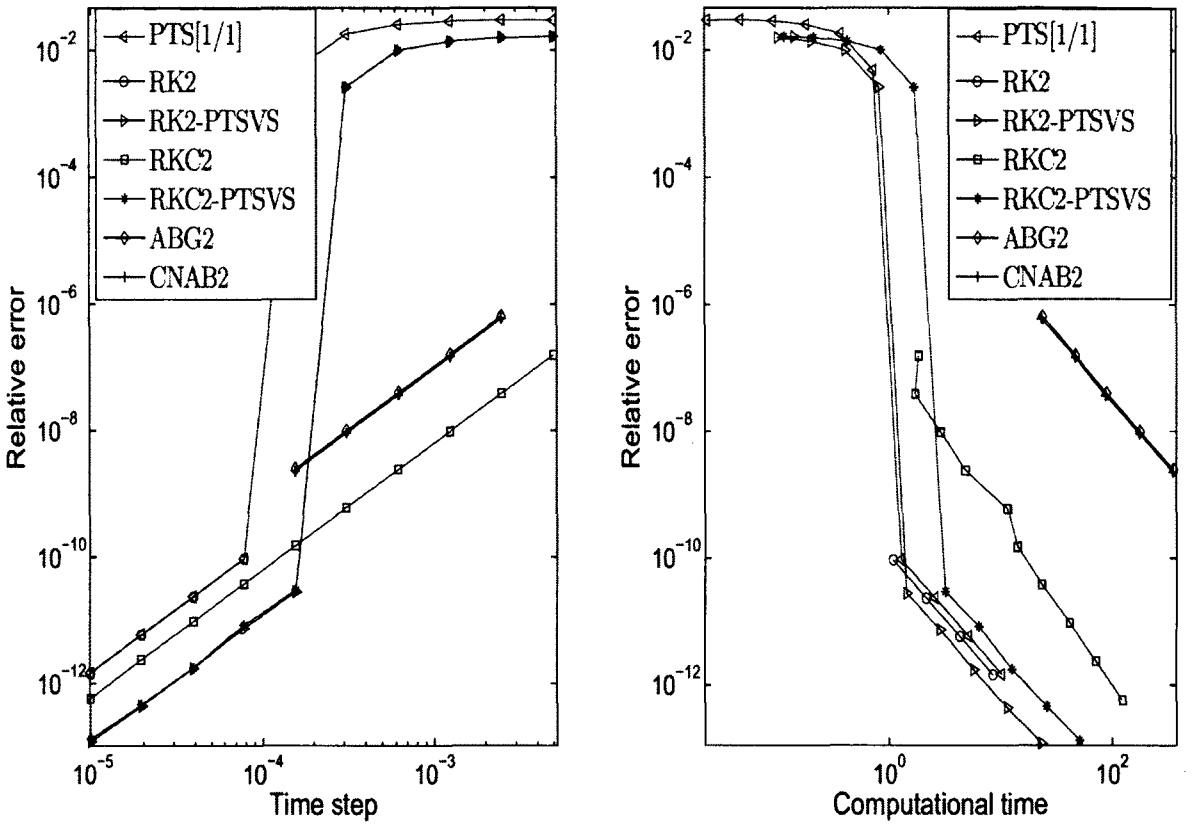

Figure 4.8: Solving Fisher equation using RK2-PTSVS and RKC2-PTSVS with a Gaussian initial condition and $t_0 = 0$, $t_f = 0.1$; $x \in [-20, 20]$, $dx = 0.005$, $\frac{\alpha}{dx^2} = 5000$, $R = 1$, stiff case.

time step and higher accuracy. Overall, in terms of computational expense, the RK2-PTSVS approach appears to provide the best alternative in both regions. This is expected since each operators are solved by a suitable scheme.

4.3.2 Fitzhugh–Nagumo Equation

Now consider the nonlinear Fitzhugh–Nagumo equation

$$u_t = \alpha u_{xx} + R(u - a_1)(a_2 - u)(u - a_3), \tag{4.3.3}$$

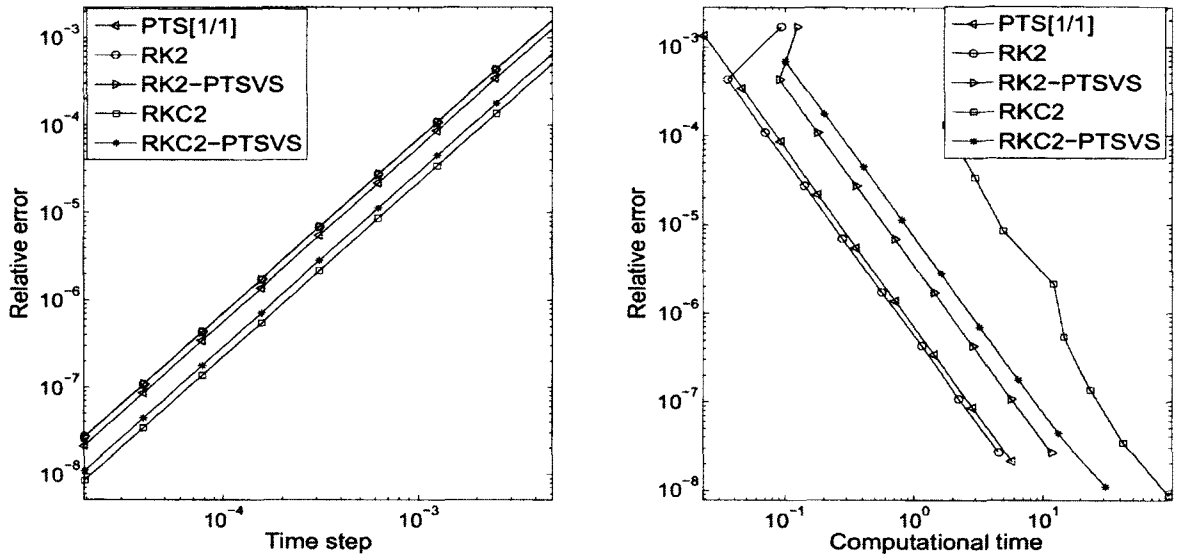

Figure 4.9: Solving the Fitzhugh–Nagumo equation ($\frac{\alpha}{dx^2} = 1$ and $R = 1$) with a Gaussian initial condition.

with $a_1 = 0$, $a_2 = 0.1$, $a_3 = 10$.

We note that this nonlinearity now admits travelling wave solutions, so we will also consider different initial conditions.

Gaussian Initial Condition

First, the Fitzhugh–Nagumo equation (4.3.3) with a Gaussian initial condition (4.3.2) is solved. We start by introducing the case of non-stiff heat and non-stiff reaction operators by setting $\frac{\alpha}{dx^2} = 1$ and $R = 1$. As shown in Figure 4.9, all schemes perform well with slightly less computational time for RK2 and PTS[1/1].

For the case of a stiff diffusion operator $\frac{\alpha}{dx^2} = 5000$ and non-stiff reaction term $R = 1$, as shown in Figure 4.10, both RK2-PTSVS and RKC2-PTSVS provide an acceptable error in shorter time, while, PTS[1/1] and RK2 give a better error in slightly longer time. For higher accuracy, PTS[1/1] and RK2 perform better.

When $\frac{\alpha}{dx^2} = 1$ and $R = 100$, the reaction term is dominant. For RK2-PTSVS

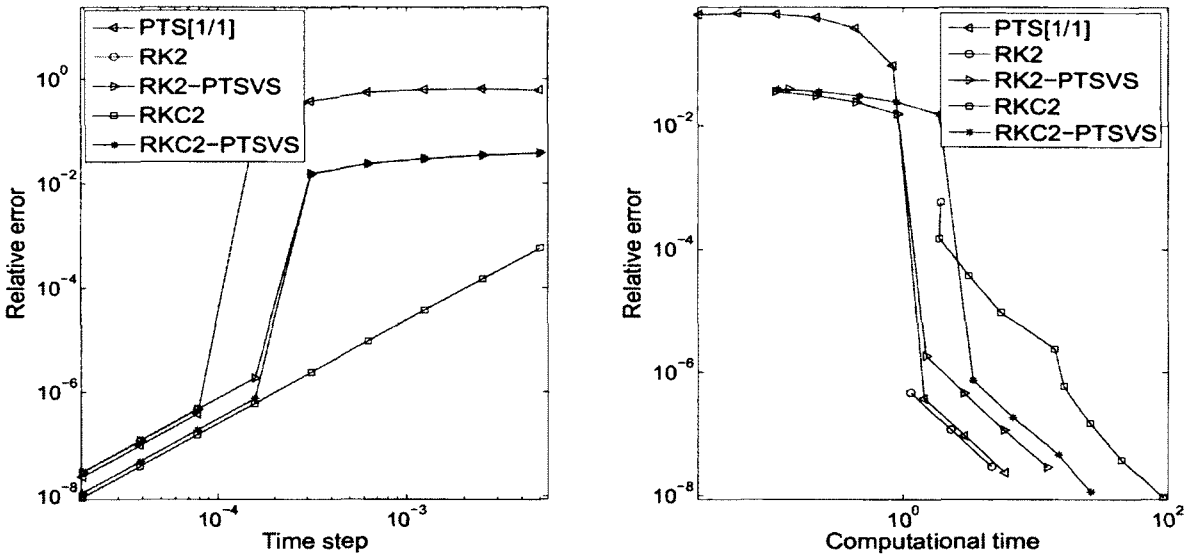

Figure 4.10: Solving the Fitzhugh–Nagumo equation ($\frac{\alpha}{dx^2} = 5000$ and $R = 1$) with a Gaussian initial condition.

and RKC2-PTSVS, as we said in the beginning of this section, we will solve the reaction term by PTS[1/1] and the diffusion by RK2 and RKC2, respectively. The good performance of RK2-PTSVS is seen in Figure 4.11. A small relative error in a fast computational time was obtained.

The last possible case to be considered is $\frac{\alpha}{dx^2} = 5000$ and $R = 100$. Both terms are stiff. Figure 4.12 shows that, as expected, PTS[1/1] on the full equation offers a better performance than the split schemes.

Travelling Wave Initial Condition

The same analysis can be repeated for the Fitzhugh–Nagumo equation (4.3.3) with a travelling wave initial condition

$$u_0 = \frac{1}{1 + \exp\left(\frac{-5}{6}t_0\right) + \frac{\sqrt{6}}{6}x^2}, \tag{4.3.4}$$

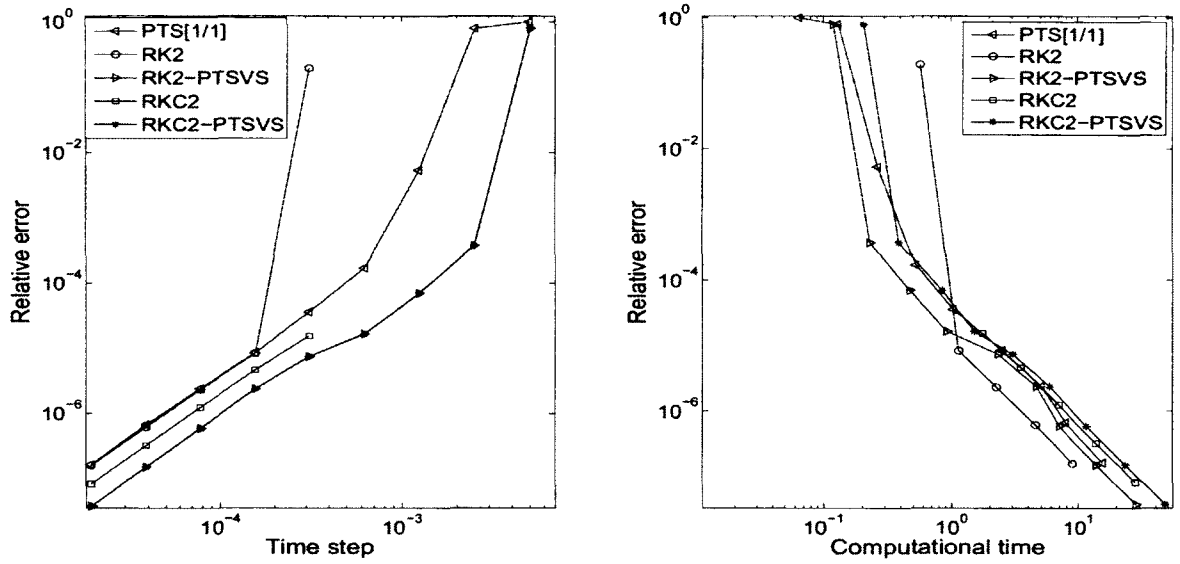

Figure 4.11: Solving the Fitzhugh–Nagumo equation ($\frac{\alpha}{dx^2} = 1$ and $R = 100$) with a Gaussian initial condition.

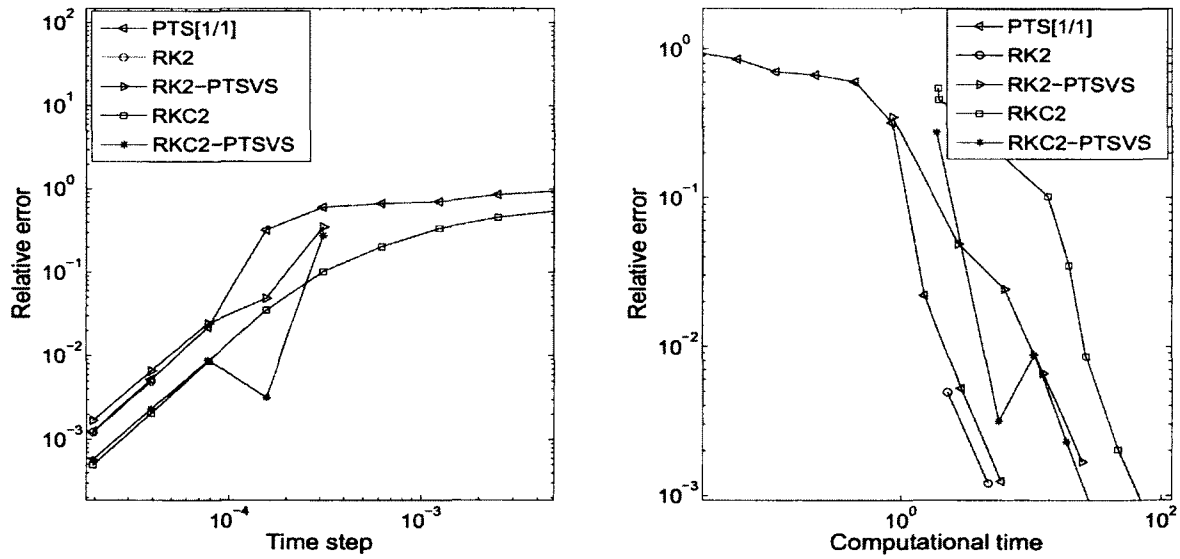

Figure 4.12: Solving the Fitzhugh–Nagumo equation ($\frac{\alpha}{dx^2} = 5000$ and $R = 100$) with a Gaussian initial condition.

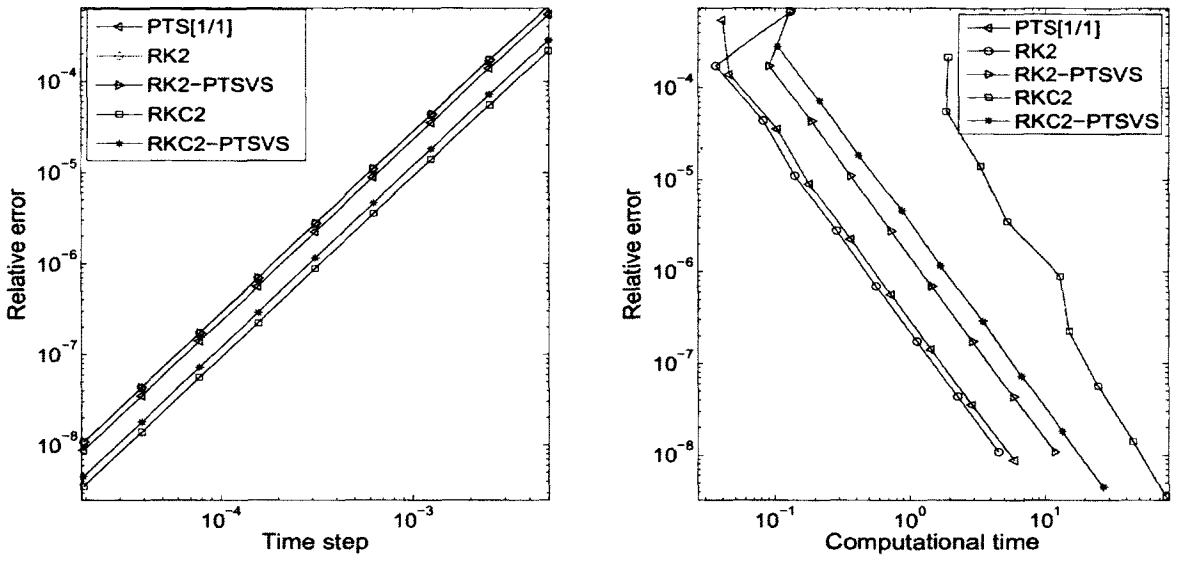

Figure 4.13: Solving the Fitzhugh–Nagumo equation ($\frac{\alpha}{dx^2} = 1$ and $R = 1$) with a travelling wave initial condition.

where $u(x \to -\infty) = a_3$. The four previous cases are generated and presented in Figures 4.13, 4.14, 4.15, and 4.16, respectively.

In the first case, when $\frac{\alpha}{dx^2} = 1$ and $R = 1$, all schemes perform well which is shown in Figure 4.13. However, the computational time does matter. The performance of RK2 and PTS[1/1] are almost the same in this case.

In the second case, $\frac{\alpha}{dx^2} = 5000$ and $R = 1$, RK2-PTSVS and RKC2-PTSVS lead to 10^{-3} relative error in less computational time than the others. If the user requires a smaller error, here RK2 and PTS are the most efficient.

For the case $\frac{\alpha}{dx^2} = 1$ and $R = 100$, both RK2-PTSVS and RKC2-PTSVS perform much better than the PTS[1/1] RK2 and RKC2. The performance of RK2-PTSVS here, shown in Figure 4.15, is similar to the result when a Gaussian I.C. is considered.

The last case, $\frac{\alpha}{dx^2} = 5000$ and $R = 100$, both operators are stiff. Figure 4.16, as expected, shows a very good result when PTS[1/1] is used. This support the use of PTS[1/1] for stiff problem, particularly when the result is required quickly.

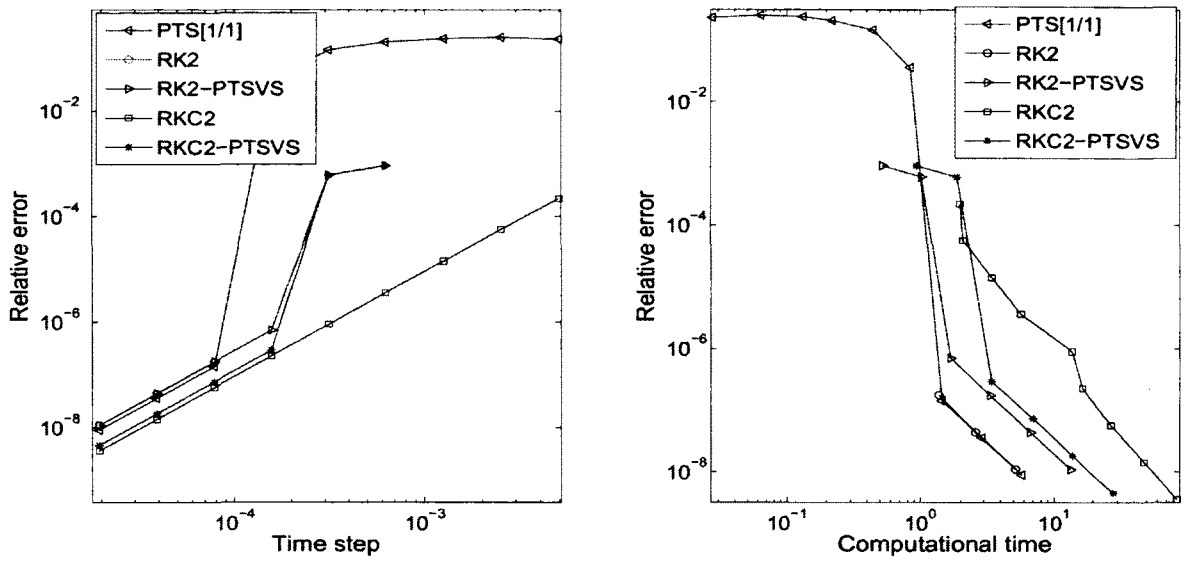

Figure 4.14: Solving the Fitzhugh–Nagumo equation ($\frac{\alpha}{dx^2} = 5000$ and $R = 1$) with a travelling wave initial condition.

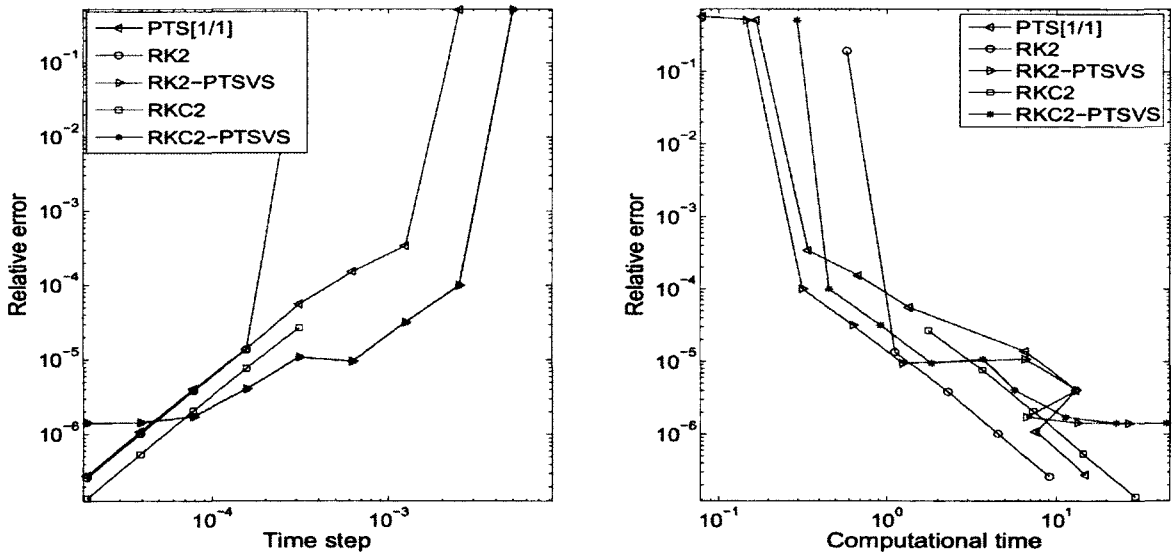

Figure 4.15: Solving the Fitzhugh–Nagumo equation ($\frac{\alpha}{dx^2} = 1$ and $R = 100$) with a travelling wave initial condition.

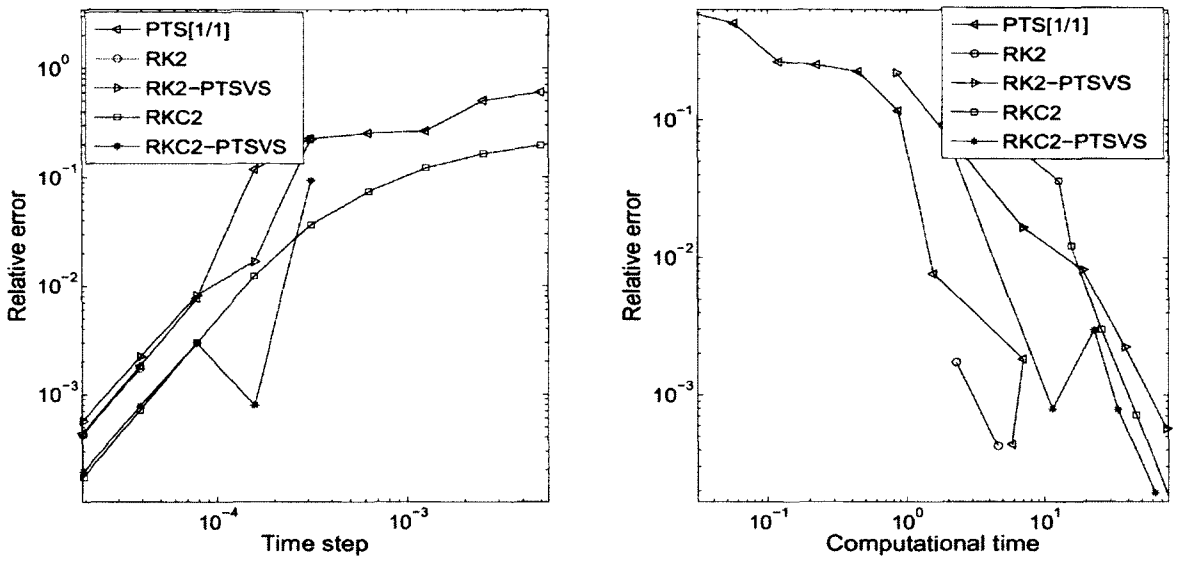

Figure 4.16: Solving the Fitzhugh-Nagumo equation ($\frac{\alpha}{dx^2} = 5000$ and $R = 100$) with a travelling wave initial condition.

Based on these tests, it suggests that the performance of the PTS based schemes does not depend on the initial condition. The critical factor is the stiffness of the term(s).

4.3.3 Final Remarks

In general, based on the above experiments, when a relative error in $[10^{-2}, 10^{-1}]$ is required, the PTS[1/1] scheme is able to compute the solution in a relatively short computational time. For a problem with a stiff diffusion and a stiff reaction term, PTS[1/1] is the best solver.

For the RK2-PTSVS and RKC2-PTSVS schemes, the expected stability region has less restrictive CFL condition than the stability region of the RK2 and RKC2 schemes, respectively. Moreover, the RK2-PTSVS scheme provides good performance when one of the operators is stiff and the other is not.

	Heat Term	Reaction Term	Performance
Case 1	non-stiff	non-stiff	All schemes perform well but RK2 is slightly faster and more accurate.
Case 2	non-stiff	stiff	RK2-PTSVS is faster and more accurate.
Case 3	stiff	non-stiff	For Fisher equation RK2-PTSVS and RKC2-PTSVS schemes are faster and more accurate, while for Fitzhugh-Nagumo equation, PTS[1/1] is faster and more accurate.
Case 4	stiff	stiff	PTS[1/1] scheme is faster and more accurate.

Table 4.1: Performances summary of the PTS[1/1], RK2-PTSVS and RKC2-PTSVS schemes.

A summary of the behaviours and performances of the PTS[1/1], RK2-PTSVS and RKC2-PTSVS schemes for all possible cases is presented in Table 4.1.

4.4 Summary

The reaction diffusion equation was solved by a number of Padé based schemes. The PTS[1/1] scheme has singularities controlled by Taylor. The selected optimum threshold in the previous chapters, ϵ^* close to $\sqrt{2}$, was confirmed to be optimal here as well.

New splitting schemes were constructed. They were mixtures of PTS[1/1], RK2, and RKC2. Strang's way of splitting was the basis for these new schemes, RK2-PTS and RKC2-PTS, which leads to second order methods. Another approach is using

variable step size in the splitting. These approaches are RK2-PTSVS and RKC2-PTSVS. From a stability point of view, the scalar case shows good stability region for the RK2-PTSVS and RKC2-PTSVS schemes. Both schemes have unbounded stability regions.

It was found that the performance of these schemes is independent of the initial condition what matters is if the initial condition leads to singularity or not. Even if the stiff factor plays a big role on the scheme's behaviour, the PTS[1/1] scheme gives an acceptable result in most cases, especially for a problem with highly diffusive reaction diffusion equations. The RK2-PTSVS scheme is recommended for a problem with a non-stiff diffusion operator and a stiff reaction term.

Chapter 5

Conclusion and Future Directions

Numerical methods for reaction diffusion equations of the form

$$u_t = \alpha\, u_{xx} + R\, f(u), \tag{5.0.1}$$

are studied.

Beginning with a literature review of numerical methods in Chapter 1, our primary goal was to find another scheme that would provide better performance. As we know, the stability region of explicit schemes are restricted by CFL conditions while implicit schemes can be computationally costly. These factors motivated the development of an explicit scheme. One which is highly stable for the purposes of the present application and is associated with lower computational costs because it is explicit. Padé time stepping (PTS) provides such a scheme. It is based on a Padé approximation for time stepping each component of an ODE system. We developed the PTS[1/1] scheme which is of order two to solve reaction diffusion equations.

Some conclusions are summarized in the coming section. Directions for future work are also presented in the next coming section.

5.1 Conclusion and Discussion

The PTS[1/1] is an explicit scheme based on rational approximation but it is highly stable. It is noted that since the scheme is in rational form, spurious singularities can arise in rare cases. Due to the singularities of the PTS[1/1], a local error control is proposed.

In Chapter 2, we introduced the PTS[1/1] scheme for solving ODEs and PDEs. The purely diffusive case when $\alpha \neq 0$ and $R = 0$ was considered. It was shown that the PTS[1/1] scheme has some singularities. We controlled these singularities by introducing the local error control threshold (LECT) and by using a different scheme. The value of the LECT for each proposed approach was discussed. We ended up with three approaches, $\text{PTS}[1/1]_{\text{Taylor}}$, $\text{PTS}[1/1]_{[0/2]}$, and $\text{PTS}[1/1]\left(\frac{h}{s}\right)$. The $\text{PTS}[1/1]_{\text{Taylor}}$ approach showed the best performance especially for strongly diffusive equations. A threshold ϵ^* is needed when the denominator is close to zero to prevent local error growth. Another scheme of order two is used instead of PTS[1/1] when some condition is met. Numerical runs were used to empirically determine the optimal LECT ϵ^*. Based on broad numerical experimentation we find that LECT, ϵ^*, is close to $\sqrt{2}$, though due to the complexity of the algorithm we were unable to prove this formally. It is also noted that another condition was required to prevent spurious fixed point iteration.

In Chapter 3, the case when $\alpha = 0$ and $R \neq 0$ was solved by the PTS[1/1]. The performance of $\text{PTS}[1/1]_{\text{Taylor}}$ and $\text{PTS}[1/1]_{[0/2]}$ on a class of nonlinear equations was introduced. A stability character of nonlinear ODE was discussed. We showed that the PTS[1/1] scheme is stable for solving a large class of nonlinear ODEs.

The full problem of reaction diffusion equations, the case when $\alpha \neq 0$ and $R \neq 0$, was solved by PTS[1/1] in Chapter 4. In addition, new splitting schemes which used PTS[1/1] were modified. The use of both Strang and variable step size splitting schemes were studied. The stability analysis of RK2-PTSVS, PTS-RK2VS, RKC2-

PTSVS, and PTS-RKC2VS was discussed. Different examples of reaction diffusion equations were tested. The results also showed a better performance for $\text{PTS}[1/1]_{\text{Taylor}}$ when both coefficients, α and R, are stiff. RK2-PTSVS is preferable for some cases.

In conclusion, a good scheme, $\text{PTS}[1/1]_{\text{Taylor}}$, was established to solve reaction diffusion equations. The PTS[1/1] scheme can be used to solve general ODEs. We found that PTS[1/1] is a good solver for reaction diffusion equations when both operators are stiff. The PTS-RK2VS has better performance for problems with one stiff operator.

5.2 Future Directions

This thesis suggests the possibility of applying PTS approaches in future studies, studies related to higher accuracy order, different controllers, splitting schemes, and a system of reaction diffusion equations. These future directions are briefly described below.

Higher Order of PTS

In this work, we focus on numerical methods with second-order accuracy in time. These methods have been well studied and have produced satisfactory results. PTS[1/1] behaves better than some competitive schemes when reaction diffusion equations with two stiff operators are solved. As we defined in Chapter 2, PTS[1/1] is constructed by Taylor coefficients c_0, c_1, and c_2. Thus, there is dependence on these coefficients. Thus, considering a higher order of PTS will solve the issue when some Taylor coefficients approach zero. For example, if PTS[2/2] scheme is considered, c_0, c_1, c_2, c_3, and c_4 are used. However, some difficulties may arise when going to higher order, such as the complexity of studying and monitoring these coefficients.

Controllers of PTS

As discussed before, the PTS scheme may have singularities that need to be controlled. For this reason, PTS[1/1]$_{\text{Taylor}}$, PTS[0/2], and PTS[1/1] $\left(\frac{h}{s}\right)$ were used. More methods can be developed either from the same PTS class such as PTS[1/2] and PTS[2/1], or any explicit scheme such as RK2 and RKC2. In addition, we may use PTS[1/1] $\left(\frac{h}{s(D)}\right)$ instead of PTS[1/1] $\left(\frac{h}{s}\right)$ where the variable term $s(D)$ depends on how many sub-steps will resolve the singularity issue, i.e., we may have $s = 2$ for the first singularity and $s = 5$ for the second singularity and so on. However, in practice, we found $s = 4$ was sufficient for the application at hand.

Splitting Schemes

By splitting with RK2 and RKC2, we end up with RK2-PTS, RKC2-PTS, RK2-PTSVS, and RKC2-PTSVS. Other types of explicit methods can be considered. In addition, higher-order splitting methods, which have higher computational cost, can be investigated. However, high computational work may lead to a better overall scheme in certain cases.

Systems of Reaction Diffusion Equations

We limited ourselves to the case of a single reaction diffusion equation. We obtained good results for PTS[1/1] for large scale problem and we expect it to demonstrate the same behaviour for two-dimensional or larger cases. On the other hand, the implicit scheme will suffer from inverting a non-diagonal matrix.

In conclusion, given the potential benefits of PTS type schemes on certain classes of equations and in certain regimes, further study and development is warranted.

Bibliography

[1] *IEEE Standard for Binary Floating-Point Arithmetic*, 754-1985, New York, IEEE, 1985.

[2] Abdulle, A., *Fourth order Chebyshev methods with recurrence relation*, SIAM J. Sci. Comput., 2002, Vol. 23, No. 6, pp. 2041–2054.

[3] Abdusalam, H.A., *Analytic and approximate solutions for Nagumo telegraph reaction-diffusion equation*, App. Math. and Comp., 2004, **157**, pp. 515–522.

[4] Ascher, U., Ruuthy, S., Wetton, B., *Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations*, SIAM J. Numer. Anal., 1995, **32**, pp. 797–823.

[5] Amundsen, D., Bruno, O., *Time stepping via one-dimensional Padé approximation*, J. of Sci. Comp., 2007, **30**, pp. 83–115.

[6] Ascher, U., Ruuth, S., Spiteri, R, *Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations*, App. Numer. Math., 1997, **25**, pp. 151–167.

[7] Abdulle, A., Medovikov, A., *Second order Chebyshev methods based on orthogonal polynomials*, Numer. Math., 2001, **90**, pp. 1–18.

[8] Baker, M., *Analytical aspects of a minimax problem* (in Dutch), Technical Note 62, Mathematical Center, 1971.

[9] Baker, G. A., Graves-Morris, P., *Padé Approximants Part 1: Basic Theory, Encyclopedia of Mathematics and Its Applications, Vol. 13*, Addison-Wesley, 1981.

[10] Blennerhassett, P. J., *On the generation of waves by wind*, Philos. Trans. Roy. Soc. London Ser. 1980, **1441**, pp. 451–494.

[11] Beylkin, G., Keiser, J., Vozovoiy, L., *A new class of time discretization schemes for the solution of nonlinear PDEs*, J. of Comp. Phys., 1998, **147**, pp. 362–387.

[12] Boffi, V., Nonnenmacher, T., *Exact solution to the nonlinear Boltzmann equation for the diffusion of test particles in a host medium*, Il Nuovo Cimento, 1985, **85**, pp. 165–181.

[13] Burden, R., Faires, J., *Numerical analysis*, ITP, California, 1997.

[14] Butcher, J., *Numerical analysis of ordinary differential equations*, Wiley, 1985.

[15] Calvo, M., Palencia, C., *Avoiding the order reduction of Runge–Kutta methods for linear initial boundary value problems*, Math. of Comp., 2001, **71**, pp. 1529–1543.

[16] Capasso, V., Maddalena, L., *Convergence to equilibrium states for a reaction-diffusion system modelling the spatial spread of a class of bacterial and viral diseases*, J. Math. Biol., 1981, **13**, pp. 173–184.

[17] Capasso, V., Wilson, R.E., *Analysis of a reaction-diffusion system modelling man environment man epidemics*, 1997, SIAM J. Appl. Math., **57**, pp. 327–346.

[18] Carpenter, M., Gottlieb, D., Abarbanel, S., Don, W., *The theoretical accuracy of RK tmie discretizations for IBVP: a study of the boundary error*, SIAM J. Sci. Comput., 1995, **16**, pp. 1241–1252.

[19] Cercignani, C., *The Boltzmann equation and its applications*, Springer, 1988.

[20] Chen Z., Guo B., *Analytic solutions of the Fisher equation*, Journal Phys. A: Math. Gen., 1991, **24**, pp. 645–650.

[21] Chen Z., Guo B., *Analytic solutions of the Nagumo equation*, IMA J. of App. Math., 1992, **48**, pp. 107–115.

[22] Chou, C.S., Zhang, Y.T., Zhao, R., Nie, Q., *Numerical methods for stiff Reaction-Diffusion System*, Dynamical System Series B, 2007, **7**, pp. 515–525.

[23] Crampin, E. J., Maini, P. K., *Reaction-diffusion models for biological pattern formation*, Methods and Application of Analysis, 2001, **8**, pp. 415–428.

[24] Cox, S., Matthews, P., *Exponential time differencing for Stiff Systems*, J. of Comp. Phys., 2002, **176**, pp. 430–455.

[25] Douglas, J., Gunn, J., *A general formulation of alternating direction methods. Part I: Parabolic and hyperbolic problems*, Numer. Math., 1964, **6**, pp. 428–453.

[26] Du, Q., Zhu, W., *Stability analysis and application of the exponential time differencing schemes*, J. of Comp. Math., 2004, **22**, pp. 200–209.

[27] Du, Q., Zhu, W., *Analysis and applications of the exponential time differencing schemes and their contour integration modifications*, BIT Numer. Math., 2005, **45**, pp. 307–328.

[28] Epstein, I. R., Pojman, J. A., *An introduction to nonlinear chemical dynamics*, Oxford UP, 1998.

[29] Ferragut, L., Asensio, M., Monedero, S., *A numerical method for solving convection-reaction-diffusion multivalued equations in fire spread modelling*, Advances in Engineering Software, 2007, **38**, pp. 366–371.

[30] Feng, Z., Chen, G., Meng, Q., *A reaction-diffusion equation and its travelling wave solutions*, International J. of Non-Linear Mechanics, 2010, **45**, pp. 634–639.

[31] Guillou, A., Lago, B., *Stability regions of one-step and multistep formulas for differential equations; Investigation of formulas with large stability boundaries (French)*, AFCAL, Grenoble, 1960, 43–56.

[32] Ghatak, A., Thyagarajan, K., *Introduction to fibre optics*, Cambridge University Press, UK, 1998.

[33] Habib, S., Molina-Paris, C., Deisboeck, T.S., *Complex dynamics of tumors: Modelling an emerging brain tumor system with coupled reaction-diffusion equations*, Physica A, 2003, **327**, pp. 501–17.

[34] Hairer, E., Nørsett, S.P., Wanner, G., *Solving ordinary differential equations I: nonstiff problems*, Springer, 2000.

[35] Hairer, E., Wanner, G., *Solving ordinary differential equations II: Stiff and differential-algebraic problems*, Springer, 2004.

[36] van der Houwen, P., Sommeijer, B., *On the internal stability of explicit, m-stage Runge–Kutta methods*, Z. Angew. Math. Mech., 1980, **60**, pp. 479–485.

[37] Huang, W., *Traveling waves for a biological reaction-diffusion model*, J. of Dynamics and Diff. Eq., 2004, **16**, pp. 745–765.

[38] Hundsdorfer, W., Portero, L., *A note on iterated splitting schemes*, J. of Comp. and App. Math., 2007, **201**, pp. 146–152.

[39] Hundsdorfer, W., Verwer, J. G., *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer, New York, 2007.

[40] Jarausch, H., *Analyzing stationary and periodic solutions of systems of parabolic partial differential equations by using singular subspaces as reduced basis*, Mathematical and Computer Modelling, 1994, **20**, pp. 69–87.

[41] Knabner, P., Angermann, L., *Numerical methods for elliptic and parabolic partial differential equations*, Springer, New York, 2003.

[42] Kolmogorov, A.N., Petrovskii, I.G., Piskunov, N.S., *A study of the diffusion equation with increase in the amount of substance, and its application to a biological problem in selected works of A.N. Kolmogorov*, Bull. Moscow Univ., Math. Mech., 1937 , **6**, pp. 1–26.

[43] Koto, T., *IMEX Runge–Kutta schemes for reaction-diffusion equations*, J. of Comp. and App. Math., 2008, **215**, pp. 182–195 .

[44] Koza, Z. , *Numerical analysis of reversible $A+B=C$ reaction-diffusion systems*, Eur. Phys. J. B, 2003, **32**, pp. 507–511.

[45] Kassam, A., Trefethen, L., *Fourth-order time stepping for stiff PDEs*, SIAM J. Sci. Comput., 2005, **26**, 1214-1233.

[46] Lambert, J.D., *Numerical methods for ordinary differential equations: The initial value problem*, Wiley, 1991.

[47] LeVeque, R., *Finite difference methods for ordinary and partial differential equations*, SIAM, Philadelphia, 2007.

[48] Lewis, M.A., Schmitz, G., Kareiva, P., Trevors, J.T., *Models to examine containment and spread of genetically engineered microbes*, Mol. Ecol., 1996, **5**, pp. 165–175.

[49] Maginu, K., *Reaction-diffusion equation describing morphogenesis. I. waveform stability of stationary solutions in a one dimensional model*, Mathematical Bioscience, 1975, **27**, pp. 17–98.

[50] Marciniak-Czochraa, A., Kimmel, M., *Reaction-diffusion approach to modelling of the spread of early tumors along linear or tubular structures*, J. of Theoretical Biology, 2007, **244**, pp. 375–387.

[51] Mcinerney, D., Schnell, S., Baker, R.E., Maini, P. K., *A mathematical formulation for the cell-cycle model in somitogenesis: Analysis, parameter constraints and numerical solutions*, Mathematical Medicine and Biology, 2004, **21**, pp. 85–113.

[52] Meier, S.A., Peter, M.A., Bohm, M., *A two-scale modelling approach to reaction-diffusion processes in porous materials*, Computational Materials Science, 2007, **39**, pp. 29–34.

[53] Miura, T., Maini, P., *Speed of pattern appearance in reaction-diffusion models: implications in the pattern formation of limb bud mesenchyme cells*, Bulletin of Mathematical Biology, 2004, **66**, pp. 627–649.

[54] Murray, J.D., *Mathematical biology*, 1989, Springer, New York.

[55] Murray, J.D., *Mathematical biology spatial models and biomedical applications*, 2003, Springer, New York.

[56] Nagumo, J., Arimoto, S., Yoshizawa, S., *Bistable transmission lines*, IEEE Trans., Circuit Theory, 1965, **12**, pp. 400–412.

[57] Nie, Q., Zhang, Y.T., Zhao, R., *Efficient semi-implicit schemes for stiff systems*, J. of Comp. Phys., 2006, **214**, pp. 521–537.

[58] Olmos, D., Shizgal, B., *A pseudospectral method of solution of Fishers equation*, J. of Comp. and App. Math., 2006, **193**, pp. 219–242.

[59] Pao, C., *Nonlinear parabolic and elliptic equations*, Plenum Press, 1992.

[60] Pearson, J., *Complex patterns in a simple system*, Science, 1993, **261**, pp. 189–192.

[61] Pachepsk, E., Lutscher, F., Nisbet, R.M., Lewis, M.A., *Persistence, spread and the drift paradox*, Theor. Popul. Biol., 2005, **67**, pp. 61–73.

[62] Pell, T.M., Davis, T.G., *Diffusion and reaction in polyester melts*, J. Polym. Sci., 1973, **11**, pp. 1671–1682.

[63] Press, W., Vetterling, W., Teukolsky, S., Flannery, B., *Numerical recipes: The art of scientific computing 3rd edition*, Cambridge University Press, 2007.

[64] Portero, L., Jorge, J., Bujanda, B., *Avoiding order reduction of fractional step Runge–Kutta discretizations for linear time dependent coefficient parabolic problems*, Applied Numerical Mathematics, 2004, **48**, pp. 409–424.

[65] Quarteroni, A., Valli, A., *Numerical approximation of partial differential equations*, 1997, Springer, New York.

[66] Shi, C., Roberts, G.W., Kiserow, D.J., *Effect of supercritical carbon dioxide on the diffusion coefficient of phenol in poly bisphenol A carbonate*, J. Polym. Sci. Part B: Polym. Phys., 2003, **41**, pp. 1143–1156.

[67] Ramos, J.I., *A finite volume method for one-dimensional reaction-diffusion problems*, App. Math. and Comp., 2007, **188**, pp. 739–748.

[68] Ruuth, S., *Implicit-explicit methods for reaction-diffusion problems in pattern formation*, J. Math. Biol., 1995, **34**, pp. 148–176.

[69] Sanz-Serna, J., Verwer, J., Hundsdorfer, W, *Convergence and order reduction of Runge–Kutta schemes applied to evolutionary problems in partial differential equations*, Numer. Math., 1986, **50**, pp. 405–418.

[70] Shampine, L., Sommeijer, B., Verwer, J., *IRKC: An IMEX solver for stiff diffusion-reaction PDEs*, J. of Comp. and App. Math., 2006, **196**, pp. 485–497.

[71] Stewartson, K., Stuart, J.T., *A Non-linear instability theory for a wave system in plane poiseuille flow*, J. Fluid Mechanics, 1971, **48**, pp. 529–545.

[72] Stoer, J., Bulirsch, R., *Introduction to numerical analysis*, Springer, New york, 2002.

[73] Strang, G., *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal., 1968, **5**, pp. 506–517.

[74] Strang, G., *Introduction to applied mathematics*, Wellesley-Cambridge Press, 1986.

[75] Strikwerda, J., *Finite difference schemes and partial differential equations*, SIAM, Philadelphia, 2004.

[76] Tadjeran, C., *Stability analysis of the Crank–Nicholson method for variable coefficient diffusion equation*, Commun. Numer. Meth. Engng, 2007, **23**, pp. 29–34.

[77] Tavakoli, R., Davami, P., *2D parallel and stable group explicit finite difference method for solution of diffusion equation*, App. Math. and Comp., 2007, **188**, pp.1184–1192.

[78] Teixeira, J., *Stable schemes for partial differential equations: The one-dimensional reaction-diffusion equation*, Mathematics and Computers in Simulation, 2004, **64**, pp. 507–520.

[79] Terazima, M., *Diffusion coefficients as a monitor of reaction kinetics of biological molecules*, Physical Chemistry Chemical Physic, 2006, **8**, pp. 545–557.

[80] Thomas, J. W., *Numerical partial differential equations: Finite difference methods*, Springer, New York, 1995.

[81] Trefethen, L., *Finite difference and spectral methods for ordinary and partial differential equations*, Cornell University, 1996.

[82] Turing, A.M., *The chemical basis of morphogenesis*, Phil. Trans. Roy. Soc. Lond., 1952, pp. 37–72.

[83] Twizell, E.H., Gumel, A.B., Cao, Q., *A second-order scheme for the brusselator reaction-diffusion system*, J. of Math. Chem., 1999, **26** , pp. 297–316.

[84] Verwer, J.G., Hundsdorfer, W., *Stability and convergence of the Peaceman–Rachford ADI method for initial-boundary value problems*, Mathematics of Computation, 1989, **187**, pp. 81–101.

[85] Verwer, J.G., Sommeijer, B.P., *An Implicit-explicit Runge–Kutta–Chebyshev scheme for diffusion-reaction equations*, SIAM J. Sci. Comp.., 2004, **25**, pp. 1824–1835.

[86] Verwer, J.G., Sommeijer, B.P., Hundsdorfer, W., *Convergence properties of the Runge–Kutta–Chebyshev method*, Numer. Math., 1990, **57**, pp. 157–178.

[87] Verwer, J.G., Sommeijer, B.P., Hundsdorfer, W., *RKC time-stepping for advection diffusion reaction problems*, J. of Comp. Phys., 2004, **201**, pp. 61–79.

[88] Xenophontos, C., Oberbroecklin, L., *A numerical study on the finite element solution of singularly perturbed systems of reaction-diffusion problems*, App. Math. and Comp., 2007, **187**, pp. 1351–1367.

[89] Yamada, H., Nakagaki, T., Baker, R.E., Maini, P.K., *Dispersion relation in oscillatory reaction-diffusion systems with self-consistent flow in true slime mold*, J. Math. Biol., 2007, **54**, pp. 745–760.

[90] Yee, H., Sweby, P., Griffiths, D., *Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations*, J. of Comp. Phys., 1991, **97**, pp. 249–310.

[91] Zou, X., *Delay induced travelling wave fronts in reaction-diffusion equations of KPP-Fisher type*, J. of Comp. and App. Math., 2002, **146**, pp. 309–321.