August 2016

An Exponential Time Differencing Scheme with a Real Distinct Poles Rational Function for Advection-Diffusion Reaction Equations

Emmanuel Owusu Asante-Asamani
University of Wisconsin-Milwaukee

An Exponential Time Differencing Scheme with a Real Distinct Poles Rational Function for Advection-Diffusion-Reactions systems

by

Emmanuel Asante-Asamani

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Mathematics

at

The University of Wisconsin–Milwaukee

August 2016

ABSTRACT

An Exponential Time Differencing Scheme with a Real Distinct Poles Rational Function for Advection-Diffusion-Reactions systems

by

Emmanuel Asante-Asamani

The University of Wisconsin-Milwaukee, 2016
Under the Supervision of Bruce Wade

A second order Exponential Time Differencing (ETD) scheme for advection-diffusion-reaction systems is developed by using a real distinct poles rational function for approximating the underlying matrix exponentials. The scheme is proved to have second order convergence. It is demonstrated to be robust for reaction-diffusion systems with non-smooth initial and boundary conditions, sharp solution gradients, and stiff reaction terms. In order to apply the scheme efficiently to higher dimensional problems, a dimensional splitting technique is also developed. This technique can be applied to all ETD schemes and is found, in the test problems considered, to reduce computational time by up to 80%.

To my mum, who ensured that I received the best education possible on a limited income;

my wife, whose support kept me in the program

and my siblings, whose confidence in my abilty made me dare to begin.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGMENTS

I give glory to God almighty for offering me wisdom, strength and discipline to complete this PhD. I am grateful to my advisor, Dr. Bruce Wade, for consistently believing in my problem-solving ability and offering clear directives for advancing this work. I wish to thank all my committee members Dr. Dexuan Xie, Dr. Peter Hinow, Dr. Tzu-chu Lin and Dr. Lei Wang for taking time to review my dissertation and offering useful suggestions for improvement. Many thanks to Dr. Abdul Khaliq of Middle Tennessee State University, whose work on Parallel LOD methods for time dependent PDE's inspired this effort. Finally, I am very grateful to the Department of Mathematical Sciences, UWM for fully funding me for the entire duration of my graduate studies.

Chapter 1

Introduction

Time-dependent advection-diffusion reaction equations are mathematical models of the form

$$\frac{\partial u_i}{\partial t} + \nabla \cdot (\vec{a_j} u_j) = \nabla (D_j \nabla u_j) + f(u_j) \quad j = 1, 2, 3, \cdots s$$

which describe the time evolution of chemical or biological species in a fluid medium. Here, $\vec{a}_j = (a_{1j}, a_2 j, a_{3j})$ denotes the velocity field for each of s- species concentrations u_j, D_j the corresponding diffusion matrix and $f_j(u)$a reaction term describing the interaction between the various species. These equations are widely used as mathematical models for describing transport of pollutants in the atmosphere [33, 40], ground and surface water [17]; for tracking progression of tumours [2, 6, 13, 29] and studying pattern formation in biological species [22, 31]. In time, chemical reactions can have very fast and slow process thus creating a multiscale phenomenon. In space the diffusion of species, resulting from either brownian motion or concentration gradients, usually occurs at smaller length scales in comparison to the transport from advection. The multiscale nature of these models results in stiff equations which are challenging to solve numerically [15, 36].

A standard numerical approach for solving these brand of problems is to use the method-of-lines to reduce the partial differential equation (PDE) to an ordinary differential equation(ODE). Hairer & Wanner [15] define stiff equations as those for which explicit methods

do not work. Implicit methods such as Runge-Kutta, Rosenbrock and Backward difference schemes are known to effectively integrate stiff equations. Unfortunately, these methods are computationally expensive and may be impractical for certain applications. To help reduce the computational effort involved in integrating stiff problems other linearly implicit methods [1, 20, 34], semi-implicit methods [7], and projection methods [14] have been introduced. The class of implicit-explicit methods, commonly refered to as IMEX schemes, have also gained prominence in integrating stiff problems [17]. These schemes are designed to integrate the stiff part of the problem implicitly and the non-stiff part explicity, thus ofsetting any stability constraints while offering computational speed. While these schemes have been largely sucessful in reducing the computational time, they are vulnerable to instabilities when both the diffusion and reaction terms are stiff, or when mis-matched boundary and initial conditions lead to spurious oscillation.

In order to solve a much broader range of problems, time integration schemes which maintain stability when applied to non-smooth and stiff chemistry problems are of vital importance. More recently Exponential Time Differencing (ETD) schemes have been developed [8, 19, 24], which make use of a single step representation of the evolutionary dynamics followed by an appropriate discretization of the exponentials that arise. A major attraction of these schemes is in the exact treatment of the linear part of the model through the exponential solution operator, and the semi-implicit treatment of the reaction terms. Variants of the proposed ETD scheme have been developed over the years which adopt different approximations to the integral of the non-linear reaction term [11, 16, 19]. B. Kleefeld et al. introduced an ETD Crank -Nicolson Scheme [21] which utilizes a Padé-(1,1) rational approximation for the matrix exponential. The resulting scheme, though highly efficient, is not L-stable and hence does not damp out spurious oscillations generated by non-smooth initial and boundary conditions. A follow-up paper [37] addresses this problem by proposing the use of a Padé-(0,2) rational approximation, which is L-acceptable. However, the partial fraction decomposition of this approximation, used in deriving the scheme, has complex poles

and requires complex arithmetic in all applications, which may slow the evolution process, depending on the matrices.

For multidimensional systems, the discretized diffusion matrices are often largely sparse with wide bands, which tend to slow down direct solvers during evolution. In [4] an ETD-LOD scheme was developed to reduce the storage requirements for solving higher dimensional problems and speed up the evolution through extrapolation of a first order ETD scheme and a simple type of locally one-dimensional (LOD) splitting to achieve second order accuracy.

In this work, we present a new Exponential Time Differencing Scheme, ETD-RDP, which utilizes a non-Padé rational approximation with real and distinct poles for approximating the matrix exponentials. The scheme is L-stable and thus damps out spurious oscillations. The added advantage of having real and distinct poles for the partial fraction decomposition is that we avoid complex arithmetic and can take advantage of parallel implementation to speed up computation. We compare the performance of the scheme to well known time integrations schemes. We also present techniques to speed-up evolution through different types of locally one-dimensional-splitting and without using extrapolation. Instead, we work directly with a second order ETD scheme. In the first approach we break up the PDE into sub-problems and apply a Strang composition of the sub-solution operators to recover the solution. The second method uses an integrating factor substitution to achieve a natural splitting of the PDE along its spatial dimensions. We apply the technique to split the second order ETD-RDP and ETD-CN schemes and examine its performance.

In the remainder of this chapter, some standard time discretization schemes are introduced which will later be used for investigating the performance of ETD-RDP as well as a brief introduction to Pad/'e rational functions. We close with some introductory material on semi-group theory and its application to reaction diffusion equations is provided. The ETD-RDP scheme is developed in chapter two along with an analysis of its absolute stability region. Chapter three provides a description of the proposed dimensional splitting techniques along with algorithms for implementing them for ETD-RDP and ETD-CN for two dimen-

sional problems. We show in chapter 4 that our proposed scheme is second order accurate and provide numerical tests to support its performance on some well known stiff problems in chapter five. Chapter six highlights some recommendations and possibile directions for future work..

1.1 A Survey of Time Discretization Schemes for Stiff Problems

Runge-Kutta Schemes(RKS)

Consider the general non-autonomous formulation of an intial value problem for a system of ordinary differential equations

$$u_t = \mathcal{G}(t, u), \quad t > 0, u(0) = u_0 \tag{1.1}$$

with $\mathcal{G} : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ and $u_0 \in \mathbb{R}^m$. In the case of an advection-diffusion-reaction equation reduced to a system of ODEs through a method-of-lines discretization we have $\mathcal{G}(t, u) = -Au + f(t, u)$ with differential matrix A and nonlinear reaction term f. Discretizing in time by setting $t_n = nk, n = 0, 1, 2, \cdots$ with time step $0 < k < 1$ and integrating from t_n to t_{n+1} we obtain the single step evolution

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} \mathcal{G}(t, u(t))dt.$$

The main idea of a Runge-Kutta scheme is to approximate the integral by an appropriate quadrature rule and estimate the solution as

$$u_{n+1} = u_n + k \sum_{i=1}^{s} b_j \mathcal{G}(t_{ni}, u_{ni}), \tag{1.2}$$

where u_{ni} is an approximation to the solution $u(t_{ni})$ at the quadrature node $t_{ni} = t_n + c_i k, c_i \in [0,1], i = 1, 2, \cdots s$ with s being the stage order. Each intermediate approximation u_{ni} is obtained by performing a similar integration on (1.1) but from t_n to t_{ni}, and afterwards approximating the resulting integral by an appropriate quadrature rule to obtain,

$$u_{ni} = u_n + k \sum_{j=1}^{s} a_{ij} \mathcal{G}(t_{nj}, u_{nj}), \quad i = 1, 2, \cdots s. \tag{1.3}$$

The form of the coefficient matrix $\mathcal{A} = (a_{ij})$ and quadrature coefficient $b = (b_1, b_2, \cdots, b_s)$ defines the type of Runge-Kutta scheme. The Butcher-array

$$\begin{array}{c|c} c & \mathcal{A} \\ \hline & b^T \end{array} = \begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ c_2 & a_{21} & \cdots & a_{2s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

provides a convenient means of representing these methods. A general procedure for implementing Runge-Kutta methods is to solve the system (1.3) for the intermediary solutions and substitute them into the main scheme (1.2). Explicit Runge-Kutta schemes have strictly lower triangular coeffiecient matrix, $\mathcal{A}$ and so require minimal computational effort. A common example is the Forward Euler scheme which is a single stage RKS with $\mathcal{A} = 0, c = 0, b = 1$. For implicit methods, where $\mathcal{A}$ tends to have non-zero diagonal enteries, a Newton-type iteration is required for obtaining intermediate solutions if $\mathcal{G}$ is nonlinear function of the unknown solution u. Of particular importance are diagonally implicit RK schemes (DIRK), for which $\mathcal{A}$ is lower triangular with non-zero diagonal elements. These methods offer significant savings in computational time compared to methods having $a_{ij} \neq 0, j > i$ because each intermediate approximation can be obtained sequentially. A more efficient subclass of DIRK shemes, the singly diagonally implicit (SDIRK) schemes, have identical

diagonal entries. This permits, in the case where $\mathcal{G}$ is linear, the use of a single LU decomposition for recovering all the intermediate solutions. Another class of implicit RK schemes are the Labatto schemes. For these schemes the coefficients c_i are chosen as the zeros of the polynomial

$$\frac{d^{s-2}}{dx^{s-2}}(x^{s-1}(x-1)^{s-1})$$

and the quadrature coefficients b_j chosen to satisfy

$$\sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q} \quad q = 1, \cdots, 2s-2.$$

For our numerical tests we will consider the Labatto IIIA for which the quadrature coefficients a_{ij} are chosen such that

$$\sum_{j=1}^{s} a_{ij} c_j^{q-1} = \frac{c_i^q}{q}, \quad i = 1, \cdots, s \quad q = 1, \cdots, s$$

and the Labatto IIIB scheme for which a_{ij} must satisfy

$$\sum_{j=1}^{s} b_i c_i^{q-1} a_{ij} = \frac{b_j(1-c_j^q)}{q}, \quad j = 1, \cdots, s \quad q = 1, \cdots, s.$$

Second order Labatto IIIA and Labatto IIIB along with other implicit Runge-Kutta schemes implemented in this work are presented in Table A12. For a more indepth discussion of Runge-Kutta methods see Hairer & Wanner[15].

Rosenbrock Schemes

Rosenbrock schemes are Runge-Kutta type methods which require the solution of a linear system of algebraic equations to obtain intermediate approximations instead of the nonlinear systems resulting from most implicit Runge-Kutta schemes. In this way they help to reduce computational time. Consider the system of odes (1.1) with $\mathcal{G}(t,u) = \mathcal{G}(u)$, a nonlinear

function of u. Applying a diagonally implicit RK scheme leads to the discretization

$$u_{n+1} = u_n + \sum_{i=1}^{s} b_i k_i$$
$$k_i = k\mathcal{G}\left(u_n + \sum_{j=1}^{i-1} a_{ij}k_j + a_{ii}k_i\right) \quad i = 1, \cdots s.$$

Now consider the first order approximation of $\mathcal{G}$,

$$\hat{k_i} = k\mathcal{G}(g_i) + k\mathcal{G}^{'}(g_i)a_{ii}\hat{k_i} \quad i = 1, \cdots s$$

with

$$g_i = u_n + \sum_{j=1}^{i-1} a_{ij}\hat{k_j}$$

and $\hat{k}_j$ the approximate value of k_j. Set $J = \mathcal{G}^{'}(g_i) = \mathcal{G}^{'}(u_n)$ so that the jacobian need not be computed at each stage, then the s-stage Rosenbrock scheme is given as

$$\hat{k_i} = k\mathcal{G}\left(u_n + \sum_{j=1}^{i-1} a_{ij}\hat{k_j}\right) + kJa_{ii}\hat{k_i} \quad i = 1, \cdots s$$
$$u_{n+1} = u_n + \sum_{i=1}^{s} b_i\hat{k_i}.$$

Notice that the intermediate approximations $\hat{k}_i$ can now be obtained by solving successive linear algebraic equations with matrix $(I - kJa_{ii})$. In this way the Newton iterative procedure is avoided. To improve the accuracy of the intermediate approximations the linear combination $kJ\sum_{j=1}^{s}\gamma_{ij}\hat{k}_j$ is used instead of $kJa_{ii}\hat{k}_i$ with γ_{ij} chosen to minimize error and

improve stability. Thus the general s-stage Rosenbrock scheme becomes

$$\hat{k}_i = k\mathcal{G}\left(u_n + \sum_{j=1}^{i-1} a_{ij}k_j\right) + kJ\sum_{j=1}^{i}\gamma_{ij}k_j \quad i = 1, \cdots s$$
$$u_{n+1} = u_n + \sum_{i=1}^{s} b_i\hat{k}_i.$$

For our numerical experiments we implement the 2-stage second order L-stable scheme

$$\begin{aligned} u_{n+1} &= u_n + \frac{3}{2}\tilde{k_1} + \frac{1}{2}\tilde{k_2}, \\ \tilde{k_1} &= k\mathcal{G}(u_n) + \gamma kJ\tilde{k_1} \\ \tilde{k_2} &= \tau\mathcal{G}(u_n + \tilde{k_1}) - 2\tilde{k_1} + \gamma k\tilde{k_2} \end{aligned} \tag{1.4}$$

with $\gamma = 1 + \frac{1}{2}\sqrt{2}$ from ([17],pg 154). For an indepth discussion of the stage order and stability requirements of Rosenbrock schemes see [15].

Linear Multistep Schemes

Linear multistep methods make use of multiple approximations from previous time levels to compute the current solution. They have the general form,

$$\sum_{j=0}^{q}\alpha_j w_{n+j} = k\sum_{j=0}^{q}\beta_j\mathcal{G}(t_{n+j}, w_{n+j}) \quad n = 0, 1, 2, \cdots .$$

The scheme is explicit if $\beta_q = 0$ and implicit otherwise. A major advantage of Linear multistep methods over Runge-Kutta is the need to solve just one nonlinear system instead of s for an s-stage Runge-Kutta method. However, the requirement for q starting values can lead to loss of accuracy or stability if not computed at very small time steps. For sufficiently smooth $\mathcal{G}$ the method has order p if

$$\sum_{j=0}^{q}\alpha_j = 0; \sum_{j=0}^{q}\alpha_j j^i = i\sum_{j=0}^{q}\beta_j j^{i-1} \text{ for } i = 1, 2, 3, \cdots p \text{ [17, Pg. 172]}.$$

A very popular multistep method, particularly for stiff problems, is the backward difference formula (BDF) introduced by Curtis & Hirchschfelder [10]. It requires

$$\beta_q = 1, \beta_j = 0 \quad (0 \leq j \leq q-1)$$

and α_j chosen conveniently to attain an optimal order of accuracy. In this work, ETD-RDP's performance is compared to the second order backward difference scheme (BDF2)

$$\frac{3}{2}u_{n+2} - 2u_{n+1} + \frac{1}{2}u_n = k\mathcal{G}(u_{n+1}). \tag{1.5}$$

For a more comprehensive discussion of multistep methods see Hairer & Wanner [15].

IMEX Schemes

IMEX methods have emerged as competitive schemes for time integration of problems comprising very stiff and mildly or non-stiff parts. Advection-diffusion equations fit very well into this category with the advective parts being non-stiff and the diffusion part stiff. Also, some advection-diffusion reaction equations with mild or non-stiff reactions are also suitable. Consider seperating the ODE system (1.1) into the form

$$u_t = \mathcal{G}_0(t, u(t)) + \mathcal{G}_1(t, u(t)) \tag{1.6}$$

where $\mathcal{G}_0(t, u(t))$ represents the non-stiff component and $\mathcal{G}_1(t, u(t))$ the stiff part. The seperation of the stiff and non stiff parts allows the use of implicit schemes, which are known to have excellent stability properties to handle the stiffness while using explicit methods to speed up computations of nonstiff parts. In this way the entire computation is performed in an efficient and stable manner. The implicit method could be a one-step or multistep method[9]. Here, the performance of ETD-RDP is explored against the following 2^{nd} order IMEX schemes:

(a) IMEX-θ

$$u_{n+1} = u_n + k\mathcal{G}_0(t_n, u_n) + (1-\theta)k\mathcal{G}_1(t_n, u_n) + \theta k\mathcal{G}_1(t_{n+1}, u_{n+1}) \quad \theta \geq \frac{1}{2}$$

which combines the explicit Euler method with the A-stable implicit θ-method.

(b) IMEX-CNLF

$$u_{n+1} - u_{n-1} = 2k\mathcal{G}_0(t_n, u_n) + k\mathcal{G}_1(t_{n+1}, u_{n+1}) + k\mathcal{G}_1(t_{n-1}, u_{n-1})$$

developed by combining the explict midpoint (Leap-Frog) with the implicit trapezoidal rule (Crank Nicolson).

(c) IMEX-BDF2

$$\begin{aligned}\frac{3}{2}u_{n+1} - 2u_n + \frac{1}{2}u_{n-1} = {} & 2k\mathcal{G}_0(t_n, u_n) - k\mathcal{G}_0(t_{n-1}, u_{n-1}) + \gamma k\mathcal{G}_1(t_{n+1}, u_{n+1}) \\ & + 2(1-\gamma)k\mathcal{G}_1(t_n, u_n) - (1-\gamma)k\mathcal{G}_1(t_{n-1}, u_{n-1}). \quad \gamma \geq 0\end{aligned}$$

derived from a combination of the explicit and implicit two-step BDF schemes. Here, we take $\gamma = 1$ as recommended by Hundsdorfer & Verwer [17].

(d) IMEX-Adams

$$\begin{aligned}u_{n+1} - u_n = {} & \frac{3}{2}k\mathcal{G}_0(t_n, u_n) - \gamma k\mathcal{G}_1(t_{n+1}, u_{n+1}) \\ & + (\frac{3}{2} - 2\gamma)k\mathcal{G}_1(t_n, u_n) + (\gamma - \frac{1}{2})k\mathcal{G}_1(t_{n-1}, u_{n-1}).\end{aligned}$$

which uses the explict and implicit two-step Adam's methods. In our numerical experiments the scheme with $\gamma = \frac{9}{16}$, known to provide maximum damping [3], is used.

(e) IMEX-TR

$$u^*_{n+1} = u_n + k\mathcal{G}_0(t_n, u_n) + \frac{1}{2}k\gamma k\mathcal{G}_1(t_n, u_n) + \frac{1}{2}k\mathcal{G}_1(t_{n+1}, u^*_{n+1}) \tag{1.7}$$
$$u_{n+1} = u_n + \frac{1}{2}k\mathcal{G}(t_n, u_n) + \frac{1}{2}k\mathcal{G}(t_{n+1}, u^*_{n+1}).$$

which is derived by applying the implicit and explicit trapezoidal rule [17, Pg.391]. A major limitation of this approach is in the application to reaction-diffusion equations with stiff chemistry. Here, one is forced to use an explicit method to handle potentially stiff reaction terms which can weaken the stability of the scheme. Details about the stability of IMEX schemes for advection-diffusion reactions equations is discussed in chapter 2. For a more detailed discussion of IMEX methods see [15].

1.2 Padé Rational Approximation

Padé schemes are rational functions which have high order of accuracy in approximating the exponential, e^z, for complex number z. There were introduced by Padé [28]. Existing ETD-Padé schemes which use rational functions to approximate the matrix exponential use Padé approximations. Though ETD-RDP is developed with non-Padé rational function, the details in this section offer essential information with which to evaluate the performance of the new scheme.

Theorem 1.2.1. [15, Thm 3.12] *The (k,j)-Padé approximation to e^z is given by*

$$R_{kj}(z) = \frac{P_{kj}(z)}{Q_{kj}(z)}$$

where

$$P_{kj}(z) = 1 + \frac{k}{j+k}z + \frac{k(k-1)}{(j+k)(j+k-1)}\frac{z^2}{2!} + \cdots + \frac{k(k-1)\cdots 1}{(j+k)\cdots(j+1)}\frac{z^k}{k!}$$

$$Q_{kj}(z) = 1 - \frac{j}{k+j}z + \frac{j(j-1)}{(k+j)(k+j-1)}\frac{z^2}{2!} - \cdots + \frac{(-1)^j j(j-1)\cdots 1}{(k+j)\cdots(k+1)}\frac{z^j}{j!}$$

with error

$$e^z - R_{kj}(z) = (-1)^j \frac{j!k!}{(j+k)!(j+k+1)!} z^{j+k+1} + \mathcal{O}(z^{j+k+2})$$

Here are some key results on rational function $R_{kj}(z)$ that will be referenced from later parts of the text.

Theorem 1.2.2. [5] *If $k = j$ the $R_{kj}(z)$ is A-acceptable.*

Theorem 1.2.3. [32] *If $j \geq k$ then $R_{kj}(z)$ is A_0-acceptable.*

Theorem 1.2.4. [12] *If $j \geq k+1$ or $j = s+2$, then $R_{kj}(z)$ is L-acceptable.*

In the development of ETD-Padé schemes, the (0,1)-Padé

$$\frac{1}{1-z}$$

(1,1)-Padé

$$\frac{1+\frac{1}{2}}{1-\frac{1}{2}z}$$

and (0,2)-Padé

$$\frac{1}{1-z+\frac{z^2}{2!}}$$

have so far been used to approximate e^z. Table A13 summarizes some other well known Padé approximations.

1.3 Semigroups of Semilinear Evolution Equations

The analysis of error in exponential time differencing schemes relies heavily on estimates which can be obtained by employing principles from semigroup theory. Infact, the scheme itself is derived from an exact evolution of semilinear parabolic problems whose existence and

uniqueness is grounded in semigroup theory. This section therefore provides fundamental results that will guide both the development of the scheme and its error analysis.

Strongly continous semigroups

Definition 1.3.1. Consider the one parameter operators
$\{S(t); t \geq 0\}$ defined on a Banach space B. If the operators satisfy

(a) $S(0) = I$

(b) $S(t_1 + t_2) = S(t_2)S(t_1) = S(t_1)S(t_2)$

(c) $\parallel S(t) \parallel \leq 1$

(d) For any $u \in B$, $S(t)u \in C([0, +\infty), B)$

then $\{S(t) : t \geq 0\}$ is a one-parameter strongly continuous semigroup of contractions or simply a C_0 semigroup in the Banach space B.

Definition 1.3.2. Suppose $\{S(t) : t \geq 0\}$ is a C_0 semigroup defined on a Banach space B. Let D be a subset of B such that for $x \in D$ $S(t)x$ is differentiable at $t = 0$ from the right i.e

$$D = \{x \in B | \lim_{h \to +0} \frac{S(h)x - x}{h} \text{ exists}\}.$$

For $x \in D$, define the operator A as

$$-Ax = \lim_{h \to +0} \frac{S(h)x - x}{h}.$$

Then A, clearly linear, is said to be the *infinitesmal generator of* $S(t)$.

Lemma 1.3.3. [39, Lem 2.1.1] *For any* $x \in D$, $S(t)x \in C^1([0, +\infty), B)$. *Morever for* $t \geq 0$

$$x - S(t)x = \int_0^t AS(\tau)xd\tau = \int_0^t S(\tau)Axd\tau$$

and

$$\frac{d(S(t)x)}{dt} + A(S(t)x) = 0.$$

It follows from this Lemma that $u = S(t)x$ is a classical solution to the initial value problem for the abstract first order evolution equation

$$\begin{aligned} \frac{du}{dt} + Au &= 0 \\ u(0) &= x. \end{aligned} \tag{1.8}$$

The infinitesmal generator A is proved to be closed and densely defined in [39]. A natural question then becomes, for a given linear operator A, what properties would gaurantee that A is the infinitesmal generator for a C_0 semigroup. For that the following Hille-Yoshida theorem is useful.

Theorem 1.3.4. [39, Thm 2.2.1] *Let A be a linear operator defined in a Banach space B,*

$$A : D(A) \subset B \to B.$$

Then the necessary and sufficient conditions for A being an infinitesmal generator of a C_0-semigroup of contractions are

(i) A is a densely defined operator in B.

(ii) for all $\lambda > 0$, $\lambda I + A$ is a bijective mapping and

$$\| (\lambda I + A)^{-1} \| \leq \frac{1}{\lambda}$$

From the proof of this theorem given in [39], we see that the C_0-semigroup generated by A is constructed as the limit of a sequence of C_0-semigroups $(e^{-tA_\lambda} : \lambda > 0)$ each generated by the bounded linear operator $A_\lambda = \frac{1}{\lambda}(I - J_\lambda)$ with $J_\lambda = (I + \lambda A)^{-1}$. This construction provides some motivation for the representation $S(t) := e^{-tA}$.

The next definition and lemma provide a convenient way of identifying the infinitesmal generator of a C_0 semigroup.

Definition 1.3.5. Let A be a linear operator defined in a Banach space B, $A : D(A) \subset B \to B$. If for any $x, y \in D(A)$ and any $\lambda > 0$,

$$\| x - y \| \leq \| x - y + \lambda(Ax - Ay) \|,$$

then A is said to be an accretive operator. Morever, if A is a densly defined accretive operator, and $I + A$ is surjective, i.e $R(I + A) = B$, then A is said to be maximal accretive operator, in short *m-accretive*

Lemma 1.3.6. [39, Lem 2.2.2] *If A is m-accretive, then A is a closed operator and for all $\lambda > 0$, $R(I + \lambda A) = B$, and*

$$\| (I + \lambda A)^{-1} \| .$$

Therefore the necessary and sufficient condition for A being an infinitesmal generator of a C_0 semigroup is that A is m-accretive.

Theorem 1.3.7. [39, Thm 2.2.2] *Suppose that A is m-accretive in a Banach space B, and $u_0 \in D(A)$, then* (1.8) *has a unique classical solution u such that*

$$u \in C([0, +\infty), D(A)) \cap C^1([0, +\infty), B).$$

Moreover, the following estimates hold

$$\| u(t) \| \leq \| u_0 \| \quad \forall t \geq 0$$
$$\| \frac{du}{dt}(t) \| \leq \| Au_0 \| \quad \forall t \geq 0$$

where $D(A)$ is understood as a Banach space equipped with the graph norm.

Consider now the semilinear evolution equation

$$\frac{du}{dt} + Au = F(u) \tag{1.9}$$
$$u(0) = u_0.$$

where A is maximal accretive operator from a dense subset $D(A)$ in a Banach space B into B, and F is a nonlinear operator from B into B. We are interested in the conditions under which the problem admits a local classical or mild solution.

Definition 1.3.8. [39, Defn 2.5.1] Suppose F is a nonlinear operator from a Banach space B into B. F is said to satisfy the *local Lipschitz condition* if for any positive constant $M > 0$, there is a positive constant L_M depending on M such that when $u, v \in B$, $\| u \| \leq M$ and $\| v \| \leq M$

$$\| F(u) - F(v) \| \leq L_M \| u - v \| .$$

Theorem 1.3.9. [39, Thm 2.5.4] *Suppose that A is m-accretive and F is a nonlinear operator from a Banach space B into B satisfying the local Lipschitz condition. Then for any $u_0 \in B$ there is a positive constant $T > 0$ depending on $\| u_0 \|$ such that problem (1.9) in $[0, T]$ admits a unique local mild solution $u \in C([0, T], B)$ such that*

$$u(t) = S(t)u_0 + \int_o^t S(t - \tau)F(u(\tau))d\tau \quad \forall t \in [0, T].$$

Furthermore if $u_0 \in D(A)$, then u is LIpschitz continuous in $t \in [0, T]$. If B is a reflexive banach space, then u is a classical solution.

Anaylic semigroups

Reaction diffusion equations fit well into the framework discussed above since the Laplacian is an infinitesmal generator of a strongly continuous semigroup of contractions. Unfortunately,

more general second order elliptic operators of the form

$$Au := -\sum_{i,j=1}^{n} a_{ij}(x)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{n} b_i(x)\frac{\partial u}{\partial x_i} + c(x)u \tag{1.10}$$

which shows up in advection-diffusion equations do not generate c_0 semigroups of contractions ([39],pg 59). The subsequent theory provides an extention of the previous theory to cover this special case and offers a framework to support the application of ETD Schemes to advection-diffusioni reaction equations.

Definition 1.3.10. Let

$$\triangle = \{z : \phi < argz < \phi_2\}$$

with $\phi_1 < 0 < \phi_2$ and for $z \in \triangle$, let $T(z)$ be a bounded linear operator. The family $T(z), z \in \triangle$ is an analytic semigroup in $\triangle$ if

(i) $z \to T(z)$ is analytic in $\triangle$

(ii) $T(0) = I$ and $\lim_{z\to 0, z\in\triangle} T(z)x = x$ for $x \in B$

(iii) $T(z_1 + z_2) = T(z_1)T(z_2)$ for $z_1, z_2 \in \triangle$

Theorem 1.3.11. [39, Thm 2.6.2] *Let $T(t)$ be a uniformly bounded C_0-semigroup. Let A be the infinitesmal generator of $T(t)$ and assume $0 \in \rho(-A)$. Then the following statements are equivalent:*

(a) T(t) can be extended to an analytic semigroup in a sector

$$\triangle_\delta = \{z : |argz| < \delta\}$$

and $\| T(z) \|$ is uniformly bounded in every closed subsector $\bar{\triangle}_\delta, \delta' < \delta$, of $\triangle_\delta$.

(b) There exists a constant C such that for every $\sigma > 0, \tau \neq 0$,

$$\| R(\sigma + i\tau : -A) \| = \| ((\sigma + i\tau)I + A)^{-1} \| \leq \frac{C}{|\tau|}$$

(c) There exists $0 < \delta < \frac{\pi}{2}$ *and* $M > 0$ *such that*

$$\rho(-A) \supset \Sigma = \{\lambda : |arg\lambda| < \frac{\pi}{2} + \sigma\} \cup \{0\}$$

and

$$\| R(\lambda; -A) \| \leq \frac{M}{|\lambda|}$$

for $\lambda \in \Sigma, \lambda \neq 0$.

(d) T(t) is differentiable for $t > 0$ *and there is a positive constant* C *such that*

$$\| AT(t) \| \leq \frac{C}{t} \quad \forall t > 0.$$

Now let $S(t)$ be a C_0-semigroup satisfying $\| S(t) \| \leq Me^{wt}$, then by the transformation $T(t) = S(t)e^{-wt}$ the results in Theorem 1.3.11 can be generalised to cover the semigroup $S(t)$ (see [39, Remark 2.6.1]).

Application to reaction-diffusion equations

Consider the intial boundary value problem for the semilinear heat equation:

$$\begin{aligned} u_t - \Delta u &= f(u), \\ u|_\Gamma &= 0, \\ u|_{t=0} &= u_0(x) \end{aligned} \tag{1.11}$$

where Ω is assumed to be a bounded domain in $\mathbb{R}^n$ with smooth boundary Γ. For convenience of analysis, this problem can be converted to the abstract semilnear evolution equation:

$$\frac{du}{dt} + Au = F(u), \tag{1.12}$$
$$u(0) = u_0$$

with $A = -\Delta$, $F(u) = f(u)$.

Theorem 1.3.12. [39, Thm 2.7.4] *Suppose that $f \in C^1(\mathbb{R})$, and $f'(u)$ is uniformly bounded. Then for any $u_0 \in L^2(\Omega)$, problem* (1.12) *admits a unique mild solution u such that*

$$u \in C([0, +\infty), L^2).$$

Furthermore, if $u_0 \in H^2 \cap H_0^1$, then problem (1.12) *has a unique classical solution u such that*

$$u \in C^1([0, +\infty), L^2) \cap C([0, +\infty), H^2 \cap H_0^1)$$

Theorem 1.3.13. [39, Thm 2.7.5] *Suppose*

$$f \in C^3(\mathbb{R}), f(0) = 0,$$

and $n \leq 3$. Then for any

$$u_0 \in H^2 \cap H_0^1$$

problem (1.12) *admits a unique maximal classical solution u such that*

$$u \in C^1([0, T_{max}), L^2) \cap C([0, T_{max}), H^2 \cap H_0^1).$$

Furthermore, there is an alternative:

(i) either $T_{max} = +\infty$, i.e, there is a unique global classical solution,

(ii) or $T_{max} < +\infty$, *and*

$$\lim_{t \to T_{max}-0} \| u \|_{H^2} = +\infty.$$

Theorem 1.3.14. [39, Thm 2.7.6] *Suppose that*

$$f \in C^1(\mathbb{R}), f(0) = 0.$$

Then for any

$$u_0 \in C_0(\bar{\Omega}),$$

problem (1.12) *admits a unique maximum mild solution* u *such that*

$$u \in C([0, T_{max}), C_0(\bar{\Omega}).$$

Furthermore, there is an alternative:

(i) either $T_{max} = +\infty$, *i.e, there is a global classical solution,*

(ii) or $T_{max} < +\infty$, *and*

$$\lim_{t \to T_{max}-0} \| u \|_{C_0(\bar{\Omega})} = +\infty.$$

Theorem 1.3.15. [39, Thm 2.7.7] *Suppose*

$$f(u) = -u^3 + u,$$

$n \leq 3$, *and*

$$u_0 \in H^2(\Omega) \cap H_0^1(\Omega).$$

Then $T_{max} = +\infty$, *i.e problem* (1.12) *admits a unique global classical solution* u *such that*

$$u \in C([0, T_{max}), H^2 \cap H_0^1) \cap C([0, +\infty), L^2).$$

Chapter 2

The ETD Real Distinct Poles Scheme (ETD-RDP)

2.1 Background to ETD Schemes

We are interested in deriving a numerical scheme for the initial value problem of the semi-linear evolution equation:

$$u_t + Au = f(u) \tag{2.1}$$
$$u(0) = u_0$$

in a complex Banach space B. We assume that A is a closed, densely defined linear operator, with resolvent set $\rho(A)$ of A such that, with $\delta \in (0, \frac{\pi}{2})$,

$$\rho(A) \supset \Sigma_\delta = \{z \in \mathbb{C}; \frac{\pi}{2} - \delta \leq |arg z| \leq \pi, z \neq 0\} \cup \{0\} \tag{2.2}$$

and that the resolvent $R(z; A) = (zI - A)^{-1}$ satisfies

$$\| R(z; A) \| \leq M(1 + |z|)^{-1}. \tag{2.3}$$

Under these assumptions it follows from Theorem (1.3.11) that A is the infinitesmal generator of a uniformly bounded strongly continuous semigroup $E(t) = e^{-tA}, t \geq 0$, which may be expressed as

$$E(t) = \frac{1}{2\pi i} \int_{\Gamma} e^{-zt} R(z; A) dz \tag{2.4}$$

where $\Gamma = \{z : |argz| = \psi \in (\frac{\pi}{2} - \delta, \frac{\pi}{2})\}$ with Imz increasing along Γ. In addition we assume that F is a sufficiently smooth nonlinear operator from B into B satisfying the local Lipschitz condition.

For initial data, $u_0 \in B$, it follows from Theorem 1.3.9 that the problem (4.1) admits a unique local mild solution satisfying the integral equation

$$u(t) = E(t)u_0 + \int_0^t E(t-s) f(u(s)) ds, \quad \forall t \in [0, T], \tag{2.5}$$

where $T > 0$ depends on $\| u_0 \|$. Over the time interval $[t_n, t_{n+1}]$ the solution satisfies the recurrence relation

$$u(t_{n+1}) = E(k)u(t_n) + \int_{t_n}^{t_{n+1}} E(t_{n+1} - s) f(u(s)) ds, \quad k = \Delta t$$

which after the change of variable $s = t_n + \tau k$, is reduced to the more convenient form

$$u(t_{n+1}) = E(k)u(t_n) + k \int_0^1 E(k - \tau k) f(u(t_n + \tau k)) d\tau. \tag{2.6}$$

This equation is the basis for deriving all ETD schemes. ETD shemes have gained popularity because the linear terms, which are typiclly the source of stiffness, are handled exactly by the exponetial operator. The method of choice depends on the approach used in estimating the integral and the matrix exponentials that emerge once the semigroup is discretized, using say a finte difference method. Following the approach introduced by Cox and Mathews [8], we use a linear approximation of the nonlinear functions to generate our

second order scheme. Set $g(\tau) = f(u(t_n + \tau k))$ within the interval [0,1], then

$$\hat{g}(\tau) = g(0) + [g(1) - g(0)]\tau \approx g(\tau)$$

when substituted into Equation (2.6) with $u_{n+1} \approx u(t_{n+1})$, transforms it to the form

$$\begin{aligned} u_{n+1} = E(k)u_n &+ \left(k \int_0^1 E(k - \tau k)d\tau\right) f(u_n) \\ &+ \left(k \int_0^1 E(k - \tau k)\tau d\tau\right) [f(u_{n+1}) - f(u_n)] \end{aligned} \tag{2.7}$$

which is much more convenient to integrate. We give two brief Lemma's for carrying out the integration.

Lemma 2.1.1. *Under the stated assumptions on A,*

1. $$k \int_0^1 E(k - \tau k)d\tau = A^{-1}(I - e^{-kA})$$

2. $$I := k \int_0^1 E(k - \tau k)\tau d\tau = A^{-1}(I - e^{-kA}) + k^{-1}A^{-2}(I - e^{-kA} + kAe^{-kA})$$

Proof. From Lemma 1.3.7, we have for part 1,

$$\begin{aligned} \frac{d}{d\tau}\left(A^{-1}e^{-(1-\tau)kA}\right) &= kAA^{-1}e^{-(1-\tau)kA} \\ &= ke^{-(1-\tau)kA} \end{aligned}$$

Consequently,

$$\begin{aligned} k\int_0^1 E(k-\tau k)d\tau &= \int_0^1 ke^{-(1-\tau)kA}d\tau \\ &= \int_0^1 \frac{d}{d\tau}\left(A^{-1}e^{-(1-\tau)kA}d\tau\right) \\ &= A^{-1}(I-e^{-kA}). \end{aligned}$$

Now for part 2, make the change of variable $s=(1-\tau)k$, then

$$\begin{aligned} I &= \int_0^k e^{-sA}(1-k^{-1}s)ds \\ &= \int_0^k e^{-sA}ds - k^{-1}\int_0^k e^{-sA}sds \\ &= II - III. \end{aligned}$$

For II we make use of the result,

$$\frac{d}{d\tau}(A^{-1}e^{-sA}) = -A^{-1}Ae^{-sA} = -e^{-sA}$$

to evaluate the integral,

$$\int_0^k e^{-sA}ds = -\int_0^k \frac{d}{d\tau}(A^{-1}e^{-sA})ds = A^{-1}(I-e^{-kA})$$

For III, recognize that

$$\frac{d}{ds}(sA^{-1}e^{-sA}) = A^{-1}e^{-sA} - se^{-sA}.$$

Therefore

$$\begin{aligned} \int e^{-sA}sds &= \int A^{-1}e^{-sA}ds - \int \frac{d}{ds}(sA^{-1}e^{-sA})ds \\ &= A^{-1}(-A^{-1}e^{-sA}) - sA^{-1}e^{-sA} + C. \end{aligned}$$

Now for the definite integral,

$$III = k^{-1}\int_0^k e^{-sA}sds = -k^{-1}A^{-2}(I - e^{-kA}) + A^{-1}e^{-kA}$$
$$= k^{-1}A^{-2}(I - e^{-kA} + kAe^{-kA})$$

□

Substituting the result of this lemma into Equation (2.7) we arrive at the initial semi-discrete form of the scheme

$$u_{n+1} = e^{-kA}u_n + A^{-1}(I - e^{-kA})f(u_n) + \frac{A^{-2}}{k}(kA - I + e^{-kA})[f(u_{n+1}) - f(u_n)] \qquad (2.8)$$

We consider this form as semi-discrete since the spatial A has not yet been discretized. Once this is done, i.e by replacing the operator with an equivalent matrix, the scheme will be fully discrete. The current scheme is however fully implicit and would require Newton-type iterations to recover the approximate solution. For computational efficiency we seek a linearly implicit implementation. By using the constant approximation $f(u(t_n + \tau k)) = f(u(t_n))$ in Equation 2.6 and integrating, we obtain the first order accurate approximation

$$u^*_{n+1} = e^{-kA}u_n + A^{-1}(I - e^{-kA})f(u_n).$$

By setting $f(u_{n+1}) = f(u^*_{n+1})$ in Equation 2.8, we obtain our final semi-discrete scheme

$$u_{n+1} = e^{-kA}u_n + A^{-1}(I - e^{-kA})f(u_n) + \frac{A^{-2}}{k}(kA - I + e^{-kA})[f(u^*_{n+1}) - f(u_n)] \qquad (2.9)$$
$$u^*_{n+1} = e^{-kA}u_n + A^{-1}(I - e^{-kA})f(u_n)$$

2.2 RDP Rational Approximation

One of the major concerns Cox and Mathews had with thier proposed ETDRK schemes was in dealing with the numerical cancellation errors inherent in evaluating coefficients such as $A^{-1}(I - e^{-kA})$, particularly when A has eigen values close to zero. Kassam and Trefethen in [18] introduced a means of curbing this difficulty by using a contour integral to evaluate the coefficient. Thier approach however depends heavily on choosing a contour in the complex plane that completely encloses all the eigen values of A. This presents new challenges each time the problem is changed or the spatial resolution is adjusted. In [21, 25, 26, 37] Padé rational functions were used in approximating the matrix exponentials, leading to the class of ETD Padé schemes. These schemes cleverly avoid numerical cancellation errors through a series of matrix algebraic operations, performed after replacing the exponential matrix with an appropriate Padé scheme. The ETD Crank Nicolson (ETD-CN) and ETD Padé(0,2) (ETD-P02) are among existing ETD Padé schemes. A major concern with these schemes is the lack of damping of the ETD-CN scheme as well as the complex poles that arise out of the ETD-P02 scheme which further slow down the evolution. Also for multidimensional problems, it would be advantageous to empoly parallel techniques to speed up evolution. This would require a good seperation between the poles of the rational approximation, a property which most Padé schemes lack. Therefore, to facilitate easy parallelization while ensuring L-stability, second order accuracy and a relatively small error constant, we seek a non-Padé rational function of the form

$$r(z) = \frac{1 + a_1 z}{(1 - b_1 z)(1 - b_2 z)} \tag{2.10}$$

with $a_1, b_1, b_2 \in \mathbb{R}, b_1 \neq b_2$.

Lemma 2.2.1. [35] *if* $b_1 + b_2 + a_1 = 1$ *and* $b_1 + b_2 - b_1 b_2 = \frac{1}{2}$, *then r(z) is a second order approximation to* e^z *i.e*

$$r(z) - e^z = C_3 z^3 + \mathcal{O}(z^{p+2})$$

with error constant

$$C_3 = \frac{a_1}{2} - \frac{1}{6}. \tag{2.11}$$

Proof. Consider the power series expansion of the exponential

$$e^z = 1 + z + \frac{z^2}{2} + \frac{z^3}{3!} + o(z^4)$$

and the rational function

$$r(z) = 1 + (a_1 + b_1 + b_2)z + (b_2 + b_1(a_1 + b_1 + b_2) - b_1 b_2)z^2 + o(z^3).$$

The coefficients of z and z^2 clearly agree when $b_1 + b_2 + a_1 = 1$ and
$b_1 + b_2 - b_1 b_2 = \frac{1}{2}$. In the expansion of $r(z)$ the coefficient of the z^3 is calculated to be

$$a_3 = \frac{r^{(3)}(0)}{3!} = a_1 b_1 b_2 + (b_1^2 + b_2^2)(a_1 + b_1 + b_2)$$

making the error constant

$$C_3 = a_1 b_1 b_2 + (b_1^2 + b_2^2)(a_1 + b_1 + b_2) - \frac{1}{6}.$$

Substituting our order equations and simplifiying we get

$$C_3 = \frac{1}{3} - \frac{1}{2}(b_1 + b_2).$$

Now setting $b_1 + b_2 = 1 - a_1$ we obtain

$$C_3 = \frac{a_1}{2} - \frac{1}{6}$$

□

Definition 2.2.2 (L-Acceptable)**.** A rational approximation $r(z)$ of e^z is said to be A-acceptable, if $|r(z)| < 1$ whenever $Re(z)$ is negative and L-acceptable if, in addition $|r(z)| \to 0$ as $Re(z) \to -\infty$.

Lemma 2.2.3. *The rational approximation* (2.10) *is L-acceptable if* $b_1 > 0$ *and* $b_2 > 0$.

Proof. follows from Maximum-Modulus principle. □

From the order equations in Lemma (2.2.1) we can deduce that the choice $b_1 b_2 = \frac{1}{2} - a_1$ leads to an L-acceptable scheme. Applying the result from Lemma (2.2.3) we have the following corollary

Corollary 2.2.4. *If* $a_1 < \frac{1}{2}$ *then the rational approximation r(z) is L-acceptable.*

Notice that if $a_1 = \frac{1}{2}$, then from the order equations, $b_1 + b_2 = \frac{1}{2}$, $b_1 b_2 = 0$ which implies either $b_1 = \frac{1}{2}, b_2 = 0$ or vice versa. Such a choice of a_1 and b_1 leads one to the [1/1] Padé rational function

$$\frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}$$

which is A-acceptable (Figure 2.2,Theorem 1.2.2).

We are interested in deriving a rational approximation which is L-acceptable, has as small an error constant as possible and has good separation between the real poles b_1, b_2 to facilitate easy parallelization. Norsett and Wolfbrandt have shown in [27, Theorem 3.1] that rational approximations of the form (2.10) have smallest error constant when $b_1 = b_2 = b$. Applying this to the order equations we find that $b = 1 - \frac{\sqrt{2}}{2}, a = \sqrt{2} - 1$,results in the optimal error constant $0.040\bar{4}$ when subtituted into (2.11).Therefore we would require $\sqrt{2} - 1 < a < \frac{1}{2}$. It is obvious from (2.11) that the error increases as a approaches $\frac{1}{2}$ and decreases as it approaches $\sqrt{2} - 1$. Also, the L-acceptability of the approximation is threatened as we approach $\frac{1}{2}$ whereas values in the nieghborhood of $\sqrt{2} - 1$ threaten the capability to develop parallel algorithms. Khaliq and Voss identified in [35] that the choice of $a = \frac{5}{12}$ leading to $b_1 = \frac{1}{4}$ and $b_2 = \frac{1}{3}$ gives an error constant of $0.041\bar{6}$ which is near optimal. The resulting

rational approximation

$$r(z) = \frac{1 + \frac{5}{12}z}{(1 - \frac{1}{4}z)(1 - \frac{1}{3}z)} \tag{2.12}$$

with partial fraction decomposition

$$= -\frac{8}{1 - \frac{1}{4}z} + \frac{9}{1 - \frac{1}{3}z}$$

is L-acceptable (Figure 2.2) and permits easy parallelization. We call this rational function the Real Distinct Poles (RDP) rational approximation.

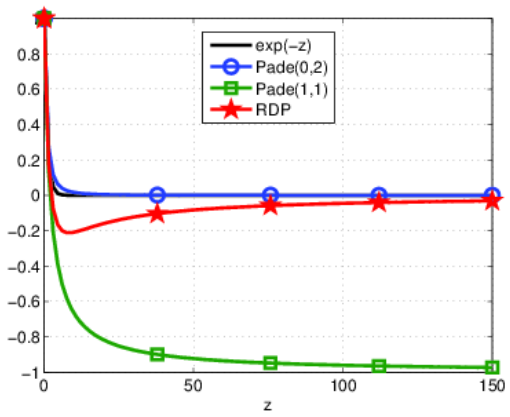


Figure 2.1: L-Acceptability of RDP in comparison with Padé-(0,2), Padé-(1,1).

2.3 Derivation of ETD-RDP Scheme

For the convenience of deriving the numerical scheme we will assume that the operator A has been discretized using an appropriate spatial discretization scheme to obtain a matrix. Since A is invertible by assumption and $r(z)$ is defined on the spectrum of kA we have the representation

$$r(-kA) = \left(I - \frac{5}{12}kA\right)\left[\left(I + \frac{1}{3}kA\right)^{-1}\left(I + \frac{1}{3}kA\right)^{-1}\right]. \tag{2.13}$$

Now, replacing the matrix exponentials in the semi-discrete scheme (2.9) by $r(-kA)$ we obtain the fully discrete scheme

$$v_{n+1} = r(-kA)v_n + A^{-1}(I - r(-kA))f(v_n) + \frac{A^{-2}}{k}(kA - I + r(-kA))[f(v^*_{n+1}) - f(v_n)] \tag{2.14}$$

$$v^*_{n+1} = r_0(-kA)v_n + A^{-1}(I - r_0(-kA))f(v_n) \tag{2.15}$$

where $r_0(z)$ denotes the $[0/1]$ Padé approximation $\frac{1}{1-z}$ and $v_{n+1} \approx u_{n+1}$.

Simplifying the predictor scheme (2.15)

$$\begin{aligned} v^*_{n+1} &= (I + kA)^{-1}v_n + A^{-1}(I - (I + kA)^{-1})f(v_n) \\ &= (I + kA)^{-1}v_n + A^{-1}(I + kA - I)(I + kA)^{-1}f(v_n) \\ &= (I + kA)^{-1}v_n + k(I + kA)^{-1}f(v_n) \\ &= (I + kA)^{-1}(v_n + kf(v_n). \end{aligned}$$

Now for the main scheme, we first provide the following estimates,

$$\begin{aligned} A^{-1}(I - r(-kA) &= A^{-1}\left(I - \left(I - \frac{5}{12}Ak\right)\left(I + \frac{Ak}{4}\right)^{-1}\left(I + \frac{Ak}{3}\right)^{-1}\right) \\ &= A^{-1}\left[\left(I + \frac{Ak}{4}\right)\left(I + \frac{Ak}{3}\right) - I + \frac{5Ak}{12}\right]\left(I + \frac{Ak}{4}\right)^{-1}\left(I + \frac{Ak}{3}\right)^{-1} \\ &= A^{-1}\left(I + \frac{Ak}{4} + \frac{Ak}{3} + \frac{A^2k^2}{12} - I + \frac{5Ak}{12}\right)\left(I + \frac{Ak}{4}\right)^{-1}\left(I + \frac{Ak}{3}\right)^{-1} \\ &= A^{-1}\left(Ak + \frac{A^2k^2}{12}\right)\left(I + \frac{Ak}{4}\right)^{-1}\left(I + \frac{Ak}{3}\right)^{-1} \end{aligned}$$

hence,

$$A^{-1}(I - r(-Ak)) = k\left(I + \frac{Ak}{12}\right)\left(I + \frac{Ak}{4}\right)^{-1}\left(I + \frac{Ak}{3}\right)^{-1}. \tag{2.16}$$

Similarly,

$$\begin{aligned}
\frac{A^{-2}}{k}(r(-kA)-I+kA) &= \frac{A^{-2}}{k}\left[\left(I-\frac{5}{12}Ak\right)\left(I+\frac{1}{4}Ak\right)^{-1}\left(I+\frac{1}{3}Ak\right)^{-1}-I+kA\right] \\
&= \frac{A^{-2}}{k}\left[\left(I-\frac{5}{12}Ak\right)-\left(I+\frac{1}{4}Ak\right)\left(I+\frac{1}{3}Ak\right)+kA\left(I+\frac{1}{4}Ak\right)\left(I+\frac{1}{3}Ak\right)\right] \\
&\left(I+\frac{1}{4}Ak\right)^{-1}\left(I+\frac{1}{3}Ak\right)^{-1} \\
&= \frac{A^{-2}}{k}\left(I-\frac{5}{12}Ak-I-\frac{kA}{4}-\frac{1}{3}Ak-\frac{1}{12}A^2k^2+kA+\frac{1}{4}A^2k^2+\frac{1}{3}A^2k^2+\frac{1}{12}A^3k^3\right) \\
&\left(I+\frac{1}{4}kA\right)^{-1}\left(I+\frac{1}{3}kA\right)^{-1}
\end{aligned}$$

Simplifying we have

$$\frac{A^{-2}}{k}(r(-kA)-I+kA) = \frac{A^{-2}}{k}\left(\frac{1}{2}A^2k^2+\frac{1}{12}A^3k^3\right)\left(I+\frac{1}{4}kA\right)^{-1}\left(I+\frac{1}{3}kA\right)^{-1}$$

leading to,

$$\frac{A^{-2}}{k}(r(-kA)-I+kA) = \frac{k}{2}\left(I+\frac{1}{6}kA\right)\left(I+\frac{1}{4}kA\right)^{-1}\left(I+\frac{1}{3}kA\right)^{-1}. \tag{2.17}$$

Substituting (2.16) and (2.17) into the main fully discrete scheme (2.14), we obtain the final scheme,

$$\begin{aligned}
v_{n+1} &= \left(I-\frac{5}{12}Ak\right)\left(I+\frac{1}{4}Ak\right)^{-1}\left(I+\frac{1}{3}Ak\right)^{-1}v_n \\
&+\frac{k}{2}\left(I+\frac{Ak}{4}\right)^{-1}\left(I+\frac{Ak}{3}\right)^{-1}f(t_n,v_n) \\
&+\frac{k}{2}\left(I+\frac{1}{6}kA\right)\left(I+\frac{1}{4}kA\right)^{-1}\left(I+\frac{1}{3}kA\right)^{-1}f(t_{n+1},v^*) \\
v^*_{n+1} &= (I+kA)^{-1}(v_n+kf(v_n).
\end{aligned}$$

Implementing the scheme as it currently is would involve six matrix solves and two matrix multiplications, which would be highly inefficient. In an attempt to improve the compu-

tational efficiency of the scheme, we make use of the partial fraction decomposition of the rational functions,

$$\left(I-\frac{5}{12}Ak\right)\left(I+\frac{1}{4}Ak\right)^{-1}\left(I+\frac{1}{3}Ak\right)^{-1}=9\left(I+\frac{1}{3}Ak\right)^{-1}-8\left(I+\frac{1}{4}Ak\right)^{-1}$$
$$\left(I+\frac{Ak}{4}\right)^{-1}\left(I+\frac{Ak}{3}\right)^{-1}=4\left(I+\frac{1}{3}Ak\right)-3\left(I+\frac{1}{4}Ak\right)^{-1}$$
$$\left(I+\frac{Ak}{6}\right)\left(I+\frac{Ak}{4}\right)^{-1}\left(I+\frac{Ak}{3}\right)^{-1}=2\left(I+\frac{1}{3}Ak\right)^{-1}-\left(I+\frac{1}{4}Ak\right)^{-1}.$$

After substituting these into the scheme and reorganization of the terms, a final efficient scheme:

$$\begin{aligned} v_{n+1} &= \left(I+\frac{1}{3}Ak\right)^{-1}[9v_n+2kf(t_n,v_n)+kf(t_{n+1},v^*)] \\ &\quad+\left(I+\frac{1}{4}Ak\right)^{-1}[-8v_n-\frac{3k}{2}f(t_n,v_n)-\frac{k}{2}f(t_{n+1},v^*)] \\ v^* &= (I+Ak)^{-1}(v_n+kf(v_n)) \end{aligned} \tag{2.18}$$

which we refer to as the Real Distinct Poles (RDP) Scheme. The scheme can be implemented using the following algorithm:

(1) Solve for first order predictor v^*_{n+1}

$$(I+Ak)v^*_{n+1}=v_n+kF(t_n,v_n)$$

(2) Solve for a_{n+1} (Processor 1)

$$\left(I+\frac{1}{3}Ak\right)a_{n+1}=9v_n+2kF(t_n,v_n)+kF(t_{n+1},v^*_{n+1})$$

(3) Solve for b_{n+1} (Processor 2)

$$\left(I + \frac{1}{4}Ak\right) b_{n+1} = -8v_n - \frac{3}{2}kF(t_n, v_n) - \frac{k}{2}F(t_{n+1}, v^*_{n+1})$$

(4) Obtain approximate solution v_{n+1}

$$v_{n+1} = a_{n+1} + b_{n+1}$$

2.4 Emperical Stability Analysis of ETD-RDP

Consider the scalar test equation

$$u_t + qu = N(u). \tag{2.19}$$

We apply the semidiscrete form of the second order ETD (2.9) to the scalar test equation (2.19) leading to

$$u_{n+1} = e^{-m}u_n + k\mu\left[\frac{(1-e^{-m})}{m} - \frac{(m-1+e^{-m})}{m^2}\right]u_n + k\mu\left(\frac{(m-1+e^{-m})}{m^2}\right)$$
$$\left[e^{-m}u_n + k\mu\frac{(1-e^{-m})}{m}u_n\right]$$

with $qk = m$. Simplifying we obtain

$$u_{n+1} = e^{-m}u_n + k\mu\left[\frac{(1-e^{-m})}{m} + \frac{(m-1+e^{-m})(e^{-m}-1)}{m^2}\right]u_n \tag{2.20}$$
$$+ (k\mu)^2\left[\frac{(m-1+e^{-m})(1-e^{-m})}{m^3}\right]u_n \tag{2.21}$$

As discribed in [1] we generate different stability boundaries on the plane (λ_r, λ_i) using

the substitution,

$$\mu k = \lambda = \lambda_r + i\lambda_i$$

We look for the region in the complex-λ plane where the solution u_n remains bounded as $n \to \infty$.The solution can be sought in the form,

$$u_n = z^n \text{ where } \quad z = |z|e^{i\theta}$$

Cleary the solution grows with n if $|z| > 1$ and it decays if $|z| < 1$. Hence the boundary of the stability region is determined by the condition $|z| = 1$. Thus, to find the boundary we set $z = e^{i\theta}$ with θ ranging from 0 to 2π.

substituting $k\mu = \lambda_r + i\lambda_i$ and $U_n = e^{in\theta}$ we have,

$$e^{i(n+1)\theta} = e^{-m}e^{in\theta} + (\lambda_r + i\lambda_i)be^{in\theta} + (\lambda_r + i\lambda_i)^2 ae^{in\theta}$$

with,

$$b = \frac{(1 - e^{-m})}{m} + \frac{(m - 1 + e^{-m})(e^{-m} - 1)}{m^2}$$
$$a = \frac{(m - 1 + e^{-m})(1 - e^{-m})}{m^3}$$

dividing through by $e^{in\theta}$ we have,

$$e^{i\theta} = e^{-m} + (\lambda_r + i\lambda_i)b + (\lambda_r + i\lambda_i)^2 a$$

leading to,

$$(\lambda_r + i\lambda_i)^2 a + (\lambda_r + i\lambda_i)b = (\cos(\theta) - e^{-m}) + i\sin(\theta) \tag{2.22}$$

expanding out and comparing real and imaginary parts we have,

$$(\lambda_r^2 - \lambda_i^2)a + \lambda_r b = \cos(\theta) - e^{-m} \tag{2.23}$$

$$2\lambda_r \lambda_i a + \lambda_i b = \sin(\theta) \tag{2.24}$$

solving,

$$\lambda_i = \frac{\sin(\theta)}{2\lambda_r a + b}$$

substituting into (2.23) we have

$$a\lambda_r^2 - a\frac{\sin^2(\theta)}{(2\lambda_r a + b)^2} + \lambda_r b = \cos(\theta) - e^{-m}$$

which simplifies to

$$(4\lambda_r^2 a^2 + 4\lambda_r ba + b^2)\lambda_r^2 a - \sin^2(\theta)a + \lambda_r b(4\lambda_r^2 a^2 + 4\lambda_r ba + b^2) = (\cos(\theta) - e^{-m})(4\lambda_r^2 a^2 + 4\lambda_r ba + b^2).$$

Expanding and reorganizing coefficients we obtain the quatic equation

$$A\lambda_r^4 + B\lambda_r^3 + C\lambda_r^2 + D\lambda_r - E = 0 \tag{2.25}$$

where,

$$A = 4a^3$$

$$B = 8a^2 b$$

$$C = 5b^2 a - 4a^2(\cos(\theta) - e^{-m})$$

$$D = b^3 - 4ba(\cos(\theta) - e^{-m})$$

$$E = a\sin^2(\theta) + b^2(\cos(\theta) - e^{-m})$$

Thus our stability region will be defined by the system

$$A\lambda_r^4 + B\lambda_r^3 + C\lambda_r^2 + D\lambda_r - E = 0$$

$$\lambda_i = \frac{\sin(\theta)}{2\lambda_r a + b}$$

To investigate the variation in stability regions for the various second order ETD schemes we make the following substitution

ETDCN

$$e^{-m} \approx \frac{(1 - \frac{1}{2}m)}{(1 + \frac{1}{2}m)} \tag{2.26}$$

ETDPade02

$$e^{-m} \approx \frac{2}{2 + 2m + m^2} \tag{2.27}$$

ETDRDP

$$e^{-m} \approx \frac{9}{1 + \frac{1}{3}m} - \frac{8}{1 + \frac{1}{4}m} \tag{2.28}$$

$$\tag{2.29}$$

Discussion of Stability

The stability regions of the three second order ETD schemes seem very similar except for the observation that ETDCN appears to extend further outward than ETDPADE02 and ETDRDP (Figure 2.2). Comparing ETD schemes to other second and third order schemes we observe that if the eigen values resulting from our spatial discretization or from the

Jacobian of the nonlinear term are real or have imaginary parts close to zero, we can expect ETD to remain stable at about 1.2 times the timestep required for the implicit explicit Adams-Moulton/ Adams Bashford scheme. This better stabiity property of ETD can be attributed it's somewhat semi implicitness.The method can be described as stiffly stable according to the language of [2].

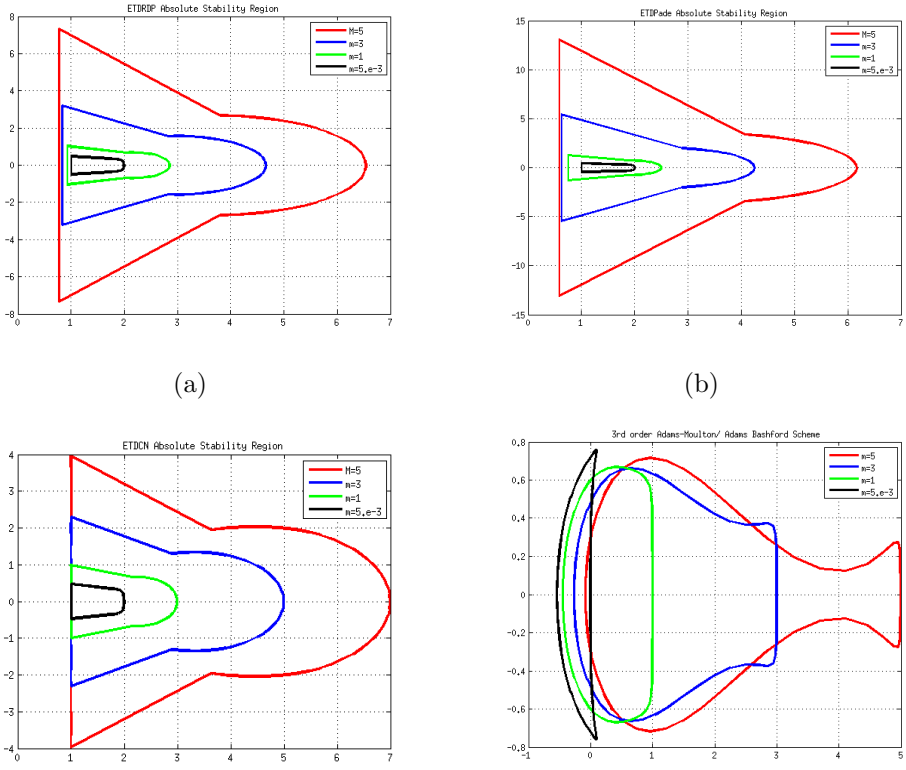


Figure 2.2: stability regions of (a) ETD-RDP (b) ETD-P02 (c) ETD-CN (d) second order implict-explicit Adams-Moulton/Bashford scheme.

Chapter 3

Dimensional Splitting of ETD Schemes

We propose two approaches for solving the multi-dimensional parabolic system:

$$\begin{aligned} u_t + Au &= f(u) \qquad \text{in } \Omega, \qquad t \in (0, T\ ) && (3.1)\\ u|_{\partial\Omega} &= g(x) && (3.2)\\ u(\cdot, 0) &= u_0 \qquad\quad \text{in } \Omega, \end{aligned}$$

by solving a sequence of one-dimensional problems. Here Ω is a bounded domain in $\Re^d$ with Lipschitz continuous boundary and A is matrix obtained by discretizing a diffusion operator and $g(x)$ denotes some prescribed boundary condition.

Lets consider a two-dimensional problem for a system of m reacting species, where the diffusion operator has the natural splitting, $Du_{xx} + Du_{yy}$, and $D \in \Re^{m\times m}$ is a diffusion coefficient matrix. It is well know that the corresponding matrices A_1 and A_2, obtained through a finite difference discretization, can be expressed in Kronecker product notation as

$$A_1 = I_{m_2} \otimes B_{m_1} \otimes D$$
$$A_2 = B_{m_2} \otimes I_{m_1} \otimes D \tag{3.3}$$

on an $m_1 \times m_2$ Cartesian grid where the Kronecker product of two matrices $A_{m\times n}$ and $B_{p\times q}$ is defined to be the $mp \times nq$ matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}.$$

B_n is the usual one-dimensional difference matrix, which has the form

$$B_n = -\frac{1}{h^2}\begin{pmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{pmatrix} \in \Re^{n\times n}, \quad h = \frac{1}{n+1}.$$

in the case of homogeneous Dirichlet boundary conditions, and the corresponding form

$$B_n = -\frac{1}{h^2}\begin{pmatrix} -2 & 2 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 2 & -2 \end{pmatrix} \in \Re^{n\times n}, \quad h = \frac{1}{n+1}.$$

for homogeneous Neumann boundary conditions. I_{m_2} and I_{m_1} are identity matrices. The

diffusion coefficient matrix is of the form

$$D = \begin{pmatrix} d_1 & 0 & & \\ 0 & d_2 & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & d_m \end{pmatrix}.$$

3.1 Integrating Factor Approach

Here we make use of a change of variable to obtain a natural dimensional splitting of ETD schemes. Consider the change of variable

$$v = e^{Bt}u \qquad \& \qquad v_t = e^{Bt}u_t + Be^{Bt}u.$$

Substituting, $v_t = e^{Bt}F(u) - e^{Bt}Au + Be^{Bt}u$. Suppose we choose B such that A and B commute, then

$$v_t + (A - B)e^{Bt}u = e^{Bt}F(u),$$

or,

$$v_t + (A - B)v = e^{Bt}F(e^{-Bt}v).$$

In a two dimensional domain the splitting can be done as $A = A_1 + A_2$, and, with $B = A_1$,

$$v_t + A_2 v = e^{A_1 t}F(e^{-A_1 t}v).$$

Set $e^{A_1 t}F(e^{-A_1 t}v) =: \tilde{F}(v)$, thus

$$\begin{aligned} v_t + A_2 v &= \tilde{F}(v), \quad v(0) = u_0, \\ u &= e^{-A_1 t}v. \end{aligned} \tag{3.4}$$

By Duhamel's principle on (3.4) and with change of variable $t = t_n + \tau$ it follows that

$$v(t_{n+1}) = e^{-kA_2}v(t_n) + \int_0^k e^{-A_2(k-\tau)}\tilde{F}(v(t_n+\tau))d\tau.$$

By a linear approximation of the non-linear function $\tilde{F}(v(t_n+\tau))$ we generate the second order semi-discrete scheme

$$v_{n+1} = e^{-kA_2}v_n + A_2^{-1}(I - e^{-kA_2})\tilde{F}(v,t_n) + \frac{A_2^{-2}}{k}(A_2k - I + e^{-A_2k}) \tag{3.5}$$

$$\times[\tilde{F}(v(t_{n+1}) - \tilde{F}(v(t_n)].$$

From (3.4) $u_n = e^{-A_1nk}v_n$, we can rewrite the semidiscrete equation in terms of the original variable u_n:

$$\begin{aligned} u_{n+1} = e^{-kA_1}e^{-kA_2}u(t_n) + A_2^{-1}(I - e^{-kA_2})e^{-kA_1}F(u_n) \\ + \frac{A_2^{-2}}{k}(A_2k - I + e^{-A_2k})[F(u_{n+1} - e^{-kA_1}F(u_n)]. \end{aligned} \tag{3.6}$$

The matrix exponentials can be approximated using an appropriate rational approximation to obtain a fully discrete scheme. The procedure can be easily extended to 3D to obtain the semidiscrete scheme

$$\begin{aligned} u(t_n) = e^{-kA_3}e^{-kA_2}e^{-kA_1}u(t_n) + A_3^{-1}(I - e^{-kA_3})e^{-kA_1}e^{-kA_2}f(u(t_n)) \\ + \frac{A_3^{-2}}{k}(A_3k - I + e^{-kA_3})[f(u(t_{n+1})) - e^{-kA_2}e^{-kA_1}f(u(t_n))]. \end{aligned}$$

Finally we show that $A_1A_2 = A_2A_1$ by using standard properties of the Kronecker prod-

uct.

$$\begin{aligned} A_1 A_2 &= (I_{m_2} \otimes B_{m_1} \otimes D)(B_{m_2} \otimes I_{m_1} \otimes D) \\ &= I_{m_2} B_{m_2} \otimes B_{m_1} I_{m_1} \otimes D^2 \\ &= B_{m_2} \otimes B_{m_1} \otimes D^2 \end{aligned}$$

similarly

$$\begin{aligned} A_2 A_2 &= (B_{m_2} \otimes I_{m_1} \otimes D)(I_{m_2} \otimes B_{m_1} \otimes D) \\ &= B_{m_2} I_{m_2} \otimes I_{m_1} B_{m_1} \otimes D^2 \\ &= B_{m_2} \otimes B_{m_1} \otimes D^2. \end{aligned}$$

If the matrices do not commute, then errors from the splitting enter into the numerical scheme and increase the overall global error. We provide details of an alternative dimensional splitting for such cases.

3.2 Strang-Type splitting

In the first approach we split (3.1) by distributing the non-linear function evenly in each coordinate direction as

$$u_t = (-A_1 u + \frac{1}{2} F(t,u)) + (-A_2 u + \frac{1}{2} F(t,u))$$

and reformulate the initial value problem into the subproblems

$$w_t + A_1 w = \frac{1}{2} F(t, w) \qquad t_n < t \leq t_{n+1} \;\; w(t_n) = w_n \tag{3.7}$$

$$v_t + A_2 v = \frac{1}{2} F(t, v) \qquad t_n < t \leq t_{n+1} \;\; v(t_n) = w(t_{n+1}) \tag{3.8}$$

with $w_0 = u_0$. Applying Duhamel's principle to each of these equations results in sub-problems and corresponding solution operators $S^1_{\Delta t}, S^2_{\Delta t}$:

$$S^1_{\Delta t} w(t_n) = w(t_{n+1}) = e^{-kA_1} w(t_n) + \frac{1}{2} \int_0^k e^{-(k-\tau)A_1} F(t_n + \tau, w(t_n + \tau)) d\tau$$

$$S^2_{\Delta t} v(t_n) = v(t_{n+1}) = e^{-kA_2} v(t_n) + \frac{1}{2} \int_0^k e^{-(k-\tau)A_2} F(t_n + \tau, v(t_n + \tau)).$$

The one-step solution to (3.1) can then be estimated by

$$u(t_{n+1}) \approx [S^2_{\Delta t} \circ S^1_{\Delta t}] w(t_n),$$

which is known to be first order accurate([17]), referred to here as Strang-simple splitting. The second order accurate estimate

$$u(t_{n+1}) \approx [S^1_{\frac{\Delta t}{2}} S^2_{\Delta t} S^1_{\frac{\Delta t}{2}}] w(t_n)$$

is referred to here as Strang-symmetric splitting. In each case the solution operators can be approximated by an appropriate ETD Scheme to obtain a fully discrete scheme. This again

has an easy extension to 3D where the equation will be split as

$$\begin{aligned} w_t + \mathcal{A}_1 w &= \frac{f(w)}{3} \quad t_n < t \le t_{n+1} \;\; w(t_n) = w_n \\ v_t + \mathcal{A}_2 v &= \frac{f(v)}{3} \quad t_n < t \le t_{n+1} \;\; v(t_n) = w(t_{n+1}) \\ s_t + \mathcal{A}_3 s &= \frac{f(s)}{3} \quad t_n < t \le t_{n+1} \;\; s(t_n) = v(t_{n+1}) \end{aligned}$$

and an appropriate ETD scheme applied to solve each subproblem.

3.3 Application to ETD-RDP

Applying the ETD-RDP scheme to (3.5) we obtain

$$\begin{aligned} v_{n+1} = &\left(I + \frac{1}{3}A_2 k\right)^{-1} [9v_n + 2k\tilde{F}(v_n) + k\tilde{F}(v_{n+1})] \\ &+ \left(I + \frac{1}{4}A_2 k\right)^{-1} [-8v_n - \frac{3k}{2}\tilde{F}(v_n) - \frac{k}{2}\tilde{F}(v_{n+1})]. \end{aligned} \tag{3.9}$$

Making the substitution

$$v_{n+1} = e^{t_{n+1}A_1} u_{n+1}$$

we have

$$\begin{aligned} e^{t_{n+1}A_1} u_{n+1} = &\left(I + \frac{1}{3}A_2 k\right)^{-1} [9e^{t_n A_1} u_n + 2ke^{t_n A_1} F(u_n) + ke^{t_{n+1}A_1} F(u_{n+1})] \\ &+ \left(I + \frac{1}{4}A_2 k\right)^{-1} [-8e^{t_n A_1} u_n - \frac{3k}{2} e^{t_n A_1} F(u_n) - \frac{k}{2} e^{t_{n+1}A_1} F(u_{n+1})] \end{aligned}$$

which reduces to

$$\begin{aligned} e^{kA_1} u_{n+1} = &\left(I + \frac{1}{3}A_2 k\right)^{-1} [9u_n + 2kF(u_n) + ke^{kA_1} F(u_{n+1})] \\ &+ \left(I + \frac{1}{4}A_2 k\right)^{-1} [-8u_n - \frac{3k}{2} F(u_n) - \frac{k}{2} e^{kA_1} F(u_{n+1})] \end{aligned}$$

finally

$$u_{n+1} = \left(I + \frac{1}{3}A_2k\right)^{-1}[e^{-kA_1}(9u_n + 2kF(u_n)) + kF(u_{n+1})] + \left(I + \frac{1}{4}A_2k\right)^{-1}[e^{-kA_1}(-8u_n - \frac{3k}{2}F(u_n)) - \frac{k}{2}F(u_{n+1})].$$

Substituting the RDP rational approximation to the exponential e^{-kA_1} we have

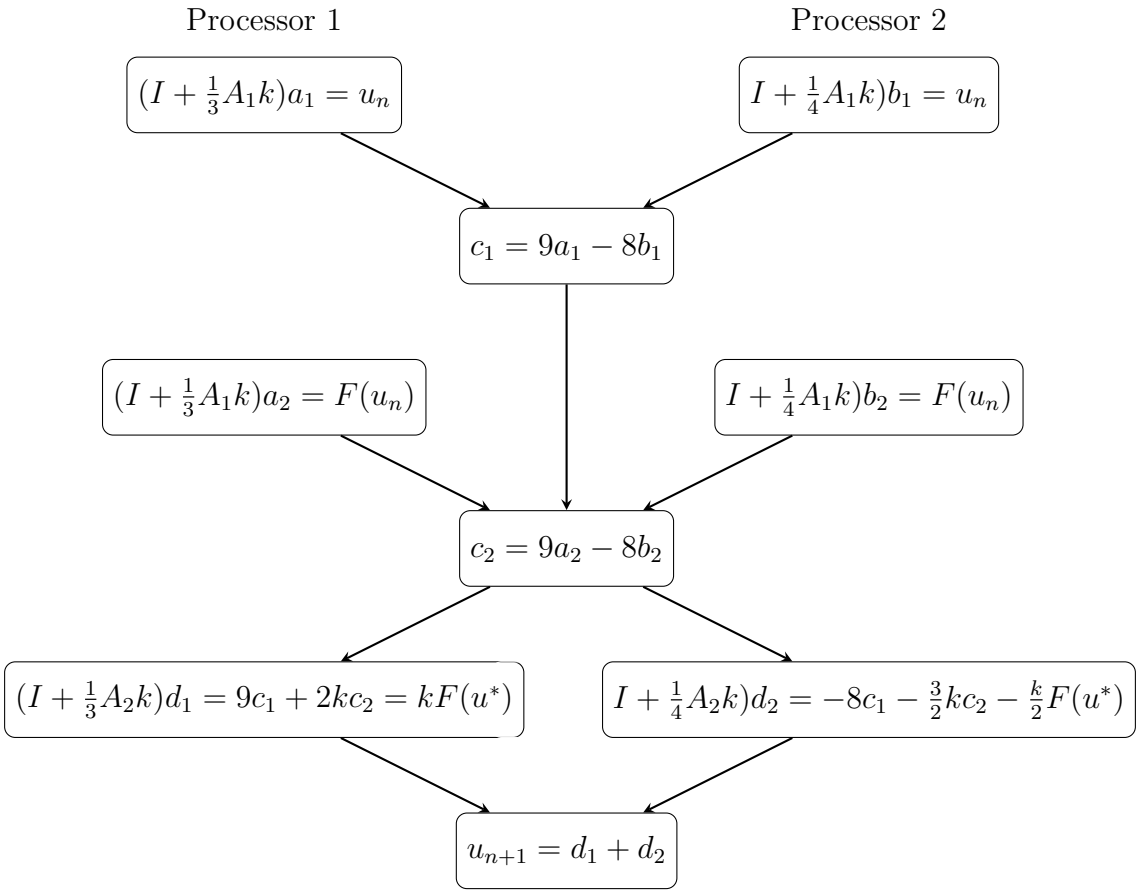
$$u_{n+1} = \left(I + \frac{1}{3}A_2k\right)^{-1}[\left[9\left(I + \frac{1}{3}kA_1\right)^{-1} - 8\left(I + \frac{1}{4}kA_1\right)^{-1}\right](9u_n + 2kF(u_n)) + kF(u_{n+1})] + \left(I + \frac{1}{4}A_2k\right)^{-1}[\left[9\left(I + \frac{1}{3}kA_1\right)^{-1} - 8\left(I + \frac{1}{4}kA_1\right)^{-1}\right](-8u_n - \frac{3k}{2}F(u_n)) - \frac{k}{2}F(u_{n+1})].$$

The full scheme is thus,

$$u^* = (I + A_1k)^{-1}(I + A_2k)^{-1}(u_n + kF(u_n))$$
$$u_{n+1} = \left(I + \frac{1}{3}A_2k\right)^{-1}[\left[9\left(I + \frac{1}{3}kA_1\right)^{-1} - 8\left(I + \frac{1}{4}kA_1\right)^{-1}\right](9u_n + 2kF(u_n)) + kF(u^*)] + \left(I + \frac{1}{4}A_2k\right)^{-1}[\left[9\left(I + \frac{1}{3}kA_1\right)^{-1} - 8\left(I + \frac{1}{4}kA_1\right)^{-1}\right](-8u_n - \frac{3k}{2}F(u_n)) - \frac{k}{2}F(u^*)].$$

Algorithm

We implement the above scheme as follows,



3.4 Application to ETD-CN

We explore the performance of our splitting techniques with the ETD-CN scheme. By applying the (1,1)-Padé approximation to the matrix exponential we obtain the following estimates

$$e^{-Ak} \approx \left(I - \frac{1}{2}kA\right)\left(I + \frac{1}{2}kA\right)^{-1},$$

$$
\begin{aligned}
A^{-1}(I - e^{-Ak}) &\approx A^{-1}\left(I - \left(I - \frac{1}{2}kA\right)\left(I + \frac{1}{2}kA\right)^{-1}\right) \\
&= A^{-1}\left(I + \frac{1}{2}kA - I + \frac{1}{2}kA\right)\left(I + \frac{1}{2}kA\right)^{-1} \\
&= k\left(I + \frac{1}{2}kA\right)^{-1},
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{A^{-2}}{k}(Ak - I + e^{-Ak}) &\approx \frac{A^{-2}}{k}\left(Ak - I + \left(I - \frac{1}{2}kA\right)\left(I + \frac{1}{2}kA\right)^{-1}\right) \\
&= \frac{A^{-2}}{k}(Ak + \frac{1}{2}k^2A^2 - I - \frac{1}{2}kA + I - \frac{1}{2}kA)\left(I + \frac{1}{2}kA\right)^{-1} \\
&= \frac{1}{2}k\left(I + \frac{1}{2}kA\right)^{-1}.
\end{aligned}
$$

Substituting these estimates into (3.6) yields the fully discrete split-ETD-CN scheme:

$$
\begin{aligned}
u_{n+1} = &\left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}\left(I - \frac{1}{2}kA_2\right)\left(I + \frac{1}{2}kA_2\right)^{-1}u_n \\
&+ k\left(I + \frac{1}{2}kA_2\right)^{-1}\left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}F(u_n) \\
&+ \frac{k}{2}\left(I + \frac{1}{2}kA_2\right)^{-1}\left[F(u^*) - \left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}F(u_n)\right]
\end{aligned} \tag{3.10}
$$

$$
\begin{aligned}
u^* = &\left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}\left(I - \frac{1}{2}kA_2\right)\left(I + \frac{1}{2}kA_2\right)^{-1}u_n \\
&+ k\left(I + \frac{1}{2}kA_2\right)^{-1}\left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}F(u_n).
\end{aligned}
$$

To reduce the number of matrix solves we reorganize the scheme, taking advantage of

commutativity:

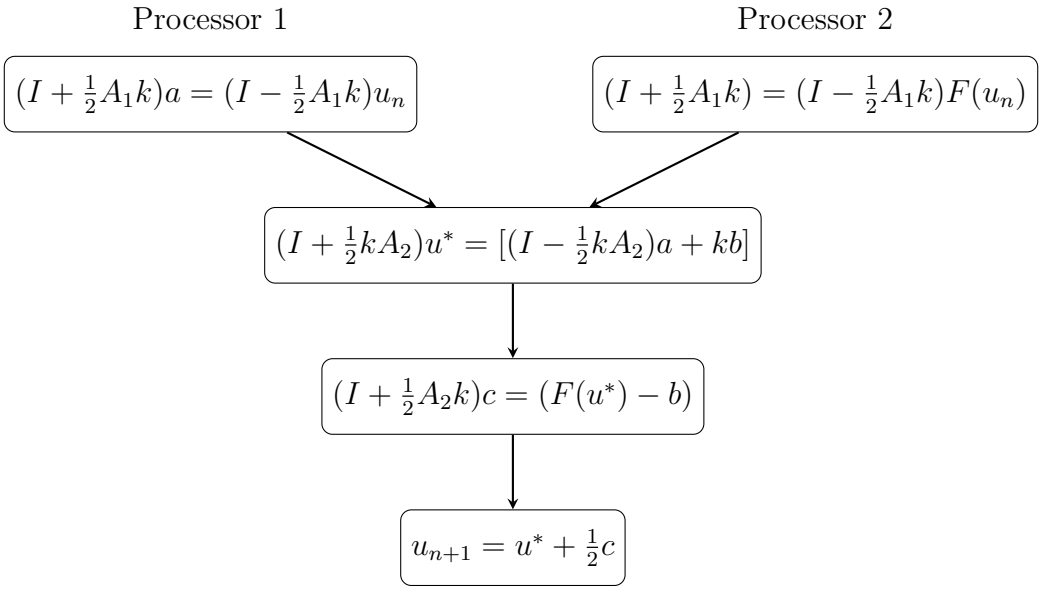
$$
\begin{aligned}
u_{n+1} = & \left(I - \frac{1}{2}kA_2\right)\left(I + \frac{1}{2}kA_2\right)^{-1}\left(I + \frac{1}{2}kA_1\right)^{-1}\left(I - \frac{1}{2}kA_1\right)u_n \\
& + k\left(I + \frac{1}{2}kA_2\right)^{-1}\left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}F(u_n) \\
& + \frac{k}{2}\left(I + \frac{1}{2}kA_2\right)^{-1}\left[F(u^*) - \left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}F(u_n)\right]
\end{aligned}
$$

which leads finally to the scheme

$$
\begin{aligned}
u_{n+1} = & \left(I + \frac{1}{2}kA_2\right)^{-1}\left[\left(I - \frac{1}{2}kA_2\right)\left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}u_n\right) \\
& + k\left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}F(u_n)\Bigg] \\
& + \frac{k}{2}\left(I + \frac{1}{2}kA_2\right)^{-1}\left[F(u^*) - \left(I - \frac{1}{2}kA_1\right)\left(I + \frac{1}{2}kA_1\right)^{-1}F(u_n)\right]
\end{aligned}
$$

Algorithm

We implement the above scheme as follows,



Chapter 4

Error Estimates

We are interested in deriving error estimates for ETD-RDP in the discretization of the semilinear evolution equation

$$\begin{aligned} u_t + Au &= f(u(t)) \quad \text{for } t \in J = (0, T], \\ u(0) &= \nu. \end{aligned} \tag{4.1}$$

We will consider the case where A is a self-adjoint positive definite operator with compact inverse in a Hilbert space H. Suppose that the semilinear equation has a sufficiently smooth solution $u : [0, T] \to V = \mathcal{D}(A)$ with derivatives in V and that $f : V \to H$ is sufficiently often Fréchet differentiable and satisfies the local Lipschitz condition. Under this assumption the composition $g : [0, T] \to H$ defined by $t \to g(t) = f(u(t))$ is a smooth mapping. It is well known that the solution to this problem satisfies the integral equation

$$u(t) = E(t)v + \int_0^t E(t - s)f(u(s))ds, \tag{4.2}$$

where $E(t) := e^{-At}$ is the analytic semigroup generated by A [39, Lemma 2.2.3, Theorem 2.6.2]. We will make extensive use of the recurrence relation,

$$u(t_{n+1}) = E(k)u(t_n) + k\int_0^1 E(k-sk)f(u(t_n+\tau k))d\tau \tag{4.3}$$

obtained through the change of variables $t_n = nk$ with $n \in \mathcal{N}^+, 0 < k \leq k_0 \in \Re^+$, and $s = t_n + \tau k$.

We will also make use of the following discrete Grownwall Lemma

Lemma 4.0.1. *Suppose ϕ_j, $\Psi_j \geq 0$ are discrete functions satisfying*

$$\delta_-\phi_j - c_0\phi_j - c_1\phi_{j-1} \leq \Psi_j, \text{ for } \quad j \geq 1, \quad \delta_-\phi_j = \frac{\phi_j - \phi_{j-1}}{k}$$

for some $c_0, c_1 \geq 0$. Then there exists k_0, $c_2, c_3 \geq 0$ s.t

$$\phi_j \leq e^{c_2 t_j}\phi_0 + c_3 k\sum_{i=1}^{j} e^{c_2(t_j - t_i)}\Psi_i.$$

Also

$$c_2 = (c_0 + c_1)(1 + c_1 k_0)^{-1}$$

and

$$c_3 = (1 - c_2 k_0)^{-1}.$$

Proof. Suppose $c_2 k_0 < 1$. let $I_j := (1 - c_2 k)^j$ be the integrating factor for $k < k_0$. Then

$$\begin{aligned}
\delta_-(I_j\phi_j) &= k^{-1}(I_j\phi_j - I_{j-1}\phi_{j-1}) \\
&= k^{-1}(I_j\phi_j - I_{j-1}\phi_j + I_{j-1}\phi_j - I_{j-1}\phi_{j-1}) \\
&= (\delta_- I_j)\phi_j + I_{j-1}(\delta_-\phi_j).
\end{aligned}$$

Also,

$$\begin{aligned}\delta_- I_j &= k^{-1}[(1-c_2k)^j - (1-c_2k)^{j-1}] \\ &= k^{-1}[(1-c_2k)^{j-1}(-c_2k)] \\ &= -c_2 I_{j-1}.\end{aligned}$$

Therefore,

$$\delta_-(I_j\phi_j) = I_{j-1}(\delta_-\phi_j - c_2\phi_j)$$

Assuming $c_1 = 0$ then by our assumption,

$$\delta_-(I_j\phi_j) \leq I_{j-1}\Psi_j$$

subsequently

$$k\sum_{j=1}^{n} I_{j-1}\delta_-(I_j\phi_j) \leq k\sum_{j=1}^{n} I_{j-1}\Psi_j$$

applying the telescoping property we have

$$I_n\phi_n - I_0\phi_0 \leq k\sum_{j=1}^{n} I_{j-1}\Psi_j.$$

Note that $I_0 = 1$ and

$$\begin{aligned} I_n^{-1} &= (1 - c_2 k)^{-n} \leq e^{c_2 n k} = e^{c_2 t_n} \\ I_n^{-1} I_{j-1} &= (1 - c_2 k)^{j-n} I_1^{-1} \\ &\leq e^{c_2 (t_n - t_j)} (1 - c_2 k)^{-1} \\ &\leq c_3 e^{c_2 (t_n - t_j)}. \end{aligned}$$

So,

$$\phi_n \leq e^{c_2 t_n} \phi_0 + c_3 k \sum_{j=1}^{n} e^{c_2 (t_n - t_j)} \Psi_j.$$

□

To facilitate our analysis we will reformulate the ETD-RDP scheme (2.18) as

$$\begin{aligned} U_{n,2} &= E_{k,2} U_n + k E_{k,2} f(U_n) \\ U_{n+1} &= E_{k,1} U_n + k \sum_{i=1}^{2} p_i(kA) f(U_{n,i}) \end{aligned} \tag{4.4}$$

where $U_{n,i} \approx u(t_n + \tau_i k)$ with $\tau_1 = 0, \tau_2 = 1$ and $U_{n,1} = U_n$.

(4.5)

The rational functions $E_{k,1}, E_{k,2}, p_1(kA), p_2(kA)$, defined as

$$\begin{aligned} E_{k,1} &:= r_1(kA) = 9 \left(I + \frac{1}{3} Ak \right)^{-1} - 8 \left(I + \frac{1}{4} Ak \right)^{-1} \\ E_{k,2} &:= r_2(kA) = (I + kA)^{-1} \\ p_1(Ak) &= 2 \left(I + \frac{1}{3} Ak \right)^{-1} - \frac{3}{2} \left(I + \frac{1}{4} Ak \right)^{-1} \\ p_2(Ak) &= \left(I + \frac{1}{3} Ak \right)^{-1} - \frac{1}{2} \left(I + \frac{1}{4} Ak \right)^{-1} \end{aligned}$$

are bounded on the spectrum of kA.

Next we prove another very useful Lemma,

Lemma 4.0.2. *The time discretization scheme* (4.4) *applied to the semilinear problem* (4.1) *is accurate of order 2.*

Proof. It suffices to show, according to [30, Lemma 9.1] that

1.
$$r_1(\lambda) = e^{-\lambda} + \mathcal{O}(\lambda^3), \text{ as } \lambda \to 0$$

and for $0 \leq l \leq 2$

2.
$$\sum_{i=1}^{2} \tau_i^l p_i(\lambda) = \frac{l!}{(-\lambda)^{l+1}} \left(e^{-\lambda} - \sum_{j=0}^{l} \frac{(-\lambda)^j}{j!} \right) + \mathcal{O}(\lambda^{2-l}), \text{ as } \lambda \to 0$$

.

The first part has been established in Lemma 2.2.1. For the second part, since $\tau_1 = 0$, it remains to show that

$$(p_1(\lambda) + p_2(\lambda)) - \frac{1}{\lambda}(e^{-\lambda}) = \mathcal{O}(\lambda^2),$$

where we have taken $\tau_1^0 = 1$,

$$p_2(\lambda) - \frac{1}{\lambda^2}(e^{-\lambda} - 1 + \lambda) = \mathcal{O}(\lambda),$$

and

$$p_2(\lambda) - \frac{2!}{(-\lambda)^3}(e^{-\lambda} - (1 - \lambda + \frac{\lambda^2}{2})) = \mathcal{O}(1)$$

as $\lambda \to 0$. Now, expressing the rational function in Taylor series, we see that

$$\begin{aligned} p_1(\lambda) &= \frac{2}{1 + \frac{1}{3}\lambda} - \frac{3}{2(1 + \frac{1}{4}\lambda)} \\ &= \frac{1}{2} - \frac{7}{24}\lambda + \frac{37}{288}\lambda^2 + \cdots . \end{aligned}$$

and

$$\begin{aligned} p_2(\lambda) &= \frac{1}{1+\frac{1}{3}\lambda} - \frac{1}{2(1+\frac{1}{4}\lambda)} \\ &= \frac{1}{2} - \frac{5}{24}\lambda + \frac{23}{288}\lambda^2 + \cdots . \end{aligned}$$

thus

$$p_1(\lambda) + p_2(\lambda) = 1 - \frac{\lambda}{2} + \frac{5\lambda^2}{24} + \cdots .$$

It follows that,

$$\begin{aligned} (p_1(\lambda) + p_2(\lambda)) - \frac{1}{\lambda}(e^{-\lambda} - 1) &= \frac{1}{24}\lambda^2 + \cdots \\ &= \mathcal{O}(\lambda^2), \text{ as } \lambda \to 0. \end{aligned}$$

Similarly

$$\begin{aligned} p_2(\lambda) - \frac{1}{\lambda^2}(e^{-\lambda} - 1 + \lambda) &= \frac{1}{8}\lambda + \cdots \\ &= \mathcal{O}(\lambda), \text{ as } \lambda \to 0. \end{aligned}$$

Finally,

$$p_2(\lambda) - \frac{2!}{(-\lambda)^3}(e^{-\lambda} - (1 - \lambda + \frac{\lambda^2}{2})) = \frac{1}{6} + \mathcal{O}(\lambda^2)) = \mathcal{O}(1), \text{ as } \lambda \to 0.$$

□

Lemma 4.0.3 (Stability Estimate)**.** *Let T_{j+1} and $T_{j,2}$ denote the local truncation errors at the main and predictor stages of the scheme* (4.4)*, then for k_0 sufficiently small with $0 < k \leq k_0$ and for $t \in (0, T]$, there exists a constant C, depending on T, such that the error*

at time t_n has bound

$$\| e_n \| \leq Ck \sum_{j=0}^{n-1} \Psi_{j+1} \tag{4.6}$$

with

$$\Psi_{j+1} = C \| T_{j,2} \| + k^{-1} \| T_{j+1} \|, \quad 1 \leq j \leq n, nk \leq T$$

Proof. Substituting the exact solution into the numerical scheme, with $u(t_n + \tau_1 k) = u(t_n)$, we have

$$u(t_n + \tau_2 k) = E_{k,2} u(t_n) + k E_{k,2} f(u(t_n)) + T_{n,2} \tag{4.7}$$

$$u(t_{n+1}) = E_{k,1} u(t_n) + k \sum_{i=1}^{2} p_i(kA) f(u(t_n + \tau_i k)) + T_{n+1} \tag{4.8}$$

where local truncation errors at the main and predictor stages are denoted by T_{n+1} and $T_{n,2}$ respectively.

We express the numerical error in the main scheme $e_{n+1} = U_{n+1} - u(t_{n+1})$ as

$$e_{n+1} = E_{k,1} e_n + k \sum_{i=1}^{2} p_i(kA)(f(U_{n,i}) - f(u(t_n + \tau_i k)) - T_{n+1}.$$

Thus

$$\| e_{n+1} \| \leq \| e_n \| + k \| \sum_{i=1}^{2} p_i(kA)(f(U_{n,i}) - f(u(t_n + \tau_i k)) \| + \| T_{n+1} \| .$$

Applying the Lipschitz assumption on f and boundedness of the rational functions, our estimate after shifting indices and setting $C = C(\nu, T)$ becomes

$$\| e_n \| \leq \| e_{n-1} \| + Ck \| e_{n-1} \| + Ck \| e_{n-1,2} \| + \| T_n \|$$

where $U_{n,1} - u(t_n + \tau_1 k) = e_{n,1} = e_n$.

Now,

$$e_{n-1,2} = U_{n-1,2} - u(t_{n-1} + \tau_2 k) = E_{k,2} e_{n-1} + k E_{k,2}(f(U_{n-1}) - f(u(t_{n-1}))) - T_{n-1,2}.$$

Taking norms and applying the Lipschitz assumption on f, we have for sufficiently small k

$$\begin{aligned} \| e_{n-1,2} \| &\leq \| e_{n-1} \| + Ck \| e_{n-1} \| + \| T_{n-1,2} \| \\ &\leq \| e_{n-1} \| + \| T_{n-1,2} \| . \end{aligned}$$

Applying this result to the global error bound, we obtain,

$$\| e_n \| \leq \| e_{n-1} \| + Ck \| e_{n-1} \| + Ck \| T_{n-1,2} \| + \| T_n \|$$

If we take $\phi_j = \| e_j \|$ and

$$\Psi_j = C \| T_{j-1,2} \| + k^{-1} \| T_j \|, \quad 1 \leq j \leq n \tag{4.9}$$

then an application of the discrete Grownwall lemma, noting that $\phi_0 = 0$, leads to the following estimate

$$\| e_n \| \leq c_3 k \sum_{j=0}^{n-1} e^{c_2 (t_{n-1} - t_j)} \Psi_{j+1},$$

where $c_2 = C(1 + Ck)^{-1}$ and $c_3 = (1 + Ck)$. By assumption, we can choose k small enough so that $c_2 \leq C$ and $c_3 \leq C$. Thus on a bounded time domain, we have the desired result. $\square$

Theorem 4.0.4. *Under the stated assumption on f, if we assume further that $g^{(l)}(t) \in \mathcal{D}(A^{2-l})$ for $l < 2$, and the initial data $v \in \mathcal{D}(A^3)$ then for the numerical scheme* (4.4)

applied to the semilinear problem (4.1)*, the following estimate for the error bound holds, for* $t \in (0, T]$

$$\| e_n \| \leq Ck^2 \left((\log n + 1)(\| A^2 v \| + \| Av \|) + \bar{t} \sup_{0 \leq s \leq t_n} \| A^2 g(s) \| + \bar{t} \sup_{0 \leq s \leq t_n} \| Ag(s) \| \right.$$
$$\left. + \bar{t} \sup_{0 \leq s \leq t_n} \| Ag^{(1)}(s) \| + \int_0^{t_n} \| g^{(1)}(\tau) \| d\tau + \int_0^{t_n} \| g^{(2)}(\tau) \| d\tau \right)$$

where the constant C *depends on* T*.*

Proof. We proceed by finding optimal error bounds for the local truncation errors for the main and predictor steps i.e. T_{j+1} and $T_{j,2}$ respectively.

From Eq. (4.7)

$$\begin{aligned} T_{j,2} &= u(t_j + \tau_2 k) - E_{k,2} u(t_j) - k E_{k,2} f(u(t_j)) \\ &= (E(k) - E_{k,2}) u(t_j) \\ &+ k \left(\int_0^1 E(k - sk) f(u(t_j + sk)) ds - E_{k,2} f(u(t_j)) \right). \end{aligned}$$

Setting $\int_0^1 E(k - sk) f(u(t_j + sk)) ds = I_k f(u(t_{j,s}))$ and taking norms we have

$$\| T_{j,2} \| \leq \| (E(k) - E_{k,2}) u(t_j) \| + k \| I_k f(u(t_{j,s})) - E_{k,2} f(u(t_j)) \| .$$

Similarly, if we set $\sum_{i=1}^{2} p_i(kA) f(u(t_j + \tau_i k))) = Q_k f(u(t_{j,\tau_i}))$ from Eq. (4.8) we have

$$\begin{aligned} T_{j+1} &= u(t_{j+1}) - E_{k,1} u(t_j) - Q_k f(u(t_{j,\tau_i})) \\ &= (E(k) - E_{k,1}) u(t_j) + k \left(I_k f(u(t_{j,s})) - Q_k f(u(t_{j,\tau_i})) \right). \end{aligned}$$

Taking norms we find that

$$\| T_{j+1} \| \leq \| (E(k) - E_{k,1}) u(t_j) \| + k \| I_k f(u(t_{j,s})) - Q_k f(u(t_{j,\tau_i})) \|.$$

First,

$$\begin{aligned}\| (E(k) - E_{k,2})u(t_j) \| &= \| (e^{-kA} - r_2(kA))u(t_j) \| \\ &= \| (kA)^{-2}(e^{-kA} - r_2(kA))(kA)^2 u(t_j) \| \\ &= k^2 \| (kA)^{-2}(e^{-kA} - r_2(kA))A^2 u(t_j) \|\end{aligned}$$

Employing spectral decomposition of A,

$$k^2 \| (kA)^{-2}(e^{-kA} - r_2(kA))A^2 u(t_j) \| \leq k^2 \sup_{\lambda \in \sigma(kA)} |\lambda^{-2}(e^{-\lambda} - r_2(\lambda))| \| A^2 u(t_j) \| .$$

Since $(e^{-\lambda} - r_2(\lambda)) \leq C\lambda^2$ for some $C > 0$,it follows that

$$\| (E(k) - E_{k,2})u(t_j) \| \leq Ck^2 \| A^2 u(t_j) \|, \quad 0 \leq j \leq n-1. \tag{4.10}$$

Next,

$$\begin{aligned}\| I_k f(u(t_{j,s})) - E_{k,2} f(u(t_j)) \| &= \| \int_o^1 E(k - sk) f(u(t_j + sk)) ds - E_{k,2} f(u(t_j)) \| \\ &= \| \int_o^1 e^{-kA(1-s)} g(t_j + sk) ds - r_2(kA) g(t_j) \| \\ &\leq \| \int_o^1 e^{-kA(1-s)} (g(t_j + sk) - g(t_j)) ds \| \\ &+ \| \int_o^1 e^{-kA(1-s)} g(t_j) ds - r_2(kA) g(t_j) \|.\end{aligned}$$

Expanding $g(t_j+sk)$ in taylor series, we find that $g(t_j+sk)-g(t_j)=\int_{t_j}^{t_j+sk} g^{(1)}(\tau)d\tau$, hence

$$\begin{aligned}\| \int_o^1 e^{-kA(1-s)}(g(t_j+sk)-g(t_j))ds\| &= \| \int_o^1 e^{-kA(1-s)} \int_{t_j}^{t_j+sk} g^{(1)}(\tau)d\tau ds\| \\ &\leq \int_o^1 \|e^{-kA(1-s)} \int_{t_j}^{t_j+sk} g^{(1)}(\tau)d\tau\| ds \\ &\leq \int_o^1 \|e^{-kA(1-s)}\| \int_{t_j}^{t_j+sk} \|g^{(1)}(\tau)\|d\tau ds \\ &\leq C\int_o^1 \int_{t_j}^{t_j+sk} \|g^{(1)}(\tau)\|d\tau ds \\ &\leq C\int_{t_j}^{t_{j+1}} \|g^{(1)}(s)\|ds.\end{aligned}$$

Also

$$\| \int_o^1 e^{-kA(1-s)}g(t_j)ds - r_2(kA)g(t_j)\| = \| \left(\int_o^1 e^{-kA(1-s)}ds - r_2(kA)\right) g(t_j)\|$$

Set $\left(\int_o^1 e^{-kA(1-s)}ds - r_2(kA)\right) = b(kA)$, then

$$\begin{aligned}\| \, b(kA)g(t_j) \, \| &= \| \, (kA)^{-1}b(kA)(kA)g(t_j) \, \| \\ &= k \, \| \, (kA)^{-1}b(kA)Ag(t_j) \, \| \\ &\leq k \, \| \, (kA)^{-1}b(kA) \, \| \| \, Ag(t_j) \, \| \\ &\leq k \sup_{\lambda\in\sigma(kA)} |\lambda^{-1}b(\lambda)| \, \| \, Ag(t_j) \, \| \, .\end{aligned}$$

Now, $|b(\lambda)| = \left|\frac{1}{\lambda}(1-e^{-\lambda}) - (1-\lambda+2\lambda^2+\mathcal{O}(\lambda^3))\right| \leq C\lambda$ for some $C>0$. Therefore

$$\| \, b(kA)g(t_j) \, \| \leq Ck \, \| \, Ag(t_j) \, \|$$

and subsequently,

$$\| I_k f(u(t_{j,s})) - E_{k,2} f(u(t_j)) \| \leq Ck \| Ag(t_j) \| + C \int_{t_j}^{t_{j+1}} \|g^{(1)}(s)\| ds, \quad 0 \leq j \leq n-1. \tag{4.11}$$

By similar development, we have the following estimates for the local truncation error at the main step,

$$\begin{aligned} \| (E(k) - E_{k,1}) u(t_j) \| &= \| (kA)^{-3} (e^{-kA} - r_1(kA))(kA)^3 u(t_j) \| \\ &= k^3 \| (kA)^{-3} (e^{-kA} - r_1(kA)) A^3 u(t_j) \| \\ &\leq k^3 \sup_{\lambda \in \sigma(kA)} |\lambda^{-3} (e^{-\lambda} - r_1(\lambda))| \| A^3 u(t_j) \| . \end{aligned}$$

Since $(e^{-\lambda} - r_1(\lambda)) = \mathcal{O}(\lambda^3)$ by (4.0.2) , it follows that

$$\| (E(k) - E_{k,1}) u(t_j) \| \leq Ck^3 \| A^3 u(t_j) \|, \quad 0 \leq j \leq n-1.. \tag{4.12}$$

Finally,

$$\begin{aligned} \|I_k f(u(t_{j,s})) - Q_k f(u(t_{j,\tau_i}))\| &= \|I_k g(t_j + sk) - Q_k g(t_j + \tau_i k)\| \\ &= \| \int_0^1 e^{-kA(1-s)} g(t_j + sk) ds - \sum_{i=1}^{2} p_i(kA) g(t_j + \tau_i k)\|. \end{aligned}$$

Expanding $g(t_j + sk)$ and $g(t_j + \tau_i k)$ in Taylor series

$$\begin{aligned} g(t_j + sk) &= g(t_j) + skg^{(1)}(t_j) + \int_{t_j}^{t_j + sk} (t_j + sk - \epsilon) g^{(2)}(\epsilon) d\epsilon \\ g(t_j + \tau_i k) &= g(t_j) + \tau_i k g^{(1)}(t_j) + \int_{t_j}^{t_j + \tau_i k} (t_j + \tau_i k - \epsilon) g^{(2)}(\epsilon) d\epsilon \end{aligned}$$

we have

$$\begin{aligned}\|I_k g(t_j+sk)-Q_k g(t_j+\tau_i k)\| \le \|I_k\left(\sum_{l=0}^{1}(sk)^l g^l(t_j)\right)-Q_k\left(\sum_{l=0}^{1}(\tau_i k)^l g^l(t_j)\right)\| \\ +\|I_k\left(\int_{t_j}^{t_j+sk}(t_j+sk-\epsilon)g^{(2)}(\epsilon)d\epsilon\right) \\ -Q_k\left(\int_{t_j}^{t_j+\tau_i k}(t_j+\tau_i k-\epsilon)g^{(2)}(\epsilon)d\epsilon\right)\|.\end{aligned}$$

For the first part,

$$\begin{aligned}\|I_k\left(\sum_{l=0}^{1}(sk)^l g^l(t_j)\right)-Q_k\left(\sum_{l=0}^{1}(\tau_i k)^l g^l(t_j)\right)\| &=\|\sum_{l=0}^{1}\left(\int_0^1 e^{-kA(1-s)}(sk)^l ds-\sum_{i=1}^{2}p_i(kA)(\tau_i k)^l\right)g^l(t_j)\| \\ &=\|\sum_{l=0}^{1}k^l\left(\int_0^1 s^l e^{-kA(1-s)}ds-\sum_{i=1}^{2}p_i(kA)\tau_i^l\right)g^l(t_j)\|.\end{aligned}$$

Set$\left(\int_0^1 s^l e^{-kA(1-s)}ds-\sum_{i=1}^{2}p_i(kA)\tau_i^l\right)=b_l(kA)$ then

$$\begin{aligned}\|\sum_{l=0}^{1}k^l\left(\int_0^1 s^l e^{-kA(1-s)}ds-\sum_{i=1}^{2}p_i(kA)\tau_i^l\right)g^l(t_j)\| &=\|\sum_{l=0}^{l}k^l b_l(kA)g^l(t_j)\| \\ &\le\sum_{l=0}^{l}\|k^l b_l(kA)g^l(t_j)\|.\end{aligned}$$

Now,

$$\begin{aligned}\|k^l b_l(kA)g^l(t_j)\| &=\|k^l(kA)^{-(2-l)}b_l(kA)(kA)^{2-l}g^l(t_j)\| \\ &=k^2\|(kA)^{-(2-l)}b_l(kA)A^{2-l}g^l(t_j)\| \\ &\le k^2\sup_{\lambda\in\sigma(kA)}|\lambda^{-(2-l)}b_l(\lambda)|\,\|A^{2-l}g^l(t_j)\|.\end{aligned}$$

Since (4.4) is a scheme of order 2 it follows from [30, Lemma 9.1] that $|b_l(\lambda)| \leq C\lambda^{2-l}$. Hence

$$\| k^l b_l(kA) g^l(t_j) \| \leq Ck^2 \| A^{2-l} g^l(t_j) \| .$$

Therefore,

$$\sum_{l=0}^{l} \| k^l b_l(kA) g^l(t_j) \| \leq Ck^2 \left(\| A^2 g(t_j) \| + \| A g^{(1)}(t_j) \| \right) .$$

Now, for the second part

$$\| \int_0^1 e^{-kA(1-s)} \int_{t_j}^{t_j+sk} (t_j + sk - \epsilon) g^{(2)}(\epsilon) d\epsilon ds - \sum_{i=1}^{2} p_i(kA) \int_{t_j}^{t_j+\tau_i k} (t_j + \tau_i k - \epsilon) g^{(2)}(\epsilon) d\epsilon \| =$$
$$\| \int_0^1 \left(\int_{t_j}^{t_j+sk} e^{-kA(1-s)} (t_j + sk - \epsilon) g^{(2)}(\epsilon) d\epsilon \right) ds - \sum_{i=1}^{2} \left(\int_{t_j}^{t_j+\tau_i k} p_i(kA)(t_j + \tau_i k - \epsilon) g^{(2)}(\epsilon) d\epsilon \right) \|.$$

For the scheme (4.4) recall that $\tau_1 = 0$ and $\tau_2 = 1$, leaving us with

$$\| \int_0^1 (\int_{t_j}^{t_j+sk} e^{-kA(1-s)} (t_j + sk - \epsilon) g^{(2)}(\epsilon) d\epsilon) ds - \int_{t_j}^{t_{j+1}} p_2(kA)(t_j + \tau_i k - \epsilon) g^{(2)}(\epsilon) d\epsilon \|.$$

By setting $\epsilon = t_j + \tau k, d\epsilon = k d\tau$ in the first expression, we will have

$$\| \int_0^1 \int_0^s e^{-kA(1-s)} k(s-\tau) g^{(2)}(t_j + \tau k) k d\tau ds - \int_{t_j}^{t_{j+1}} p_2(kA)(t_{j+1} - \epsilon) g^{(2)}(\epsilon) d\epsilon \|,$$

which can be bounded above by

$$k \| \int_0^1 \int_0^s e^{-kA(1-s)} (s-\tau) g^{(2)}(t_j + \tau k) k d\tau ds \| + Ck \int_{t_j}^{t_{j+1}} \| g^{(2)}(\epsilon) \| d\epsilon$$

Examining the first term, we see that

$$\| \int_0^1 \int_0^s e^{-kA(1-s)}(s-\tau)g^{(2)}(t_j+\tau k)kd\tau ds\| \leq \int_0^1 \int_0^s \| e^{-kA(1-s)}(s-\tau)g^{(2)}(t_j+\tau k) \| kd\tau ds$$
$$\leq C \int_0^1 \int_0^s \| g^{(2)}(t_j+\tau k) \| kd\tau ds.$$

Now substituting $\epsilon = t_j + \tau k$, and $d\epsilon = kd\tau$,

$$C \int_0^1 \int_0^s \| g^{(2)}(t_j+\tau k) \| kd\tau ds \leq C \int_0^1 \int_{t_j}^{t_j+sk} \| g^{(2)}(\epsilon) \| d\epsilon ds$$
$$\leq C \int_{t_j}^{t_{j+1}} \| g^{(2)}(\epsilon) \| d\epsilon.$$

Similar we can bound the second term as follows

$$\| \int_0^1 p_2(kA)(1-\tau g^{(2)}(t_j+\tau k)kd\tau\| \leq \int_0^1 \| p_2(kA)(1-\tau) \|\| g^{(2)}(t_j+\tau k)kd\tau \|$$
$$\leq C \int_{t_j}^{t_{j+1}} \| g^{(2)}(\epsilon) \| d\epsilon.$$

Therefore,

$$\| I_k g(t_j+sk) - Q_k f(u(t_j+\tau_i k) \| \leq Ck^2 \left(\| A^2 g(t_j) \| + \| Ag^{(1)}(t_j) \| \right) + Ck \int_{t_j}^{t_{j+1}} \| g^{(2)}(\epsilon) \| d\epsilon.$$

So the local truncation errors for the main and predictor steps will have the bounds,

$$\| T_{j,2} \| \leq Ck^2 \| A^2 u(t_j) \| + Ck^2 \| Ag(t_j) \| + Ck \int_{t_j}^{t_{j+1}} \|g^{(1)}(\tau)\| d\tau \tag{4.13}$$

$$\| T_{j+1} \| \leq Ck^3 \| A^3 u(t_j) \| + Ck^3 \left(\| A^2 g(t_j) \| + \| Ag^{(1)}(t_j) \| \right) + Ck^2 \int_{t_j}^{t_{j+1}} \| g^{(2)}(\epsilon) \| d\epsilon \tag{4.14}$$

for $0 \leq j \leq n-1$. Finally, substituting Equation(4.13) and (4.14) into Equation(4.9) and making use of the stability estimate (4.6), we obtain

$$\| e_n \| \leq Ck^2 \sum_{j=0}^{n-1} \Big(k \| A^3 u(t_j) \| + k \| A^2 u(t_j) \| + k \| A^2 g(t_j) \| + k \| A g^{(1)}(t_j) \| + k \| A g(t_j) \| + \int_{t_j}^{t_{j+1}} \| g^{(1)}(\tau) \| d\tau + \int_{t_j}^{t_{j+1}} \| g^{(2)}(\tau) \| d\tau \Big). \tag{4.15}$$

Now,

$$\begin{aligned} \| A^3 u(t_j) \| &= \| A^3 \left(E(t_j)v + \int_0^{t_j} E(t_j - s) g(s) ds \right) \| \\ &= \| A^3 E(t_j) v \| + \| A^3 \int_0^{t_j} E(t_j - s) g(s)) ds \| . \end{aligned}$$

But for $1 \leq j \leq n-1$

$$\begin{aligned} \| A^3 E(t_j) v \| &= \| A E(t_j) A^2 v \| \\ &\leq \| A E(t_j) \| \| A^2 v \| \end{aligned}$$

by the analyticity of the semigroup we have from [39, Theorem 2.6.2]

$$\| A^3 E(t_j) v \| \leq \frac{C}{t_j} \| A^2 v \|$$

For $j = 0$, it follows from our assumption on ν that $\| A^3 v \| \leq C$ hence

$$\| A^3 E(t_j) v \| \leq \frac{C}{t_{j+1}} \| A^2 v \|, \quad 0 \leq j \leq n-1.$$

Also

$$\begin{aligned}\| A^3 \int_0^{t_j} E(t_j - s)g(s))ds \| &= \| \int_0^{t_j} AE(t_j)E(-s)A^2 g(s)ds \| \\ &\leq \int_0^{t_j} \| AE(t_j) \| \| E(-s)A^2 g(s) \| ds \\ &\leq \frac{C}{t_{j+1}} \int_0^{t_{j+1}} \| A^2 g(s) \| ds\end{aligned}$$

therefore

$$\| A^3 u(t_j) \| \leq \frac{C}{t_{j+1}} \| A^2 v \| + \frac{C}{t_{j+1}} \int_0^{t_{j+1}} \| A^2 g(s) \| ds.$$

Similarly,

$$\| A^2 u(t_j) \| \leq \frac{C}{t_{j+1}} \| Av \| + \frac{C}{t_{j+1}} \int_0^{t_{j+1}} \| Ag(s) \| ds$$

Substituting these results, Equation (4.15) becomes

$$\begin{aligned}\| e_n \| \leq Ck^2 \Bigg(&\sum_{j=0}^{n-1} \frac{1}{j+1} \| A^2 v \| + k \sum_{j=0}^{n-1} \frac{1}{t_{j+1}} \int_0^{t_{j+1}} \| A^2 g(s) \| ds + \sum_{j=0}^{n-1} \frac{1}{j+1} \| Av \| \\ &+ k \sum_{j=0}^{n-1} \frac{1}{t_{j+1}} \int_0^{t_{j+1}} \| Ag(s) \| ds + k \sum_{j=0}^{n-1} \| A^2 g(t_j) \| + k \sum_{j=0}^{n-1} \| Ag^{(1)}(t_j) \| + k \sum_{j=0}^{n-1} \| Ag(t_j) \| \\ &+ \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} \| g^{(1)}(\tau) \| d\tau + \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} \| g^{(2)}(\tau) \| d\tau \Bigg).\end{aligned}$$

Upon taking summation through each individual term and simplifying the remaining expression, the desired result follows. □

which simplifies to give,

$$\| e_n \| \leq Ck^2 \Bigg((\log n + 1)(\| A^2 v \| + \| Av \|) + \bar{t} \sup_{0 \leq s \leq t_n} \| A^2 g(s) \| + \bar{t} \sup_{0 \leq s \leq t_n} \| Ag(s) \| + \bar{t} \sup_{0 \leq s \leq t_n} \| Ag^{(1)}(s) \| + \int_0^{t_n} \| g^{(1)}(\tau) \| \, d\tau + \int_0^{t_n} \| g^{(2)}(\tau) \| \, d\tau \Bigg)$$

Chapter 5

Numerical Experiments

We investigate the robustness of the ETD-RDP scheme on reaction-diffusion systems with non-smooth initial and boundary data, stiff reaction terms, widely varying diffusion coefficients and in higher space dimensions. The accuracy of the scheme in all cases was measured using relative error $\frac{\|u-\hat{u}\|_2}{\|u\|_2}$, with u and $\hat{u}$ being the reference and approximate solutions, respectively. For test problems with an exact solution, the reference solution was chosen to be the exact solution. For more complex problems without exact solutions the numerical solution using BDF2 on a finer mesh was used as the reference solution. This approach, as reported by Leveque [23], is well suited for recovering the rate of convergence of numerical schemes, though it may not reflect the true error in using the numerical scheme. The rate of convergence p was calculated using the relation

$$p \approx \frac{ln(\frac{\tilde{E}(k)}{\tilde{E}(k/2)})}{ln2}$$

where $\tilde{E(k)}$ is the approximate error on the temporal resolution k. We will make extensive use of the term 'computationally efficient'. By this we mean that for a given time step, a numerical scheme obtains higher accuracy with faster computational time. All computations were performed on a Dell Precision T5610 with intel Xeon(R) Cpu E5-2630 v2 with 2.60GHz $\times$ 17 cores and 32 GiB RAM. Computational time is reported in seconds for all test problems.

For the general test problem

$$\begin{aligned} u_t + Au &= f(u) \qquad \in \Omega \\ u|_\Gamma &= 0 \\ u|_{t=0} &= u_0(x) \end{aligned} \tag{5.1}$$

where Ω is a bounded domain in $\mathbb{R}^d$ with smooth boundary Γ, and A represents the general second order elliptic operator

$$Au \equiv -\sum_{i,j=1}^{n} \frac{\partial}{\partial x_i}(a_{ij}(x)\frac{\partial u}{\partial x_j}) + \sum_{i=1}^{n} b_i(x)\frac{\partial u}{\partial x_i} + c(x)u$$

with smooth coefficients $a_{ij}(x), b_i(x), c(x)$, we adopt the following abstract formulation to facilitate investigation of the existence and regularity of solutions,

$$\begin{aligned} u_t + Au &= F(u) \\ u(0) &= u_0 \end{aligned} \tag{5.2}$$

and set domain of A as $D(A) = H^2 \cup H_0^1$ in the Banach space $B = L^2(\Omega)$. It has been established in [39, Pg. 74] that A is a sectorial operator and thus generates an analytic semigroup in $L^2(\Omega)$. In the special case of $A = -\Delta$ which is m-accretive (see [39, Theorem 2.7.2]) we have the result that A generates a C_0- semigroup of contractions.

5.1 Model semilinear problem with exact solution

In order to validate the second order accuracy of the ETD-RDP scheme, we applied it to the semilinear test problem from [16]

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = \frac{1}{1+u^2} + \Phi(x,t) \tag{5.3}$$

for $x \in [0,1]$ and $t \in [0,1]$ subject to homogeneous Dirichlet conditions. For

$$\Phi(x,t) = x(1-x)e^t + 2e^t - \frac{1}{1+x^2(1-x)^2e^{2t}}$$

the exact solution for the test problem becomes $u(x,t) = x(1-x)e^t$. The second-order convergence of the scheme is validated by the grid refinement studies in Table 5.1. In comparison with existing second order ETD schemes, ETD-RDP is computationally more efficient than ETD-Padé(0,2) (Figure 5.1(a),Table A1). While ETD-CN appears superior to ETD-RDP in this case, its inability to damp spurious oscillations makes it unsuitable for solving problems with non-smooth data (Figure 5.5b,c) or problems having significant advection components. Some of the more common time stepping schemes, used particularly by engineers, are backward Euler (BE), Crank-Nicolson (CN) and BDF2. Comparing ETD-RDP to these schemes, we find it to be a hundred times faster Figure 5.1(b) with comparable accuracy (Table A2) once an appropriate time resolution is chosen. One reason for this superior computational speed is the absence of a Newton iteration in the ETD-RDP algorithm. The larger class of time discretization schemes i.e. implicit Runge-Kutta (IRK) schemes, which typically require Newton solve, also lag behind ETD-RDP in computational speed (Figure 5.1(d)). It is worth noting that ETD-RDP has better accuracy than the SDIRK scheme (Figure 5.1(c), Table A2). Finally, comparing ETD-RDP to IMEX schemes, we find ETD-RDP is more accurate (Figure 5.2(a)) and computationally more efficient (Figure 5.2(b), Table A3).

k	h	Error	Order	Time
0.050000	0.050000	2.9755×10^{-4}	-	0.00237
0.025000	0.025000	8.4801×10^{-5}	1.81	0.00464
0.012500	0.012500	2.2857×10^{-5}	1.89	0.00977
0.006250	0.006250	5.9423×10^{-6}	1.94	0.02184
0.003125	0.003125	1.5136×10^{-6}	1.97	0.05134

Table 5.1: Examining order of convergence of ETD-RDP for semilinear test problem.

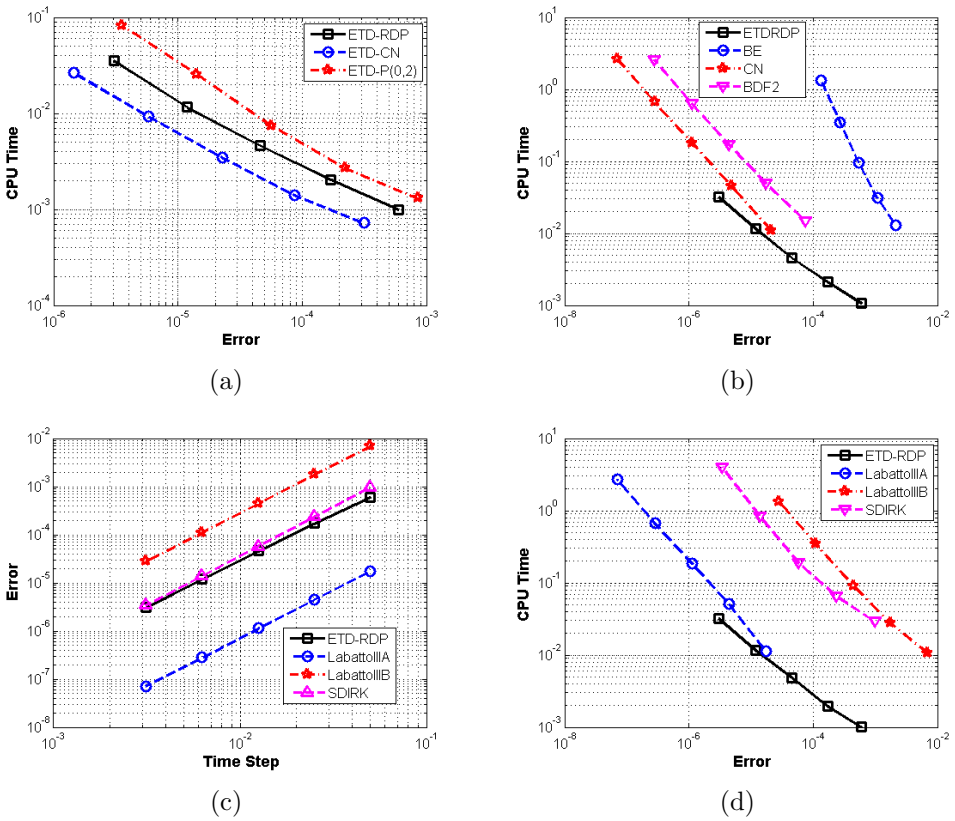


Figure 5.1: Log-log plots showing (a) computational efficiency of ETD-RDP over ETD-CN and ETD-Pade(0,2); (b) faster computational time compared with BE, CN and BDF2 (c) better accuracy and (d) more efficiency in comparison to standard Runge-Kutta schemes. In all cpu time vs error plots, lower (for better time efficiency) and left (for better accuracy) is preferable.

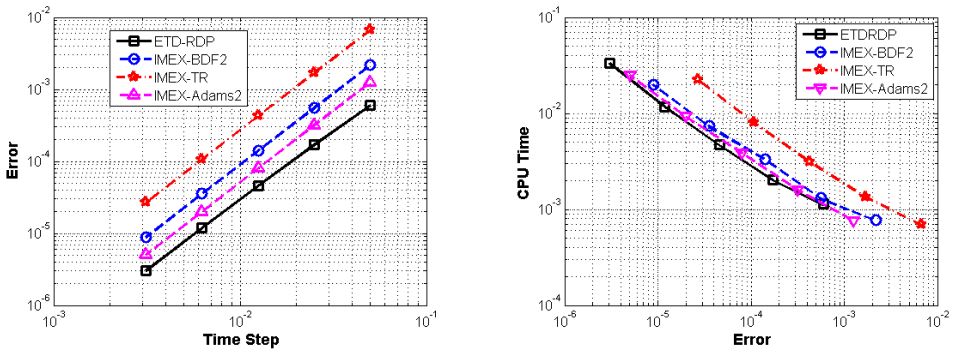
5.2 Allen-Cahn Equation

We consider the Allen-Cahn equation with solution regions near ± 1 that are flat and where the interface does not change for a relatively long period of time, then changes suddenly. The equation, taken from [21], is:

$$\frac{\partial u}{\partial t} = \epsilon \frac{\partial u^2}{\partial x^2} + u - u^3, \quad x \in [-1, 1]$$

$$u(x, 0) = 0.53x + 0.47 \sin(-1.5\pi x), t > 0$$

$$u(-1, t) = -1, u(1, t) = 1.$$



(a) (b)

Figure 5.2: Log-log plots showing (a) accuracy and (b) efficiency of ETD-RDP compred to standard second order IMEX schemes for semilinear test problem.

We put our problem in the form (5.1) by setting $\tilde{u} = u - x$ thus

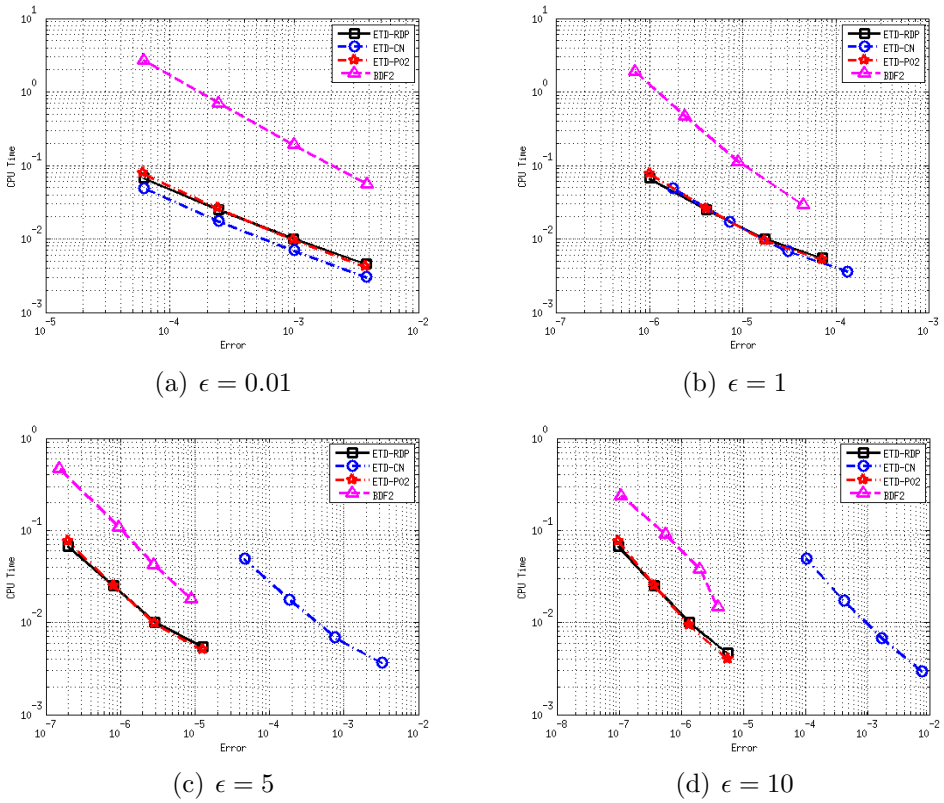
$$\frac{\partial \tilde{u}}{\partial t} = \epsilon \frac{\partial \tilde{u}^2}{\partial x^2} + (\tilde{u} + x) - (\tilde{u} + x)^3, \quad x \in [-1, 1] \tag{5.4}$$

$$\tilde{u}(x, 0) = 0.53x + 0.47 \sin(-1.5\pi x) - x, t > 0 \tag{5.5}$$

$$\tilde{u}(-1, t) = 0, \tilde{u}(1, t) = 0. \tag{5.6}$$

We now have homogenous Dirichlet boundary conditions and with little algebra can immediately see that $\tilde{u}(x, 0) \in H^2 \cup H_0^1$. It follow from Theorem 1.3.15 that the test problem admits a unique global classical solution.

The second-order accuracy of ETD-RDP for this problem is shown in Table 5.2. We also investigate the sensitivity of existing ETD-Padé schemes to varying diffusivity ϵ, in comparison to ETD-RDP. We find that ETD-CN tends to degrade in accuracy as ϵ increase, while ETD-RDP remains accurate as shown in Figure 5.3. We also notice from Figure 5.3 and Table A4 that ETD solutions to Allen-Cahn equation have comparable accuracy to BDF2 with much faster computational time. As expected ETD-RDP is computationally more efficienct than standard Runge-Kutta and Rosenbrock schemes (Figure 5.4(a)) without compromising the accuracy (Figure 5.4(b),Table A5)).



(a) $\epsilon = 0.01$ (b) $\epsilon = 1$

(c) $\epsilon = 5$ (d) $\epsilon = 10$

Figure 5.3: Log-log plots showing efficiency of ETD-RDP over ETD-CN with varying diffusivity for the Allen-Cahn Equation.

5.3 Enzyme Kinetics

We consider the two-dimensional enzyme kinetics reaction-diffusion problem [4]

$$\frac{\partial u}{\partial t} = d\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{u}{(1+u)}, \quad 0 < x, y < 1, t > 0, \tag{5.7}$$

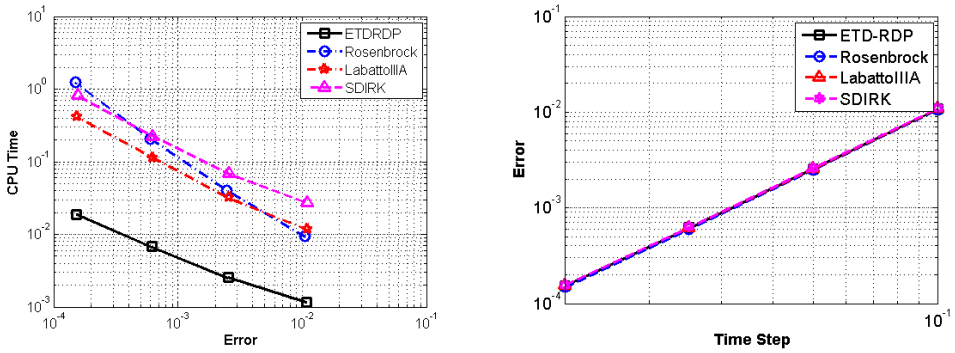
with homogeneous Dirichlet boundary conditions and initial condition,

$$u(x, y, 0) = 1, \quad 0 \leq x, y \leq 1$$

which are mismatched and tend to generate spurious oscillations. Clearly, $f \in C^1(R)$, $f'(u)$ is uniformly bounded in $\Omega = [0, 1]$ and $u(x, y, 0) \in L^2(\Omega)$, thus from Theorem 1.3.12 the problem admits a unique mild solution.

k	h	Error	Order	Time
0.1000	0.1000	1.5246×10^{-2}	-	0.00287
0.0500	0.0500	3.7895×10^{-3}	2.01	0.00422
0.0250	0.0250	9.9047×10^{-4}	1.94	0.00936
0.0125	0.0125	2.4628×10^{-4}	2.01	0.02339

Table 5.2: Examining order of convergence of ETD-RDP for Allen-Cahn Equation after 2 secs with $\epsilon = 0.01$.

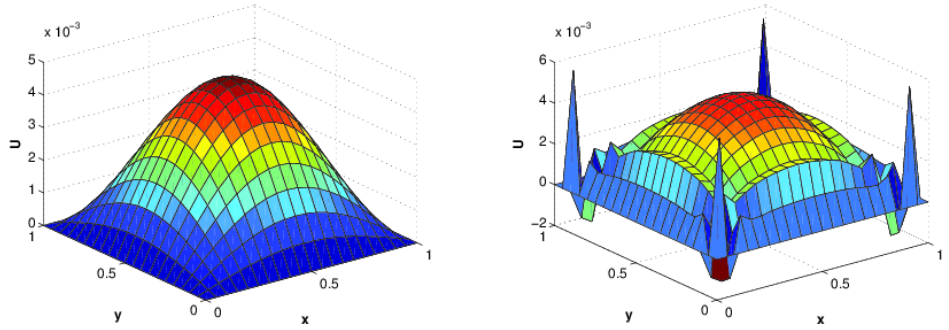


(a) (b)

Figure 5.4: Log-log plots showing efficiency of ETD-RDP over standard Runge-Kutta and Rosenbrock schemes with $\epsilon = 0.01$ for Allen-Cahn equation.

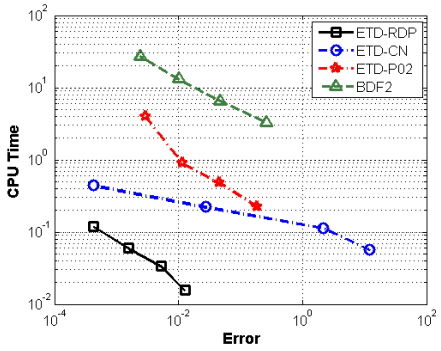
The second order accuracy of ETD-RDP for this problem is seen in Table 5.3. Examing the damping effect of the scheme, it is clear from Figure 5.5 (a-b) that its performance is superior to ETDCN which fails to recover the solution profile. Except on very fine grids, where ETD-CN regains its accuracy, ETD-RDP appears to be the most efficient among ETD-Padé schemes (Figure 5.5 (c-d)). Although the performance of ETD-CN can be improved by the use of initial smoothing steps (Figure 5.7 (a)), ETD-RDP still remains the most efficient scheme across varying diffusion coefficients (Figure 5.7). Table A6 shows ETD-RDP to be 144 times faster than BDF2 for the same level of accuracy. We also compared the performance of ETDRDP to some standard IMEX methods. As expected, IMEX-TR AND IMEX-CNLF, which are not L-stable, fail to recover the solution profile on coarse grids (Figure 5.6 (b-c)). With regards to efficiency, ETD-RDP clearly out performs all the IMEX schemes considered with enhanced computational efficiency when dimensional splitting is employed (Figure 5.6 (d)). The splitting techniques show second order convergence and are

significantly faster than the regular ETD-CN and ETD-RDP scheme when applied to the enzyme kinetics problem (Figure 5.8,Table A8 & A10).



(a) ETD-RDP

(b) ETD-CN



(c) Efficiency

Figure 5.5: (a-b) Solution of the 2D enzyme kinetics after 1 sec of simulation using time step of 0.05 and with diffusion coefficient d=0.25. (c) Log-log efficiency plot for ETD-Padé schemes. (d) Log-log plot showing convergence of ETD-Padé schemes.

5.4 The Brusselator System

Finally, we investigate the computational efficiency of the ETD-RDP scheme for a system of coupled reaction-diffusion equations using the Brusselator model from [38]:

$$u_t = \epsilon_1 \Delta u + A + u^2 v - (B+1)u,$$

$$v_t = \epsilon_2 \Delta v + Bu - u^2 v,$$

k	h	Error	Order	Time
0.1000	0.0250	1.6811×10^{-4}	-	0.05242
0.0500	0.0250	7.1119×10^{-5}	1.24	0.10457
0.0250	0.0250	2.1345×10^{-5}	1.74	0.20885
0.0125	0.0125	5.7778×10^{-6}	1.89	0.39058

Table 5.3: Examining order of convergence of ETD-RDP for 2D Enzyme Kinetics Equation after 1 second.

where the diffusion coefficients are $\epsilon_1 = \epsilon_2 = 2 \times 10^{-3}$, and the chemical parameters are $A = 1, B = 3.4$. At the boundary of the domain, Neumann conditions are imposed

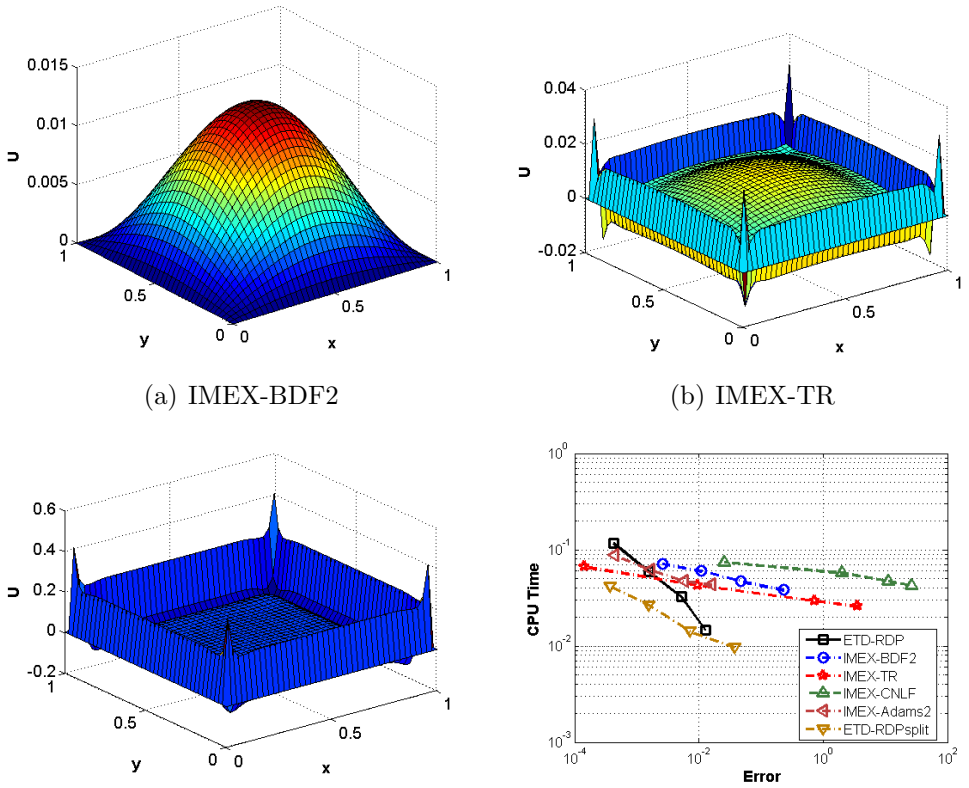
$$\frac{\partial u}{\partial n}|_{\partial\Omega} = \frac{\partial v}{\partial n}|_{\partial\Omega} = 0, \qquad \text{with } \Omega = [0,1]^d$$

The initial conditions are

$$u(x, y, 0) = \frac{1}{2} + y, v(x, y, 0) = 1 + 5x.$$

As expected the scheme is second order convergent (Table 5.4). Notice from Figure 5.9 and Table A7 that ETD-Padé-(0,2) is about 10 times slower than ETD-RDP. We associate the delay in computation to the complex arithmetic performed as a result of complex poles introduced by the Padé-(0,2) rational approximation. Thus for multidimensional, multicomponent reaction-diffusion problems ETD-RDP is preferable to ETD-Padé-(0,2). From Figure 5.9(c,d) we see that while the IMEX methods are more efficient than the regular ETD-RDP, an application of dimensional splitting speeds up the algorithm significantly recovering the dominance of ETD-RDP over IMEX methods. The splitting methods attain the expected second order convergence (Figure 5.10).

On average we observe an 85% reduction in the computational time across all the methods due to splitting (Table A9& A11).



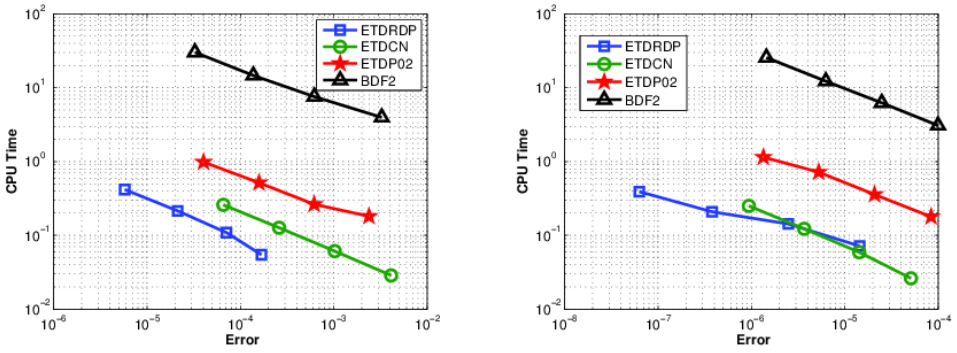
(a) IMEX-BDF2 (b) IMEX-TR

(c) IMEX-CNLF (d) Efficiency

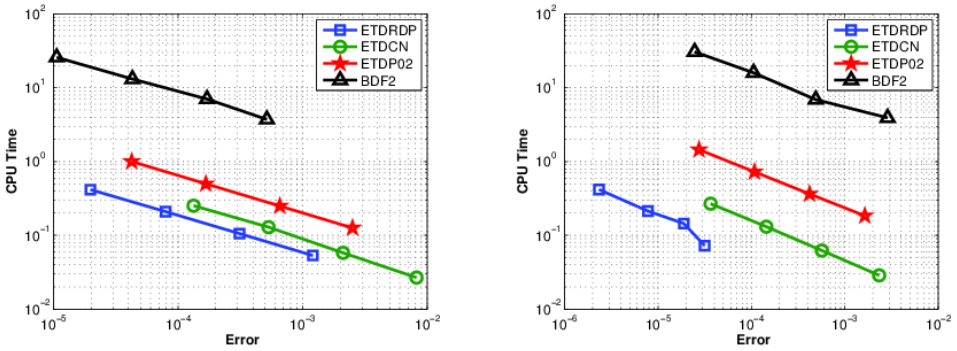
Figure 5.6: (a-c) Solution of the 2D enzyme kinetics after 1 sec of simulation using time step of 0.05 and with diffusion coefficient d=0.2. (d) Log-log efficiency plot comparing ETD-RDP with IMEX schemes.

k	Error	Order	Time
0.1000	2.4996×10^{-1}	-	0.01595
0.0500	4.8185×10^{-2}	2.38	0.02807
0.0250	1.2986×10^{-2}	1.89	0.05611
0.0125	3.3708×10^{-3}	1.95	0.06974

Table 5.4: Order of convergence of ETD-RDP for 2D Brusselator after 2 seconds with $\Delta x = 0.05$.

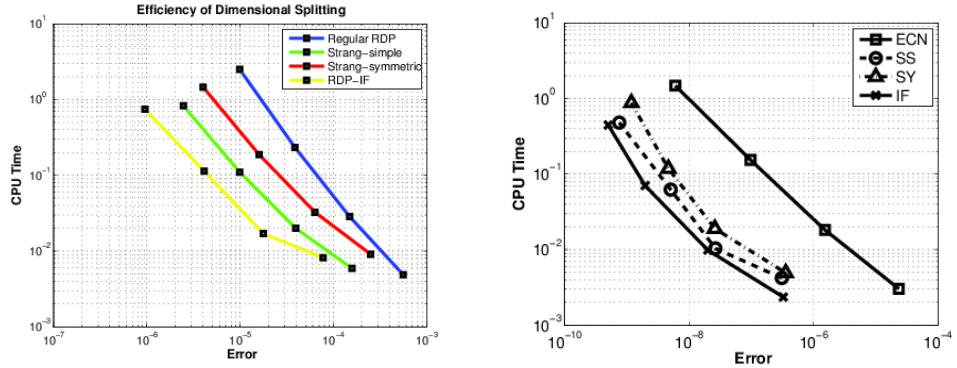


(a) $d_1 = 0.2, d_2 = 0.2$ (b) $d_1 = 0.5, d_2 = 0.5$



(c) $d_1 = 0.1, d_2 = 0.1$ (d) $d_1 = 0.3, d_2 = 0.2$

Figure 5.7: Log-log plots of efficiency of ETD Padé Schemes with varying diffusion coefficients for enzyme kinetics equation.



(a) ETD-RDP (b) ETD-CN

Figure 5.8: log-log efficiency plots comparing dimensional splitting techniques to standard ETD schemes for Enzyme Kinetics Equation.

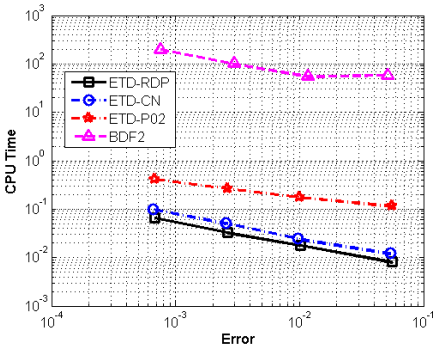
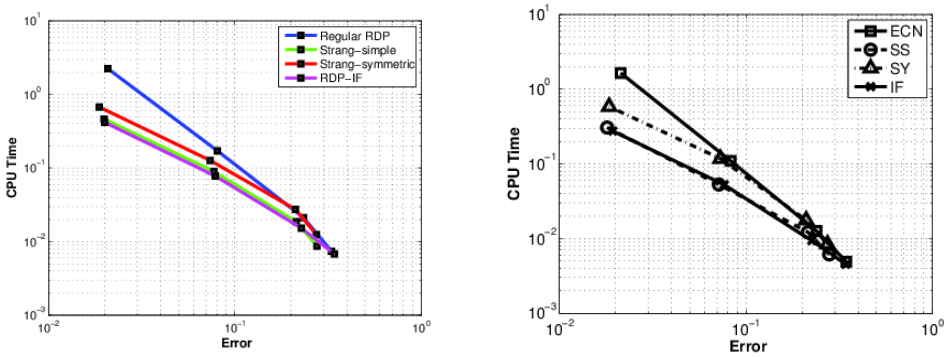


Figure 5.9: Efficiency of ETD Padé schemes for Brusselator equation



(a) ETD-RDP (b) ETD-CN

Figure 5.10: log-log efficiency plots comparing dimensional splitting techniques to standard ETD schemes for Brusselator Equation.

Chapter 6

Conclusion and Recommendation

We have derived a new second-order L-stable ETD Runge-Kutta type scheme which uses a rational approximation with real, distinct poles to approximate the matrix exponentials. We proved the second order accuracy and find that it has a stability region similar to estabilished ETD schemes as well as some well known implicit-explicit schemes. To improve the computational speed when simulating multidimensional problems we have introduced dimensional splitting techniques, currently developed for finite difference discretizations but can be extended to other spatial discretization schemes, which can be appplied to problems with both Dirichlet and Neumann boundary conditions.

Examing the performance of the scheme when applied to stiff reaction diffusion equations, we find it to be computationally faster than most implicit and diagonally implicit Runge-Kutta schemes, with the same level of accuracy as SDIRK scheme. For problems with non-smooth data, we have shown ETD-RDP to be a better alternative to IMEX-CNLF and IMEX-TR schemes which have very poor damping properties. For most of the test problems, we found the performance of RDP to be comparable to IMEX-BDF2 and IMEX-Adams schemes without dimensional splitting and computationally more efficient when splitting was applied. Compared to standard ETD Padé schemes, ETD-RDP has better damping properties than ETD-CN and is computationally more efficient than ETD-Padé02 scheme.

By comparing the standard ETD-RDP and ETD-CN scheme on several test problems, we demonstrated empirically that the dimentional splitting techniques are second order accurate. The proposed splitting methods significantly reduce the computational time required in evolving reaction diffusion systems, in some cases up to 80%.

We recommend future efforts to explore the performance of the scheme for problems with significant advection terms and evaluate the performance of the dimensional splitting techniques to 3D problems with parallelization.

Bibliography

[1] G. Akrivis, M. Crouzeiz, and C. Makridakis, *Implicit-explicit multistep methods for quasilinear parabolic equations*, Numer. Math **82** (1999), 521–541.

[2] A.R.A Anderson, M.A.J Chaplain, E.L. Newman, R.J.C. Steele, and A.M. Thompson, *Mathematical modeling of tumour invasion and metastasis*, J.Theor.Med. **2** (2000), 129–154.

[3] U.M. Ascher, S.J. Ruuth, and B. Wetton, *Implicit-explicit methods for time dependent pde's*, SIAM J. Numer. Anal **32** (1995), 797–823.

[4] H.P. Bhatt and A.Q.M. Khaliq, *The locally extrapolated exponential time differencing lod scheme for multidimensional reaction-diffusion systems*, Journal of Computational and Applied Mathematics **285** (2015), 256–278.

[5] G. Birkoff and R.S. Varga, *Discretization errors for well-set cauchy problems:i*, J. Math. and Phys. **44** (1965), 1–23.

[6] M.A.J Chaplain and A.M. Stuart, *A model mechanism for the chemotactic response of endothelial cells to tumour angiogenesis factor*, IMA J. Math. Appl. Med. Biol. **10** (1993), 149–168.

[7] L.Q. Chen and J. Shen, *Applications of semi-implicit fourier-spectral method to phase field equations*, Comput. Physics Comm **108** (1998), 147–158.

[8] S.M. Cox and P.C. Matthews, *Exponential time differencing for stiff systems*, Journal of Computational Physics **176** (2002), 430–455.

[9] M. Crouziex, *Une mé methode multipas implicite-explicite pour l'approximation des équations d'é volution paraboliques*, Numer. Math. **35** (1980), 257–276.

[10] C.F. Curtis and J.O. Hirchfelder, *Integration of stiff equations.*, Proc. Nat. Acad. Sci. U.S.A **38** (1952), 33–53.

[11] Q. Du and W. Zhu, *Analysis and applications of the exponential time differencing schemes and their contour integration modifications*, BIT Numer Math **45** (2005), 307–328.

[12] B.L. Ehle, *On padé apprximations to the exponential function and a-stable methods for the numerical solution of initial value problems*, Research Rep. CSRR 2010, Dept. AACS, University of Waterloo. (1969).

[13] R.A. Gatenby and E.T. Gawlinski, *A reaction-diffusion model of cancer invasion*, Cancer Research **56** (1996), 5745–5753.

[14] W. Gear and I. Kevrekidis, *Projective methods for stiff differential equations: Problems with gaps in thier eigevalue spectrum*, SIAM Journal on Scientific Computing **24** (2003), 1091–1106.

[15] W. Hairer and G. Wanner, *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer-Verlag Berlin, 1991.

[16] M. Hochbruck and A. Ostermann., *Explicit exponential runge-kutta methods for semilinear parabolic problems*, SIAM J Numer Anal **43** (2005), 1069–1090.

[17] W. Hundsdorfer and J.G. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations*, Springer-Verlag Berlin, 2003.

[18] A.K. Kassam and L.N. Trefethen, *Fourth-order time stepping for stiff pdes*, SIAM J. Sci. Comput **26** (2005), 1214–1233.

[19] A.Q.M. Khaliq, J. Martin-Vanquero, B.A. Wade, and M. Yousuf, *Smoothing schemes for reaction-diffusion systems with nonsmooth data*, Journal of Computational and Applied Mathematics **223** (2009), 374–386.

[20] A.Q.M. Khaliq and B.A. Wade, *On smoothing of the crank-nicolson scheme for nonhomogeneous parabolic problems*, J. Comput. Meths. in Sci. & Eng. **1** (2001), no. 1, 107–123.

[21] B. Kleefeld, A.Q.M. Khaliq, and B.A. Wade, *An etd crank-nicolson method for reaction-diffusion systems*, Numerical Methods for Partial Differential Equations **28** (2012), 1309–1335.

[22] S. Kondo and T. Miura, *Reaction-diffusion model as a framework for understanding biological pattern formation*, Science **329** (2010), 1616.

[23] Randall J. Leveque, *Finite difference methods for ordinary and partial differential equations*, Society for Industrial and Applied Mathematics (2007).

[24] J. Martín-Vaquero and B.A. Wade, *On efficient numerical methods for an initial-boundary value problem with nonlocal boundary conditions*, Appl. Math. Modelling **36** (2012), no. 8, 3411–3418.

[25] C. Moler and C.V. Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, Siam Review **45** (2003), no. 1, 3–49.

[26] C. Moler and C.Van. Loan, *Nineteen dubious ways to compute the exponential of a matrix*, Siam Review **20** (1978), 801–826.

[27] Syvert P. Norsett and Arne Wolfbrandt, *Attainable order of rational approximations to the exponential function with only real poles*, BIT **17** (1977), no. 2, 200–208.

[28] Padé, *Sur la représentation approchée d'une fonction par des fractions rationnelles.*, Ann. Ec, Norm. Sup **9** (1892), 3–93.

[29] J.A. Sherratt and J.D. Murray, *Models of epidermal wound healing*, Proc. R. Soc. Lond.B **241** (1990), 29–36.

[30] V. Thormee, *Galerkin finite element methods for parabolic problem*, Springer series in computational mathematics **25** (1997).

[31] R. Tyson, S.R. Lubkin, and J.D. Murray, *Model and analysis of chemotactic bacterial patterns in a liquid medium*, J. Math.Biol **38** (1999), 359–375.

[32] R.S. Varga, *On higher order stable implicit methods for solving parabolic partial differential equations*, J. Math. and Phys. **40** (1961), 220–231.

[33] J.G. Verwer, W.H. Hundsdorfer, and J.G. Blom, *Numerical time integration for air pollution models*, Surveys on Mathematics for Industry **10** (2002), 107–174.

[34] J. Vigo-Aguiar and B.A. Wade, *Adapted bdf algorithms applied to parabolic problems*, Numerical Methods for Partial Differential Equations **23** (2007), no. 2, 350–365.

[35] D.A. Voss and A.Q.M. Khaliq, *Parallel lod methods for second order time dependent pdes*, Computers Math. Applic. **30** (1994), no. 10, 25–35.

[36] R.A. Willoughby, *Stiff differential systems*, Plenum Press, Newyork (1974).

[37] M. Yousuf, A. Q. M. Khaliq, and B. Kleefeld, *The numerical approximation of nonlinear black–scholes model for exotic path–dependent american options with transaction cost*, International Journal of Computer Mathematics **89** (2012), no. 9, 1239–1254.

[38] P.A. Zegeling and H.P. Kok, *Adaptive moving mesh computations for reaction-diffusion systems*, Journal of Computational and Applied Mathematics **168** (2004), 519–528.

[39] S. Zheng, *Nonlinear evolution equations*, Chapman & Hall/CRC monographs and surveys in pure and applied mathematics (2004).

[40] Z. Zlater, *Computer treatment of large air pollution models*, Kluwer, Dordrecht (1995).

Appendix

A: Convergence Tables for Numerical Experiments

$k = h$	Error_{RDP}	Cpu Time	Error_{ECN}	Cpu Time
0.050000	5.9920×10^{-4}	0.00099	3.1693×10^{-4}	0.00071
0.025000	1.7084×10^{-4}	0.00702	8.6607×10^{-5}	0.00141
0.012500	4.6054×10^{-5}	0.00456	2.2672×10^{-5}	0.00331
0.006250	1.1973×10^{-5}	0.01131	5.7943×10^{-6}	0.00850
0.003125	3.0499×10^{-6}	0.03141	1.4630×10^{-6}	0.02421
k	Error_{P02}	Cpu Time	Error_{BDF2}	Cpu Time
0.050000	8.5674×10^{-4}	0.00109	7.5726×10^{-5}	0.01453
0.025000	2.2341×10^{-4}	0.00261	1.7672×10^{-5}	0.04919
0.012500	5.6679×10^{-5}	0.00731	4.4568×10^{-6}	0.17714
0.006250	1.4239×10^{-5}	0.02294	1.1195×10^{-6}	0.66461
0.003125	3.5654×10^{-6}	0.07836	2.8056×10^{-7}	2.63503

Table A1: Comparing ETD-RDP with other second-order ETD Padé schemes and BDF2 for the model semilinear problem. RDP, ECN and P02 refer to ETD-RDP, ETD-CN and ETD-P(0,2) respectively.

$k = h$	Error_{BE}	Cpu Time	Error_{RCN}	Cpu Time
0.050000	2.1410×10^{-3}	0.01358	2.1081×10^{-5}	0.01024
0.025000	1.0780×10^{-3}	0.02974	4.9197×10^{-6}	0.04612
0.012500	5.4105×10^{-4}	0.09713	1.1319×10^{-6}	0.18378
0.006250	2.7106×10^{-4}	0.35765	2.8294×10^{-7}	0.66128
0.003125	1.3567×10^{-4}	1.38511	7.0734×10^{-8}	2.68551
k	Error_{SDIRK}	Cpu Time	Error_{LBA}	Cpu Time
0.050000	9.8300×10^{-4}	0.02508	1.7512×10^{-5}	0.01524
0.025000	2.3908×10^{-4}	0.06071	4.4776×10^{-6}	0.04757
0.012500	5.8107×10^{-5}	0.19610	1.1319×10^{-6}	0.18360
0.006250	1.4152×10^{-5}	0.87389	2.8294×10^{-7}	0.67271
0.003125	3.4595×10^{-6}	4.06289	7.0734×10^{-8}	2.67684

Table A2: Computational time and accuracy grid refinement studies for backward Euler(BE), Regular Crank-Nicolson(RCN), SDIRK and the LabattoIIIA scheme on the model semilinear problem.

$k = h$	Error_{IMBD2}	Cpu Time	Error_{IMTR}	Cpu Time
0.050000	2.1843×10^{-3}	0.00077	6.7575×10^{-3}	0.00159
0.025000	5.5907×10^{-4}	0.00131	1.7097×10^{-3}	0.00301
0.012500	1.4148×10^{-4}	0.00335	4.2934×10^{-4}	0.00637
0.006250	3.5589×10^{-5}	0.00737	1.0754×10^{-4}	0.01420
0.003125	8.9249×10^{-6}	0.01996	2.6906×10^{-5}	0.03356

Table A3: Computational time and accuracy grid refinement studies for IMEX-BDF2(IMBD2) and IMEX-TR(IMTR), schemes generated for the model semilinear problem.

ϵ	k=h	Error_{RDP}	Cpu Time	Error_{ETDCN}	Cpu Time	Error_{P02}	Cpu Time	Error_{BDF2}	Cpu Time
0.01	0.0500	3.7970×10^{-3}	0.00473	3.8205×10^{-3}	0.00349	3.7514×10^{-3}	0.00472	3.9457×10^{-3}	0.005665
	0.0250	9.9318×10^{-4}	0.01053	1.0013×10^{-3}	0.00798	9.8318×10^{-4}	0.01132	9.9701×10^{-4}	0.19231
	0.0125	2.4696×10^{-4}	0.02615	2.4901×10^{-4}	0.02051	2.4443×10^{-4}	0.03042	2.4745×10^{-4}	0.0.71256
1	0.0500	6.0022×10^{-5}	0.00474	1.2559×10^{-4}	0.00350	6.0020×10^{-5}	0.00470	4.4565×10^{-5}	0.02903
	0.0250	1.5035×10^{-5}	0.01052	2.9445×10^{-5}	0.00809	1.5035×10^{-5}	0.01126	8.9798×10^{-6}	0.11315
	0.0125	3.7585×10^{-6}	0.02653	7.1363×10^{-6}	0.0204	3.7584×10^{-6}	0.03026	2.4025×10^{-6}	0.46971
5	0.0500	1.2019×10^{-5}	0.00473	3.0970×10^{-3}	0.00351	1.2019×10^{-5}	0.00468	9.0928×10^{-6}	0.01801
	0.0250	3.0079×10^{-6}	0.01051	7.4828×10^{-4}	0.00796	3.0079×10^{-6}	0.01121	2.8182×10^{-6}	0.04285
	0.0125	7.5206×10^{-7}	0.02613	1.8553×10^{-4}	0.02046	7.5206×10^{-7}	0.03000	9.5933×10^{-7}	0.10947
10	0.0500	6.0027×10^{-6}	0.00474	7.1057×10^{-3}	0.00352	6.0027×10^{-6}	0.00469	3.9457×10^{-6}	0.01487
	0.0250	1.5039×10^{-6}	0.01047	1.6784×10^{-3}	0.00797	1.5039×10^{-6}	0.01114	2.0139×10^{-6}	0.03816
	0.0125	3.7598×10^{-7}	0.02619	4.1419×10^{-4}	0.02045	3.7598×10^{-7}	0.03009	5.5832×10^{-7}	0.09154

Table A4: Comparing ETD-RDP with other second-order ETD Padé schemes for the Allen-Cahn equation. ETD-CN means ETD Crank-Nicolson and P02 the ETD-Padé-(0,2) scheme.

k=h	Error_{RDP}	Cpu Time	Error_{ROB2}	Cpu Time	Error_{IRKLA}	Cpu Time	Error_{SDIRK}	Cpu Time
0.1	1.5444×10^{-2}	0.00210	1.0598×10^{-2}	0.01887	1.0969×10^{-2}	0.01752	1.1024×10^{-2}	0.03610
0.05	3.8226×10^{-3}	0.00415	2.4902×10^{-3}	0.07339	2.5898×10^{-3}	0.05553	2.6019×10^{-3}	0.11239
0.025	9.9303×10^{-4}	0.00941	6.0354×10^{-4}	0.32931	6.2979×10^{-4}	0.17095	6.3275×10^{-4}	0.35611
0.0125	2.4649×10^{-4}	0.02336	1.4914×10^{-4}	1.56419	1.5586×10^{-4}	0.63742	1.5659×10^{-4}	1.25255

Table A5: Comparing ETD-RDP with other second-order Runge-Kutta and Rosenbrock schemes.

k	Error_{RDP}	Cpu Time	Error_{ETDCN}	Cpu Time
0.1000	1.269×10^{-2}	0.05242	1.2213×10^{1}	0.02713
0.0500	5.2843×10^{-3}	0.10457	2.1694×10^{0}	0.06069
0.0250	1.5809×10^{-3}	0.20885	2.7427×10^{-2}	0.12881
0.0125	4.3114×10^{-4}	0.39058	4.2045×10^{-4}	0.26287
k	Error_{P02}	Cpu Time	Error_{BDF2}	Cpu Time
0.1000	1.8762×10^{-1}	0.22728	2.6071×10^{-1}	4.01074
0.0500	4.6224×10^{-2}	0.48037	4.6874×10^{-2}	7.94775
0.0250	1.1755×10^{-2}	0.90339	1.6258×10^{-2}	12.74277
0.0125	2.977×10^{-3}	3.95377	2.4235×10^{-3}	28.96520

Table A6: Comparing ETD-RDP with other second order ETD Padé schemes and BDF2 for the enzyme kinetics equation with $d = 0.2$ on a fixed grid with $\Delta x = 0.025$. ETDCN refers to ETD Crank-Nicolson Scheme with 4 steps of backward Euler smoothing.

k	Error_{RDP}	Cpu Time	Error_{ECN}	Cpu Time
0.1000	2.4996×10^{-1}	0.01595	2.5191×10^{-1}	0.00664
0.0500	4.8185×10^{-2}	0.02807	5.5150×10^{-2}	0.01134
0.0250	1.2986×10^{-2}	0.05611	1.4554×10^{-2}	0.02266
0.0125	3.3708×10^{-3}	0.06974	3.7325×10^{-3}	0.04502
k	Error_{P02}	Cpu Time	Error_{BDF2}	Cpu Time
0.1000	2.6209×10^{-1}	0.08602	2.4655×10^{-1}	59.1874
0.0500	4.5198×10^{-2}	1.2221	6.6237×10^{-2}	55.4519
0.0250	1.2066×10^{-2}	0.17521	1.7065×10^{-2}	101.3988
0.0125	3.1222×10^{-3}	0.26159	$\times10^{-4}$	200.2900

Table A7: Comparing ETD-RDP with other second-order ETD Padé schemes and BDF2 for the Brusselator Model after 2 secs with $\Delta x = 0.05$.

$k = h$	Error_{RDP}	Cpu Time	Rate	Error_{SS}	Cpu Time	Order
0.1000	5.6031×10^{-4}	0.00485	-	1.5740×10^{-4}	0.00591	-
0.0500	1.4888×10^{-4}	0.02846	1.91	3.9406×10^{-5}	0.01989	2.00
0.0250	3.8636×10^{-5}	0.23314	1.95	9.8522×10^{-6}	0.11013	2.00
0.0125	9.8825×10^{-6}	2.49755	1.97	2.4612×10^{-6}	0.82556	2.00
$k = h$	Error_{SY}	Cpu Time	Rate	Error_{IF}	Cpu Time	Order
0.1000	2.4955×10^{-4}	0.00906	-	7.7449×10^{-5}	0.00817	-
0.0500	6.3224×10^{-5}	0.03255	1.98	1.7726×10^{-5}	0.01698	2.13
0.0250	1.5931×10^{-5}	0.18793	1.99	4.0984×10^{-6}	0.11310	2.11
0.0125	4.0012×10^{-6}	1.45993	1.99	9.5445×10^{-7}	0.74388	2.10

Table A8: Convergence of ETD-RDP splitting methods for enzyme kinetics. RDP refers to ETD-RDP scheme, SS the Strang simple splitting, SY the Strang symmetric and IF the integrating factor method.

$k = h$	Error_{RDP}	Cpu Time	Rate	Error_{SS}	Cpu Time	Order
0.1000	3.2981×10^{-1}	0.0074	-	2.7596×10^{-1}	0.00870	-
0.0500	2.3382×10^{-1}	0.02109	0.50	2.1380×10^{-1}	0.01859	0.37
0.0250	8.0448×10^{-2}	0.17208	1.54	7.7244×10^{-2}	0.08954	1.47
0.0125	2.0789×10^{-2}	2.24926	1.95	1.9850×10^{-2}	0.46392	1.96
$k = h$	Error_{SY}	Cpu Time	Rate	Error_{IF}	Cpu Time	Order
0.1000	2.7382×10^{-1}	0.01242	-	3.4223×10^{-1}	0.00683	-
0.0500	2.1151×10^{-1}	0.02739	0.37	2.2771×10^{-1}	0.01520	0.59
0.0250	7.3634×10^{-2}	0.12726	1.52	7.8393×10^{-2}	0.07821	1.54
0.0125	1.8725×10^{-2}	0.67638	1.98	2.0022×10^{-2}	0.41840	1.97

Table A9: Convergence of ETD-RDP splitting methods for enzyme kinetics.

$k = h$	Error_{ECN}	Cpu Time	Rate	Error_{SS}	Cpu Time	Order
0.1000	2.9447×10^{-4}	0.00383	-	2.001×10^{-4}	0.00280	-
0.0500	7.3406×10^{-5}	0.01829	2.00	4.9598×10^{-5}	0.00988	2.01
0.0250	1.8283×10^{-5}	0.15366	2.01	1.2372×10^{-5}	0.06054	2.00
0.0125	4.5594×10^{-6}	2.28658	2.00	3.0914×10^{-6}	0.44592	2.00
$k = h$	Error_{SY}	Cpu Time	Rate	Error_{IF}	Cpu Time	Order
0.1000	2.6199×10^{-4}	0.00422	-	2.1989×10^{-4}	0.00204	-
0.0500	6.4963×10^{-5}	0.01641	2.01	5.4457×10^{-5}	0.00854	2.01
0.0250	1.6204×10^{-5}	0.09897	2.00	1.3581×10^{-5}	0.0552	2.00
0.0125	4.0483×10^{-6}	0.77917	2.00	3.3929×10^{-6}	0.43117	2.00

Table A10: Convergence of ETD-CN splitting methods for enzyme kinetics.

$k=h$	Error_{ECN}	Cpu Time	Rate	Error_{SS}	Cpu Time	Order
0.1000	3.4490×10^{-1}	0.00373	-	2.7937×10^{-1}	0.01203	-
0.0500	2.3918×10^{-1}	0.01281	0.53	2.1463×10^{-1}	0.01203	0.38
0.0250	8.2716×10^{-2}	0.10076	1.53	7.1545×10^{-2}	0.05287	1.58
0.0125	2.1302×10^{-2}	1.46030	1.96	1.8012×10^{-2}	0.29253	1.99
$k=h$	Error_{SY}	Cpu Time	Rate	Error_{IF}	Cpu Time	Order
0.1000	$2.7352\ \times10^{-1}$	0.00670	-	3.4380×10^{-1}	0.00270	-
0.0500	2.1028×10^{-1}	0.01802	0.38	2.2550×10^{-1}	0.00806	0.61
0.0250	7.2922×10^{-2}	0.07947	1.53	7.6429×10^{-2}	0.03883	1.56
0.0125	1.8468×10^{-2}	0.43406	1.98	1.9321×10^{-2}	0.22363	1.98

Table A11: Convergence of ETD-CN splitting methods for brusselator after 2 secs.

Numerical Method	Butcher-array	Scheme
Backward Euler	$\begin{array}{c\|c} 1 & 1 \\ \hline & 1 \end{array}$	$u_{n+1} = u_n + k\mathcal{G}(t_{n+1}, u_{n+1})$
Crank Nicolson/ Labatto IIIA [15]	$\begin{array}{c\|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$	$u_{n+1} = u_n + \frac{k}{2}\mathcal{G}(t_n, u_n) + \frac{k}{2}\mathcal{G}(t_{n+1}, u_{n+1})$
Labatto IIIB [15]	$\begin{array}{c\|cc} 0 & \frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$	$u_{n1} = u_n + \frac{k}{2}\mathcal{G}(t_{n1}, u_{n1})$ $u_{n2} = u_n + \frac{k}{2}(\mathcal{G}(t_{n1}, u_{n1})$ $u_{n+1} = u_n + \frac{k}{2}(\mathcal{G}(t_{n1}, u_{n1}) + \mathcal{G}(t_{n2}, u_{n2})$
SDIRK [17]	$\begin{array}{c\|cc} \gamma & \gamma & 0 \\ 1-\gamma & 1-2\gamma & \gamma \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$	$u_{n1} = u_n + k\gamma\mathcal{G}(t_{n1}, u_{n1})$ $u_{n2} = u_n + k((1-2\gamma)\mathcal{G}(t_{n1}, u_{n1}) + \gamma\mathcal{G}(t_{n2}, u_{n2}))$ $u_{n+1} = u_n + \frac{k}{2}(\mathcal{G}(t_{n1}, u_{n1}) + \mathcal{G}(t_{n2}, u_{n2}))$

Table A12: Some important first and second order Runge-Kutta schemes with $\gamma = 1 - \frac{\sqrt{2}}{2}$ for SDIRK scheme.

[k/j]	0	1	2
0	$\frac{1}{1}$	$\frac{1+z}{1}$	$\frac{1+z+\frac{z^2}{2!}}{1}$
1	$\frac{1}{1-z}$	$\frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}$	$\frac{1+\frac{2}{3}z+\frac{1}{3}\frac{z^2}{2!}}{1-\frac{1}{3}z}$
2	$\frac{1}{1-z+\frac{z^2}{2!}}$	$\frac{1+\frac{1}{3}z}{1-\frac{2}{3}z+\frac{1}{3}\frac{z^2}{2!}}$	$\frac{1+\frac{1}{2}z+\frac{1}{6}\frac{z^2}{2!}}{1-\frac{1}{2}z+\frac{1}{6}\frac{z^2}{2!}}$
3	$\frac{1}{1-z+\frac{z^2}{2!}-\frac{z^3}{3!}}$	$\frac{1+\frac{1}{4}z}{1-\frac{3}{4}z+\frac{1}{2}\frac{z^2}{2!}-\frac{1}{4}\frac{z^3}{3!}}$	$\frac{1+\frac{2}{5}z+\frac{1}{10}\frac{z^2}{2!}}{1-\frac{3}{5}z+\frac{3}{10}\frac{z^2}{2!}-\frac{1}{10}\frac{z^3}{3!}}$

Table A13: Summary of some common Padé approximations

Curriculum Vitae

Emmanuel Asante-Asamani

Education

2010–Present: M.S & Ph.D., Industrial Mathematics, University of Wisconsin-Milwaukee(UWM)

2005–2009: B.S., Mathematics, University of Science and Technology, Ghana.

Research Experience

June 2014–June 2015: Research Assistant, Center for Industrial Mathematics, UWM

Mathematical Modeling of pigment and filler settling in coating formulations.

June 2012–Aug 2013: Research Assistant, Biomedical Visualization Lab, UWM

Numerical simulation of calcium signaling in cardiac ventricular myocytes.

May 2011–June 2012: Research Assistant, Public Health Informatics and Genomics Lab, UWM, Learning causal relationships in big clinical data using Bayesian Networks.

June 2009–Sep 2009: Intern, Brain Imaging Center, Cardiff University

Development of computational software for running magnetoencyphalograpy experiments.

Publications

(2016) E.O. Asante-Asamani, Lei Wang and Zeyun Yu

A cylindrical basis function for solving partial differential equations on manifolds. Procedia Computer Science, Volume 80, 2016, Pages 233 & 244. International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA

(2015) E.O. Asante-Asamani, Abdul Q.M. Khaliq and Bruce A. Wade; A real distinct poles exponential time differencing scheme for reaction-diffusion equations. Journal of Computational and Applied Mathematics.http://dx.doi.org/10.1016/j.cam.2015.09.017

(2015) E.O. Asante-Asamani and Bruce A. Wade; A Dimensional Splitting of ETD Schemes for Reaction-Diffusion Systems. Comm. in Comput. Phys. (CICP)

Talks/Posters

Talks

(2015) A real distinct poles exponential time differencing scheme for nonlinear advection-diffusion-reaction systems. Joint Mathematics Meeting.

(2014) A cylindrical basis function for solving partial differential equations on manifolds,SIAM Annual Meeting.

(2013) An efficient methodology for learning Bayesian Networks, Center for Biomedical Informatics, Harvard Medical School.

Posters

(2015) A meshfree method for numerical simulation of calcium dynamics in ventricular myocytes, Siam Conference on Computational Science and Engineering

(2013) An optimal Bayesian Network modeling approach in breast cancer risk prediction, New England Science Symposium, Harvard Medical School.

(2012) A comparative assessment of greedy equivalence search and PC algorithm for learning Bayesian Networks in complex data sets, Computational Science Symposium, Marquette University.

Teaching Experience

Associate Lecturer, UW-Waukesha

College Algebra with ALEKS, MATH 110(Fall 2015)

Prepared and presented all lectures, administered final grades.

Instructor, UWM

Calculus, MATH 231(Summer 2011-Spring 2012); Trigonometry, MATH 117 (Fall 2012-Spring 2013); Intermediate Algebra with ALEKS, MATH 105(Spring 2014);

Beginning Algebra with ALEKS, MATH 094(Fall 2015); Differential Equations and Linear Algebra (Math 234).

Prepared and presented all lectures, administered final grades.

Teaching Assistant, UWM

Business Calculus, MATH 211(Fall 2010-Spring 2011); Beginning Algebra with ALEKS, MATH 095 (Fall 2013)

Held discussion sessions, graded all homework and exams.

Awards

Research Excellence Award, UWM, Spring 2016

Vice chancellors Graduate Student Award (Fall 2010-Spring 2016)

Professional Membership

Society for Industrial and Applied Mathematics

American Mathematical Society

Computer skills

Programming Languages: PYTHON, FORTRAN

Mathematical Software: MATLAB, Mathematica, Maple, R

Languages

English, French

Interests

Tennis, Swimming