Linearly Stabilized Schemes for the Time Integration of Stiff Nonlinear PDEs

by

Kevin Chow

B.Math., University of Waterloo, 2014

Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Mathematics
Faculty of Science

SIMON FRASER UNIVERSITY
Fall 2016

Approval

Name:	**Kevin Chow**
Degree:	**Master of Science (Mathematics)**
Title:	***Linearly Stabilized Schemes for the Time Integration of Stiff Nonlinear PDEs***
Examining Committee:	**Chair:** Dr. Weiran Sun Assistant Professor

Dr. Steven Ruuth
Senior Supervisor
Professor

Dr. Robert Russell
Professor Emerius

Dr. Ben Adcock
Assistant Professor

Date Defended: 9 December 2016

Abstract

This is a blank document from which you can start writing your thesis.

linearization

stabilized explicit RK (RKC, ROCK, DUMKA)

Often, we see new, efficient solvers that are so complex, that the preferred method of choice remains those that are simple, clear, and easy to implement.

Keywords: Thesis template; Simon Fraser University; time travel paradoxes

Dedication

Acknowledgements

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we propose and analyze some new linearly stabilized schemes for the time integration of stiff nonlinear PDEs. A linearly stabilized scheme of first order has been used in a number of areas, with the first known example being from a paper by Douglas and Dupont [3] where they use this technique for the solution to a variable coefficient heat equation on rectangular domains. In subsequent years, the idea has been rediscovered by others [5, 20] and has found applications to gradient systems, Hele-Shaw flow, interface motion, image processing, and solving PDEs on surfaces [6, 15, 7, 16, 13].

In each of the references mentioned in the previous paragraph, the authors have succeeded in implemented only a first order time stepping method. Recently in [4], Duchemin and Eggers consolidated the approach and produced a second order linearly stabilized scheme they refer to as the explicit-implicit-null (EIN) method. The procedure they propose works off the first order scheme and extrapolates to second order. Moreover, they identified that the key principle for the success of any linearly stabilized scheme is unconditional stability. Indeed, a significant section of their paper is devoted to showing that their method is unconditionally stable under only a mild condition on a parameter that is introduced.

Our derivations for new linearly stabilized schemes also begins by ensuring that the newly derived schemes are in fact unconditionally stable. The techniques we employ in our stability analysis are very much those of a standard linear stability analysis and are reviewed first. A light discussion of order of accuracy and Richardson extrapolation is also included before an overview of the thesis is given.

1.1 Linear Stability Analysis

1.1.1 Stability and the scalar test equation

Linear stability analysis is predicated finding the constraints necessary on the time step-size so that the numerical solution generated using that particular time stepping method applied to the test equation,

$$u' = \lambda u, \quad \lambda < 0, \tag{1.1}$$

has properties resembling that of the exact solution to the test equation,

$$u(t^n + \Delta t) = e^{\lambda \Delta t} u(t^n). \tag{1.2}$$

Observe that, in general, the exact solution satisfies

$$\frac{\left|u(t^n + \Delta t)\right|}{\left|u(t^n)\right|} = \left|e^{\lambda \Delta t}\right| < 1, \quad \text{for all } \lambda \Delta t < 0. \tag{1.3}$$

The analogous property for numerical methods is what we will refer to as stability.

For example, applying the forward Euler method to the test equation (1.1), we get

$$\frac{u^{n+1} - u^n}{\Delta t} = \lambda u^n \iff u^{n+1} = \underbrace{(1 + \lambda \Delta t)}_{=\xi_{FE}} u^n, \tag{1.4}$$

where u^n is an approximation to $u(t^n)$. Then imposing $|\xi_{FE}| < 1$, we get

$$|1 + \lambda \Delta t| < 1 \iff -2 < \lambda \Delta t < 0, \tag{1.5}$$

and so for stability, $\Delta t < 2/\left|\lambda\right|$ must be satisfied. As the time step-size, Δt, is constrained, we say forward Euler is conditionally stable.

As another example, we may apply backward Euler to the test equation. Doing so we get

$$\frac{u^{n+1} - u^n}{\Delta t} = \lambda u^{n+1} \iff u^{n+1} = \underbrace{\frac{1}{1 - \lambda \Delta t}}_{=\xi_{BE}} u^n. \tag{1.6}$$

This time, imposing $|\xi_{BE}| < 1$ adds no new constraint to the time step-size. When no additional constraints are imposed on the time step-size, we say the numerical method is unconditionally stable.

More generally, to determine the stability constraint of any one step method, one applies said method to the test equation, rearranges as $u^{n+1} = \xi(\lambda \Delta t) u^n$, and imposes $\left|\xi(\lambda \Delta t)\right| < 1$. (The quantity $\xi(\lambda \Delta t)$ is commonly referred to as the amplification factor, and the region, $\{\lambda \Delta t \in \mathbb{C} \mid \left|\xi(\lambda \Delta t)\right| < 1\}$, the stability region.) In other words, for stability, we require that the magnitude of the amplification factor is less than one.

1.1.2 Stability contours

A stability contour plot is a graphical device for understanding the stability constraint of a method. It offers a way for us to verify calculations done analytically, or to visualize the stability region of a numerical method where an analytic solution is infeasible. Stability contours plots in this thesis all show contours of the amplification factor $\xi(\lambda\Delta t)$ plotted over a subset of the region $\{\lambda\Delta t \in \mathbb{C} \mid \mathrm{Im}(\lambda\Delta t) \geq 0\}$, with a focus on the left half plane and the negative real line. Fig. 1.1 shows stability contours of the forward Euler and the backward Euler method. Note that the regions are symmetric with respect to the real axis and thus only $\mathrm{Im}(\lambda\Delta t) \geq 0$ will be plotted.

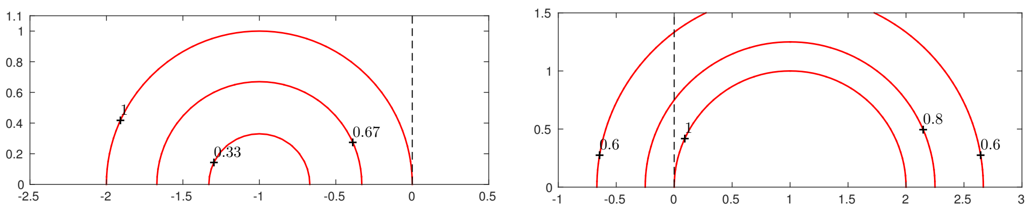

Figure 1.1: On the left is the stability contour plot for forward Euler. The stability region is the interior of the 1-contour. On the right is the stability contour plot for backward Euler. The stabilty region is the region outside the 1-contour.

1.1.3 Relation to stiff PDEs

Recall that our motivation is to develop methods suited to the time integration of stiff nonlinear PDEs. So how does the time step restriction of a numerical method derived from application to the test equation relate to time step selection for a stiff nonlinear PDE? The relation is as follows. Suppose the PDE has been discretized in space and we are to advance the solution of the resulting large system of ODEs, $u' = F(u)$, by one time step, i.e. advance the numerical solution u^n to u^{n+1}. Over just one time step, it may be reasonable to consider the linearization,

$$u' = \bar{u} + \frac{\partial F}{\partial u}\bigg|_{u=\bar{u}} (u - \bar{u}),$$

or setting $v = u - \bar{u}$, $v' = Av$. Further assuming that A is diagonalizable, $A = T^{-1}DT$, where $D = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$, we get

$$v' = T^{-1}DTv \iff (Tv)' = D(Tv) \tag{1.7}$$

$$\iff w_k' = \lambda w_k, \quad k = 1, \ldots, N. \tag{1.8}$$

In other words, under appropriate conditions, it may be fair to analyze the dynamics of the nonlinear system by inspecting the eigenvalues of the Jacobian from its linearized state.

Thus to time step, we require that the computation be stable for each eigenmode. The time step constraint will then be dictated by the largest absolute eigenvalue.

For instance, suppose we found, from a linearized system of ODEs, the eigenvalues to be $2(\cos(k\Delta x) - 1)/\Delta x^2$, $k = 1, \ldots, N$. Then the largest absolute eigenvalue can be bounded as

$$\left| \frac{2}{\Delta x^2} (\cos(k\Delta x) - 1) \right| \leq \frac{4}{\Delta x^2}, \tag{1.9}$$

and stable time step-sizes for forward Euler must then satisfy $\Delta t < \Delta x^2/4$. On the other hand, unconditionally stable methods such as backward Euler, maintain stability irrespective of the grid size Δx.

That unconditionally stable methods maintain stability irrespective of the grid size is crucial for the solution to stiff problems. In Chapter 5, we will solve problems in 2D and 3D where the largest absolute eigenvalues scale like $\mathcal{O}(h^2)$ and $\mathcal{O}(h^4)$, where h is the spatial grid size. In those cases, a conditionally stable method requiring $\Delta t = \mathcal{O}(h^k)$, $k \geq 2$, would give unnecessarily fine temporal resolution, and more critically, either push the cost of the computation to absurd levels, or else be forced to resolve poorly in space.

1.2 Order of Accuracy

If two competing numerical methods, both consuming similar levels of resources (e.g. CPU time, memory,) but one gave more accurate approximations, then likely one would be deemed superior to the other.

For a time stepping method applied to the initial value problem

$$\begin{cases} u' = F(u), & 0 \leq t \leq T, \\ u(0) = u_0, \end{cases} \tag{1.10}$$

with step-size Δt, we will say that the method is convergent of order k (or kth order accurate) if the global error behaves as

$$\|u^n - u(t^n)\| = \mathcal{O}(\Delta t^k), \quad \text{as} \quad \Delta t \to 0. \tag{1.11}$$

This points to a preference for high order accurate method. Intuitively, in order to achieve some desired level of accuracy, a low order method will require a finer time step-size than a method of higher order accuracy. Then if each time step had comparable costs, the higher order method will require less resources overall to compute the solution.

Finally, we note that familiar methods such as forward Euler and backward Euler are first order accurate.

1.2.1 Richardson extrapolation

As one of the methods under consideration relies heavily on Richardson extrapolation, we include here a brief note on this technique. A comprehensive text discussing the validity of this technique can be found in [17].

Suppose we are approximating some quantity, q, via the rule $\bar{q}$ that is dependent on the step-size h and that the error is of the form

$$q - \bar{q}(h) = C_1 h^k + C_2 h^{k+1} + \cdots . \tag{1.12}$$

Observe that if we evaluate both $\bar{q}(h)$ and $\bar{q}(h/2)$, then we can eliminate the leading order error term,

$$q - q^*(h) = q - \frac{2^k \bar{q}(h/2) - \bar{q}(h)}{2^k - 1} = C h^{k+1} + \cdots . \tag{1.13}$$

1.3 Overview

In Chapter 2, we formally introduce the notion of linear stabilization. Movitation for this technique is supplied by the need to handled a 1D stiff nonlinear PDE describing axisymmetric mean curvature flow and leads us to the well-known first order linearly stabilized time integrator and the EIN method of Duchemin and Eggers. Following that, the framework in which we analyze the stability of linearly stabilized schemes is set.

In Chapter 3 we investigate implicit-explicit (IMEX) linear multistep methods within the linear stabilization framework. A detailed comparison of the schemes based on IMEX multistep schemes and the EIN method is conducted. Our experiments suggest criteria in addition to unconditional stability are necessary for practical linearly stabilized schemes. This in turn eliminates third order and higher multistep based linearly stabilized schemes from use.

In Chapter 4, we explore the use of exponential Runge-Kutta methods to mend this deficiency. A second order and a fourth order exponential Runge-Kutta method are verified to exhibit parameter restrictions of the right form. However, other complications limits their use.

In Chapter 5, application of our linearly stabilized schemes to a number of 2D and 3D problems are presented. Not surprisingly, our second order schemes offer massive improvements over the commonly used first order linearly stabilized scheme. The experiments show that our schemes provide substantial efficiency improvement yet can be implemented with tremendous ease.

Finally, some concluding remarks are presented in Chapter 6.

Chapter 2

Adding Zero, Unconditional Stability and a Modified Test Equation

To time step for stiff, nonlinear problems, we set out two key design principles. Firstly, we want to handle the nonlinearity simply and inexpensively. Secondly, we must be free to select time step-sizes reflecting the accuracy requirement, and not step-sizes that are primarily constrained by stability. Linearly stabilized schemes, as we will see, adhere to both principles and are remarkably easy to implement.

2.1 Prototype 1D Problem

As a prototype, let us consider the following 1D mean curvature motion problem [4]:

$$u_t = \frac{u_{xx}}{1+u_x^2} - \frac{1}{u}, \qquad 0 < x < 10,\, t > 0, \tag{2.1a}$$

with initial and boundary condition

$$u(x,0) = 1 + 0.10 \sin\left(\frac{\pi}{5}x\right) \tag{2.1b}$$

$$u(0,t) = u(10,t) = 1. \tag{2.1c}$$

A time evolution of this problem is plotted in Fig. 2.1.

The presence of the u_{xx} guarantees that (2.1) is stiff, suggesting that an implicit time stepping scheme would prove more efficient. However, having to additionally handle the factor of $(1+u_x^2)^{-1}$ would suggest otherwise. Thus we are presented with a scenario where neither an implicit nor an explicit approach proves particularly palatable.

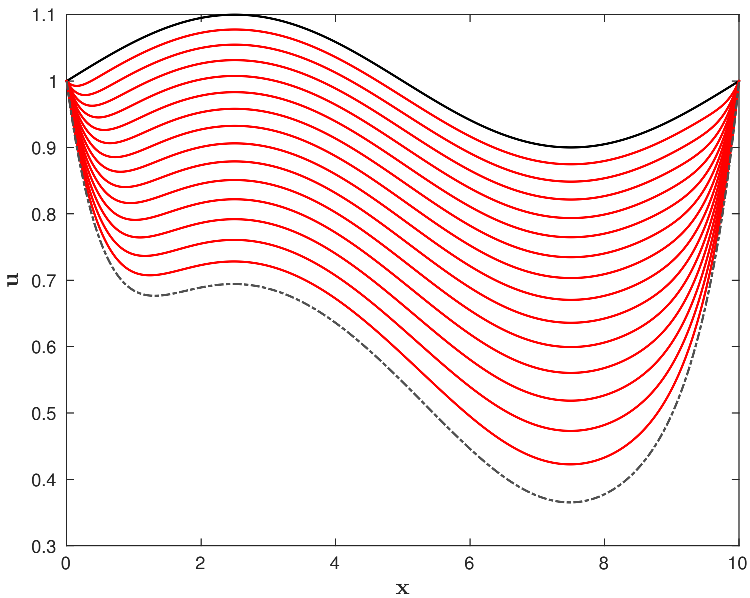

Figure 2.1: Time evolution of the solution to (2.1). The black curve is the initial state. The bottom gray dashed curve is the state at $T = 0.35$.

2.1.1 A first order linearly stabilized scheme

As demonstrated in Duchemin and Eggers [4] as well as an earlier paper by Smereka [20], an efficient method to handling (2.1) is to add and subtract a linear Laplacian term to the righthand side,

$$u_t = \underbrace{\frac{u_{xx}}{1+u_x^2} - \frac{1}{u} - u_{xx}}_{\mathcal{N}(u)} + \underbrace{\quad u_{xx} \quad}_{\mathcal{L}u}, \tag{2.2}$$

and then time step as

$$\frac{u^{n+1} - u^n}{\Delta t} = \mathcal{N}(u^n) + \mathcal{L}u^{n+1}. \tag{2.3}$$

Since this is our first instance witnessing a linearly stabilized scheme in action, we remark on some of the key properties. We first note that in the continuous case, the modified equation (2.2) is unchanged from (2.1a). Next, note in the discrete case (2.3), the nonlinear term is evaluated explicitly, and ignoring the $\mathcal{L}u^{n+1}$ term, it is a forward Euler step. Over on the linear term, the implicit solve in this time stepping procedure is a backward Euler step. This method of time stepping is known well as implicit-explicit (IMEX) or semi-implicit Euler [1, 20]. As it is a combination of explicit and implicit Euler steps, the accuracy is first order. We also note that the simplicity of the implicit term means that the related linear algebra is efficient and easy to implement. Lastly, as a result of the implicit solution to the $\mathcal{L}u$ term, we may expect this scheme to have improved stability compared to a purely explicit scheme, and indeed this is the case. Adopting second order centred differences in space, we will show next that this scheme is unconditionally stable.

2.1.2 A von Neumann stability analysis

With the prescribed spatial-temporal discretization, we have at the interior nodes,

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} = 4\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{4\Delta x^2+(u_{j+1}^n-u_{j-1}^n)^2} - \frac{1}{u_j^n} - \frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{\Delta x^2} + \frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{\Delta x^2}. \tag{2.4}$$

The von Neumann stability analysis then proceeds by writing the numerical solution as the exact solution to the difference equation (2.4) perturbed by a single Fourier mode,

$$u_j^n = \bar{u}(j\Delta x, n\Delta t) + \xi^n e^{ikj\Delta x} = \bar{u}_j^n + \xi^n e^{ikj\Delta x}. \tag{2.5}$$

Recording the result term-by term, we have

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} = \frac{\bar{u}_j^{n+1}-\bar{u}_j^n}{\Delta t} + \frac{(\xi-1)\xi^n e^{ikj\Delta t}}{\Delta t}, \tag{2.6a}$$

$$\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{\Delta x^2} = \frac{\bar{u}_{j+1}^{n+1}-2\bar{u}_j^{n+1}+\bar{u}_{j-1}^{n+1}}{\Delta x^2} + \frac{2}{\Delta x^2}\left(\cos(k\Delta x)-1\right)\xi^{n+1}e^{ikj\Delta x}, \tag{2.6b}$$

$$\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{\Delta x^2} = \frac{\bar{u}_{j+1}^n-2\bar{u}_j^n+\bar{u}_{j-1}^n}{\Delta x^2} + \frac{2}{\Delta x^2}\left(\cos(k\Delta x)-1\right)\xi^n e^{ikj\Delta x}, \tag{2.6c}$$

$$\frac{1}{u_j^n} = \frac{1}{\bar{u}_j^n+\xi^n e^{ikj\Delta x}} \approx \frac{1}{\bar{u}_j^n} - \xi^n e^{ikj\Delta x}\frac{1}{(\bar{u}_j^n)^2}, \tag{2.6d}$$

and

$$\begin{aligned}
4\frac{u_{j+1}^n-2u_j^n+u_{j-1}^n}{4\Delta x^2+(u_{j+1}^n-u_{j-1}^n)^2} &= 4\frac{\bar{u}_{j+1}^n-2\bar{u}_j^n+\bar{u}_{j-1}^n+2(\cos(k\Delta x)-1)\xi^n e^{ikj\Delta x}}{4\Delta x^2+[\bar{u}_{j+1}^n-\bar{u}_{j-1}^n-2i\sin(k\Delta x)\xi^n e^{ikj\Delta x}]^2} \\
&= 4\frac{2(\cos(k\Delta x)-1)[\frac{\bar{u}_{j+1}^n-2\bar{u}_j^n+\bar{u}_{j-1}^n}{2(\cos(k\Delta x)-1)}+\xi^n e^{ikj\Delta x}]}{4\Delta x^2+(2i\sin(k\Delta x))^2[\frac{\bar{u}_{j+1}^n-\bar{u}_{j-1}^n}{2i\sin(k\Delta x)}-\xi^n e^{ikj\Delta x}]^2} \\
&\approx 4\frac{\bar{u}_{j+1}^n-2\bar{u}_j^n+\bar{u}_{j-1}^n}{4\Delta x^2+(\bar{u}_{j+1}^n-\bar{u}_{j-1}^n)^2} \\
&\quad + 8(\cos(k\Delta x)-1)\xi^n e^{ikj\Delta x}\frac{1}{4\Delta x^2+(\bar{u}_{j+1}^n-\bar{u}_{j-1}^n)^2} \\
&\quad - 8i\sin(k\Delta x)\xi^n e^{ikj\Delta x}(\bar{u}_{j+1}^n-\bar{u}_{j-1}^n)(\bar{u}_{j+1}^n-2\bar{u}_j^n+\bar{u}_{j-1}^n) \\
&\approx 4\frac{\bar{u}_{j+1}^n-2\bar{u}_j^n+\bar{u}_{j-1}^n}{4\Delta x^2+(\bar{u}_{j+1}^n-\bar{u}_{j-1}^n)^2} \\
&\quad + \frac{2}{\Delta x^2}(\cos(k\Delta x)-1)\xi^n e^{ikj\Delta x}\frac{1}{1+(D_1\bar{u}_j^n)^2},
\end{aligned} \tag{2.6e}$$

where $D_1\bar{u}_j^n = (\bar{u}_{j+1}^n - \bar{u}_{j-1}^n)/(2\Delta x)$. Combining, (2.6) simplifies to

$$\frac{\xi - 1}{\Delta t} = \frac{2}{\Delta x^2}\frac{\cos(k\Delta x) - 1}{1 + (D_1\bar{u}_j^n)^2} + \frac{1}{(\bar{u}_j^n)^2} + \frac{2}{\Delta x^2}(\cos(k\Delta x) - 1)(\xi - 1), \tag{2.7}$$

from which we can then isolate the amplification factor,

$$\xi = 1 + \underbrace{\Delta t \frac{\frac{2}{\Delta x^2}\frac{\cos(k\Delta x)-1}{1+(D_1\bar{u}_j^n)^2} + \frac{1}{(\bar{u}_j^n)^2}}{1 - \frac{2\Delta t}{\Delta x^2}(\cos(k\Delta x) - 1)}}_{=w}. \tag{2.8}$$

In the next steps, we will show $|\xi| < 1$ for all $\Delta t > 0$, i.e. absolute stability is unconditional. We show the equivalent statement $-2 < w < 0$. First, $w < 0$. As the denominator is positive, $w > 0$ will hold true so long as the numerator is negative,

$$\frac{2}{\Delta x^2}\frac{\cos(k\Delta x) - 1}{1 + (D_1\bar{u}_j^n)^2} + \frac{1}{(\bar{u}_j^n)^2} < 0 \iff \left(\frac{\Delta x}{\bar{u}_j^n}\right)^2 < \frac{2(1 - \cos(k\Delta x))}{1 + (D_1\bar{u}_j^n)^2}. \tag{2.9}$$

This last relation is satisfied on the assumption that $\Delta x \ll \bar{u}_j^n$. Next, we examine the numerator of $w + 2$,

$$\begin{aligned} &\frac{2\Delta t}{\Delta x^2}\frac{\cos(k\Delta x) - 1}{1 + (D_1\bar{u}_j^n)^2} + \frac{\Delta t}{(\bar{u}_j^n)^2} + 2 - 4\frac{\Delta t}{\Delta x^2}(\cos(k\Delta x) - 1) \\ &\quad = 2\frac{\Delta t}{\Delta x^2}(\cos(k\Delta x) - 1)\left(\frac{1}{1 + (D_1\bar{u}_j^n)^2} - 2\right) + 2 + \frac{\Delta t}{(\bar{u}_j^n)^2}. \end{aligned} \tag{2.10}$$

Since each term is positive without restriction, we have that $w + 2 > 0$, and thus $|\xi| < 1$ for all $\Delta t > 0$.

2.1.3 Second order by Richardson extrapolation

As stated at the outset, the time stepping procedure in (2.3) is only first order. The work of Duchemin and Eggers [4] (and as was suggested but not implemented in [20],) extends the method to second order by Richardson extrapolation. Moreover, they generalized the approach with a free parameter, p, i.e.,

$$u_t = \frac{u_{xx}}{1 + u_x^2} - \frac{1}{u} - pu_{xx} + pu_{xx}, \tag{2.11}$$

and derived restrictions to p on the condition that the resulting scheme be unconditionally stable. With the semi-implicit Euler approach, they found $p \geq 0.5/(1 + (D_1\bar{u}_j^n)^2)$ to be sufficient. With the additional Richardson extrapolation, the restriction is $p \geq (2/3)/(1 + (D_1\bar{u}_j^n)^2)$.

2.2 A Modified Test Equation

This method of linear stabilization would be extremely cumbersome if in each case we had to perform a von Neumann analysis to determine the stability. Thankfully, there is an alternative. To begin, let us now consider the more general problem, $u' = \mathcal{N}(u)$, which may be from a spatial discretization of some nonlinear PDE, and we assume $\mathcal{N}(u)$ is a stiff, nonlinear term. We can modify, in a way analogous to (2.2), by subtracting and adding a linear term that "resembles" $\mathcal{N}(u)$,

$$u_t = \underbrace{(\mathcal{N}(u) - p\mathcal{L}u)}_{\text{nonlinear}} + \underbrace{p\mathcal{L}u}_{\text{linear}}, \tag{2.12}$$

and demand unconditionally stable time stepping. This last request we now address.

To progress, we abandon (2.12) and instead examine a related, but simplified scenario that we will refer to as the modified test equation:

$$u' = (1-p)\lambda u + p\lambda u, \quad \text{where} \quad \lambda < 0,\, p > 0. \tag{2.13}$$

In (2.13), the quantity λ captures the character of $\mathcal{N}$ (e.g. the eigenvalues of the Jacobian of the linearized $\mathcal{N}$), and the quantity $p\lambda u$ represents the linear component that closely resembles $\mathcal{N}(u)$. Note also when $p = 0$, the modified test equation reduces to the standard test equation, $u' = \lambda u$.

We will discuss next the stability properties of three time stepping methods as applied to the modified test equation (2.13).

2.2.1 Forward Euler

Forward Euler is a first order time stepping method that treats the righthand side explicitly. Application to (2.13) is therefore no different than to the standard test equation. Thus there is no hope of unconditional stability.

2.2.2 Linearly stabilized semi-implicit Euler

Semi-implicit Euler time stepping was demonstrated on the 1D mean curvature motion problem (2.1) via (2.2) and (2.3), and its stability further analyzed. In the case of the modified test equation, we identify $\mathcal{N}(u^n) = (1-p)\lambda u^n$ and $\mathcal{L}u^{n+1} = p\lambda u^{n+1}$, to get

$$\frac{u^{n+1} - u^n}{\Delta t} = (1-p)\lambda u^n + p\lambda u^{n+1} \iff u^{n+1} = \underbrace{\left(1 + \frac{\lambda \Delta t}{1 - p\lambda\Delta t}\right)}_{=\xi_E} u^n. \tag{2.14}$$

Enforcing absolute stability, i.e. $|\xi_E| < 1$, for all $\lambda\Delta t < 0$, we find

$$|\xi_E| < 1 \iff -2 < \frac{\lambda\Delta t}{1 - p\lambda\Delta t} < 0 \iff p > 0 \quad \text{and} \quad (2p-1)\lambda\Delta t < 2. \tag{2.15}$$

Thus unconditional stability is guaranteed if $p \geq 1/2$.

We will onwards be referring to this scheme as SBDF1.

2.2.3 Explicit-implicit null

In [4], Duchemin and Eggers extended the SBDF1 approach to second order by using Richardson extrapolation, and they referred to their methodology as explicit-implicit null (EIN). For their method, the amplification factor, ξ_{EIN}, can be expressed in terms of ξ_E,

$$\xi_{EIN} = 2\xi_E^2(\Delta t/2) - \xi_E(\Delta t) = 1 + \underbrace{\frac{z\left(p(3p-2)z^2 + 2(1-4p)z + 4\right)}{(1-pz)(2-pz)^2}}_{=w}, \tag{2.16}$$

where $z = \lambda\Delta t$. Similar to before, we enforce $|\xi_{EIN}| < 1$ for all $z < 0$ to derive a restriction on p. This is equivalent to $-2 < w < 0$. We first observe that since $p > 0$, the denominator of w is positive for all $z < 0$. Thus a necessary condition is $3p - 2 > 0 \iff p > 2/3$, as we require the quadratic in the numerator to be positive for all $z < 0$. Further, the roots of that quadratic are positive whenever they are real,

$$\frac{1}{2p(3p-2)}\left(2(4p-1) \pm 2\sqrt{(4p-1)^2 - 4p(3p-2)}\right) > 0. \tag{2.17}$$

Therefore, $p > 2/3$ is necessary and sufficient for $w < 0$. Next, we show $w + 2 > 0$. The numerator of $w + 2$ can be simplified to

$$-p(2 - 3p + 2p^2)z^3 + (2 - 8p + 10p^2)z^2 + 4(1-4p)z + 8. \tag{2.18}$$

The coefficients of the powers of z have the property

$$\begin{aligned}
&[z] = 4(1-4p) > 0 \quad \text{whenever} \quad p > 1/4,\\
&[z^2] = 2 - 8p + 10p^2 > 0 \quad \text{since the discriminant } 8^2 - 4(2)(10) < 0,\\
&[z^3] = -p(2 - 3p + 2p^2) > 0 \quad \text{since the discriminant } (-3)^2 - 4(2)(2) < 0,
\end{aligned}$$

for all $z < 0$, thus guaranteeing the numerator is positive. And since the denominator, as stated previously, is positive, we are guaranteed $w + 2 > 0$, and thus unconditional stability is guaranteed if $p > 2/3$.

Remark 1. It is perhaps more faithful to write the restrictions as $p\lambda/\lambda > 2/3$ rather than simply $p > 2/3$, as it is necessarily the ratio of the two that must satisfy the restriction, and not the parameter p. This distinction is vital for any problem beyond a simple test equation, where the eigenvalues of the nonlinear operator, $\lambda_{\mathcal{N}}$, and the eigenvalues of the linear operator, $\lambda_{\mathcal{L}}$, may follow the same scaling, e.g. $\lambda_{\mathcal{N}}, \lambda_{\mathcal{L}} = \mathcal{O}(\Delta x^{-2})$, but the actual values may be far apart, e.g. $\lambda_{\mathcal{N}} \approx 100\lambda_{\mathcal{L}}$.

Remark 2. What we now understand is that in order to linearly stabilize effectively, we need the ratio of the eigenvalues to meet a specific bound. This bound, if we assume a fixed and reasonable choice of a linear operator $\mathcal{L}$, however, is specific to the time stepping procedure, and is met by choosing a sufficiently large value of p.

Remark 3. Finally, we must mention that this provides us with a simple avenue to selecting p without a von Neumann analysis, as the latter in many cases may be infeasible. For example, the analysis in (2.6) reveals that

$$\lambda_{\mathcal{N}} = \frac{2}{\Delta x^2}\frac{\cos(k\Delta x) - 1}{1 + (D_1 \bar{u}_j^n)^2} \quad \text{and} \quad \lambda_{\mathcal{L}} = \frac{2}{\Delta x^2}(\cos(k\Delta x) - 1). \tag{2.19}$$

So then to apply, e.g. EIN, to (2.11) with centred differences, we would select p to satisfy

$$\frac{2}{3} < \frac{p\lambda_{\mathcal{L}}}{\lambda_{\mathcal{N}}} = p(1 + (D_1\bar{u}_j^n)^2) \iff p > \frac{2}{3}(1 + (D_1\bar{u}_j^n)^2)^{-1}. \tag{2.20}$$

2.3 Numerical Results

We present in this section some numerical tests to support our claims.

2.3.1 Stability contours

Stability contours plots are shown in Figs. 2.2 and 2.3 as a verification of the earlier analysis of the SBDF1 and the EIN method. In the first case, we found the parameter restriction to be $p \geq 0.5$, and this is in agreement with Fig. 2.2. When we set $p = 0.45$, the stable range along the negative real axis is clearly bounded. But at $p = 0.50$ and greater, it is stability region includes the entirety of the negative real axis.

The same is demonstrated for the EIN method in Fig. 2.3. Again we see that below the derived threshold, the stability region is bounded. But beyond that, the stability region contains the entire negative real axis.

2.3.2 Numerical convergence test

The convergence of SBDF1 and EIN are demonstrated next on (2.1) with (2.2) in place of (2.1a). We solve to time $T = 0.35$ with $N = 2048$ spatial grid nodes. As a reference solution, we use an explicit third order Runge-Kutta method with time step-size $\Delta t = 1.46 \times 10^{-5}$. (Typical explicit methods demand step-sizes comparable for stability.) With the linearly

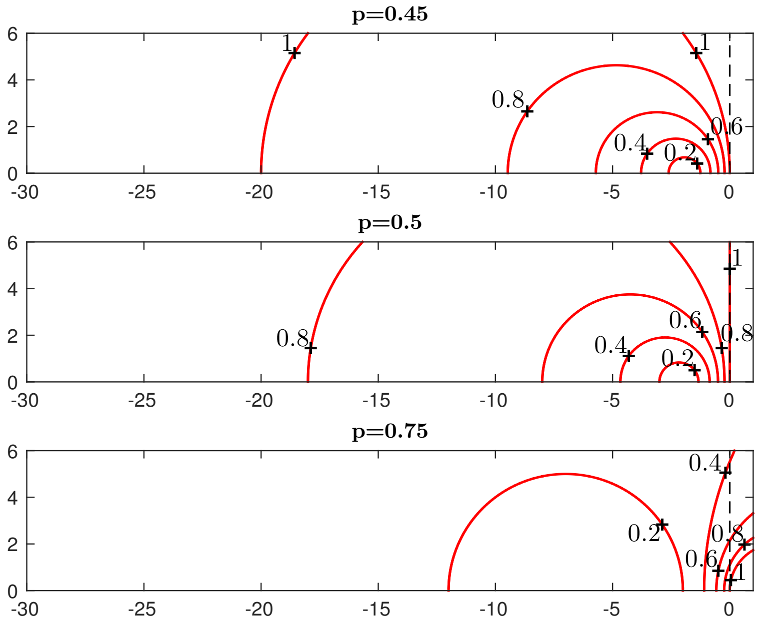

Figure 2.2: Stability contours for SBDF1 at various p.

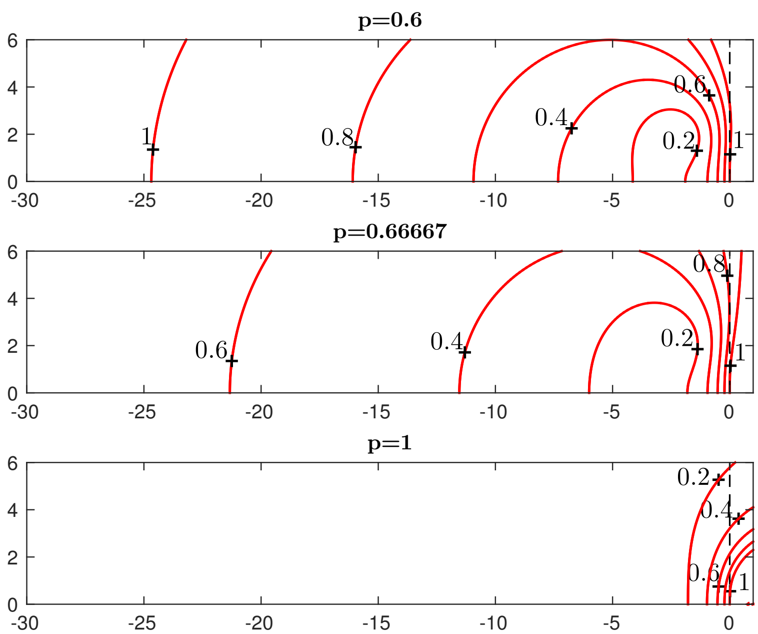

Figure 2.3: Stability contours for EIN at various p.

stabilized schemes, we will solve (2.1) with step-sizes as large as $\Delta t = 2.09 \times 10^{-2}$. We show the max norm relative error.

Results of the numerical convergence study are shown in Fig. 2.4. Both schemes converge with the expected order of accuracy. No issues with stability are observed.

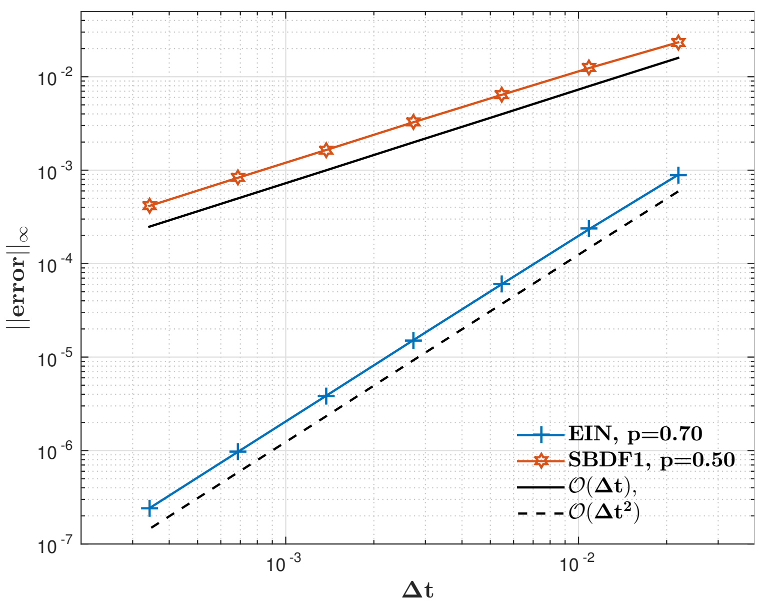

Figure 2.4: Result of numerical convergence study to (2.1) with SBDF1 and EIN. Values $p = 0.50$ and $p = 0.70$ were chosen for SBDF1 and EIN respectively.

Chapter 3

IMEX Linear Multistep Methods

For equations whose righthand side comprise of a stiff linear component and a nonstiff nonlinear part, a popular class of methods to apply are the implicit-explicit linear multistep methods[1]. The simplest of these is the semi-implicit Euler – forward Euler to the nonlinear component and backward Euler to the linear, stiff component – a scheme that we reviewed in Chapter 2.

In this chapter, we investigate the use of IMEX methods within the linear stabilization framework. Implicit solution of the added linear term provides the needed stability, and the remaining terms, including the stiff nonlinear term, are solved explicitly.

3.1 IMEX Formulas

In [1], IMEX schemes up to order four are investigated and a select number are singled out for their extensive use in the literature or for desired properties such as strong high frequency damping. As we are familiar with the first order variant, we begin by listing the second order methods of interest. These, and the higher order variants, are presented as applied to the ODE

$$u' = f + g,$$

where f we identify as the nonlinear/nonstiff component and g the stiff linear component.
Second order methods
CNAB:

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{3}{2} f^n - \frac{1}{2} f^{n-1} + \frac{1}{2}(g^{n+1} + g^n), \tag{3.1}$$

[1]We will refer to these simply as IMEX methods.

mCNAB:

$$\frac{u^{n+1}-u^n}{\Delta t} = \frac{3}{2}f^n - \frac{1}{2}f^{n-1} + \frac{9}{16}g^{n+1} + \frac{3}{8}g^n + \frac{1}{16}g^{n-1}, \tag{3.2}$$

CNLF:

$$\frac{u^{n+1}-u^{n-1}}{2\Delta t} = f^n + \frac{1}{2}(g^{n+1}+g^{n-1}), \tag{3.3}$$

SBDF2:

$$\frac{3u^{n+1}-4u^n+u^{n-1}}{2\Delta t} = 2f^n - f^{n-1} + g^{n+1}. \tag{3.4}$$

Third order methods
SBDF3:

$$\frac{1}{\Delta t}\left(\frac{11}{6}u^{n+1} - 3u^n + \frac{3}{2}u^{n-1} - \frac{1}{3}u^{n-2}\right) = 3f^n - 3f^{n-1} + f^{n-2} + g^{n+1}. \tag{3.5}$$

Fourth order methods
SBDF4:

$$\frac{1}{\Delta t}\left(\frac{25}{12}u^{n+1} - 4u^n + 3u^{n-1} - \frac{4}{3}u^{n-2} + \frac{1}{4}u^{n-3}\right) = 4f^n - 6f^{n-1} + 4f^{n-2} - f^{n-3} + g^{n+1}. \tag{3.6}$$

In the next section, we will apply these IMEX schemes to the modified test equation to determine for each scheme the range of p suitable for linear stabilization.

3.2 Analysis of the Amplification Polynomials

Let us restate here the modified test equation and the basic assumptions we make. The modified test equation is

$$u' = (1-p)\lambda u + p\lambda u, \quad \text{where} \quad \lambda < 0,\, p > 0.$$

The goal is, for each IMEX scheme, to identify the constraint on the parameter p such that when satisfied, we may freely choose the time step-size without being subject to a stability restriction.

3.2.1 Schur and von Neumann polynomials

The resulting polynomials from applying an nth order IMEX scheme to the modified test equation will be a degree n polynomial in the amplification factor, ξ. The tool of choice for analyzing these amplification polynomials is the theory of von Neumann polynomials [21, Chapter 4]. Below we give the relevant definitions and theorems.

Definition 3.1. The polynomial ϕ is a Schur polynomial if all its roots, r_q, satisfy $|r_q| < 1$.

Definition 3.2. The polynomial ϕ is a von Neumann polynomial if all its roots, r_q, satisfy $|r_q| \leq 1$.

Definition 3.3. The polynomial ϕ is a simple von Neumann polynomial is ϕ is a von Neumann polynomial and its roots on the unit circle are simple roots.

Definition 3.4. For any polynomial $\phi(\xi) = \sum_{j=0}^{n} a_j \xi^j$, we define the polynomial ϕ^* by $\phi^*(\xi) = \sum_{j=0}^{n} a_{n-j}^* \xi^j$, where * on the coefficient denotes the complex conjugate.

Definition 3.5. For any polynomial $\phi_n(\xi) = \sum_{j=0}^{n} a_j \xi^j$, we define recursively the polynomial ϕ_{n-1} by

$$\phi_{n-1}(\xi) = \frac{\phi_n^*(0)\phi_n(\xi) - \phi_n(0)\phi_n^*(\xi)}{\xi}. \tag{3.7}$$

Theorem 3.6. *ϕ_n is a simple von Neumann polynomial if and only if either*

(a) $|\phi_n(0)| < |\phi_n^*(0)|$ and ϕ_{n-1} is a simple von Neumann polynomial or

(b) ϕ_{n-1} is identically zero and ϕ_n' is a Schur polynomial.

3.2.2 Amplification polynomials of second order IMEX schemes

We start by applying CNAB (3.1) to the modified test equation (2.13). Combined with the anzatz $u^n = \xi^n$ and setting $z = \lambda \Delta t$, we get the amplification polynomial

$$\Phi_2(\xi) = \left(1 - \frac{1}{2}zp\right)\xi^2 - \left(1 + z\left(\frac{3}{2} - p\right)\right)\xi + \frac{1}{2}z(1-p). \tag{3.8}$$

The next series of steps will show that for all $z < 0$ (i.e. $\lambda < 0$,) (3.8) is a simple von Neumann polynomial if and only if $p \geq 1$. We do so by showing that Theorem 3.6 holds for Φ_2.

We first give Φ_2^* and Φ_1 as defined by the processes in Definitions 3.4 and 3.5,

$$\Phi_2^*(\xi) = \frac{1}{2}z(1-p)\xi^2 - \left(1 + z\left(\frac{3}{2} - p\right)\right)\xi + 1 - \frac{1}{2}zp, \tag{3.9}$$

and

$$\Phi_1(\xi) = \left(1 - zp + z^2\left(\frac{1}{2}p - \frac{1}{4}\right)\right)\xi - 1 - z(1-p) + z^2\left(\frac{3}{4} - \frac{1}{2}p\right). \tag{3.10}$$

Next, we verify that if $p \geq 1$, then $|\Phi_2(0)| < |\Phi_2^*(0)|$. Reformulating the expression as

$$|\Phi_2(0)| < |\Phi_2^*(0)| \iff 0 < (\Phi_2^*(0))^2 - \Phi_2(0)^2 = 1 - zp - \frac{1}{2}z^2\left(\frac{1}{2} - p\right),$$

(and keeping in mind that we ask this inequality to hold only for $z < 0$,) we find that the contribution from each term in the rightmost quadratic is positive, thus verifying the claim. Finally, we show that Φ_1 is simple von Neumann directly. Denoting the root of Φ_1 as ξ_1,

$$|\xi_1| < 1 \iff \left|\frac{(2p-3)z-2}{(2p-1)z-2}\right| < 1 \iff 0 < 8z((p-1)z-1),$$

which holds for all $z < 0$ if and only if $p \geq 1$.

Other IMEX schemes are analyzed in the same way. The amplification polynomials of the second order schemes are recorded for reference in Table 3.1, along with the parameter restriction for which we observe unconditional stability.

Table 3.1: Amplification polynomial of second order IMEX schemes. The rightmost column is the guide for choosing p.

Method	**Amplification Polynomial**	$p\lambda/\lambda \in$
CNAB	$\left(1 - \frac{1}{2}zp\right)\xi^2 - \left(1 + z\left(\frac{3}{2} - p\right)\right)\xi + \frac{1}{2}z(1-p)$	$[1, \infty)$
mCNAB	$\left(1 - \frac{9}{16}zp\right)\xi^2 - \left(1 + z\left(\frac{3}{2} - \frac{9}{8}p\right)\right)\xi + \frac{1}{2}z\left(1 - \frac{9}{8}p\right)$	$[8/9, \infty)$
CNLF	$(1 - pz)\,\xi^2 - 2z(1-p)\xi - (1 + pz)$	$[1/2, \infty)$
SBDF2	$\left(\frac{3}{2} - zp\right)\xi^2 - 2\left(1 + z(1-p)\right)\xi + \frac{1}{2} + z(1-p)$	$[3/4, \infty)$

3.2.3 Amplification polynomials of third and fourth order IMEX LMMs

We continue with the analysis for higher order IMEX schemes. Again, amplification polynomials and parameter restrictions are derived. Because the expressions and manipulations quickly become cumbersome and tedious for higher order methods, the computer algebra system, Maple$^{\text{TM}}$, was used to for the majority of the calculations.

Listed in Table 3.2 are respectively the amplification factor and the parameter restriction for SBDF3 and SBDF4. We must point out a crucial difference. In contrast to the second order methods, the derived parameter restriction leaves only a finite interval. This will be

Table 3.2: Amplification polynomial and choice of parameter when applying high order IMEX LMMs to the modified test equation for unconditional stability.

Method	**Amplification Polynomial**	$p\lambda/\lambda \in$
SBDF3	$\left(\frac{11}{6} - zp\right)\xi^3 - 3\left(1 + z(1-p)\right)\xi^2 + \frac{3}{2}\left(1 + 2z(1-p)\right)\xi - \frac{1}{3}\left(1 + 3z(1-p)\right)$	$[7/8, 2]$
SBDF4	$\left(\frac{25}{12} - zp\right)\xi^4 - 4\left(1 + z(1-p)\right)\xi^3 + 3\left(1 + 2z(1-p)\right)\xi^2 - \frac{4}{3}\left(1 + 3z(1-p)\right)\xi + \frac{1}{4}\left(1 + 4z(1-p)\right)$	$[15/16, 5/4]$

addressed further in the subsequent section where it is demonstrated that this detail renders the linearly stabilized SBDF3 and SBDF4 useless.

3.3 Convergence of Linearly Stabilized IMEX Methods

We present in this section numerical tests to support our claims. Stability contours are shown and a numerical convergence test to (2.1) is conducted. We then provide an answer as to why linearly stabilized SBDF3 and SBDF4 fails.

3.3.1 Stability contours

Presented in Figs. 3.1 to 3.4 are the stability contours for the second order IMEX schemes, (3.1) to (3.4), applied to the modified test equation, (2.13). For each, we provide stability contours with p set at $0.95p_0$, p_0, $1.5p_0$, where p_0 is the minimum required for unconditional stability (Table 3.1, rightmost column).

In Fig. 3.5 are the stability contours for SBDF3 (3.5), and in Fig. 3.6 are the stability contours for SBDF4 (3.6). In each, we plot four instances to capture finite range where unconditional stability is observed.

3.3.2 Numerical convergence test

Convergence of the proposed schemes will be tested on the 1D mean curvature motion problem, which we restate below:

$$u_t = \frac{u_{xx}}{1 + u_x^2} - \frac{1}{u} - pu_{xx} + pu_{xx}, \quad 0 < x < 10, \quad t > 0, \tag{3.11a}$$

with initial and boundary conditions

$$u(x, 0) = 1 + 0.10 \sin\left(\frac{\pi}{5}x\right) \tag{3.11b}$$

$$u(0, t) = u(10, t) = 1. \tag{3.11c}$$

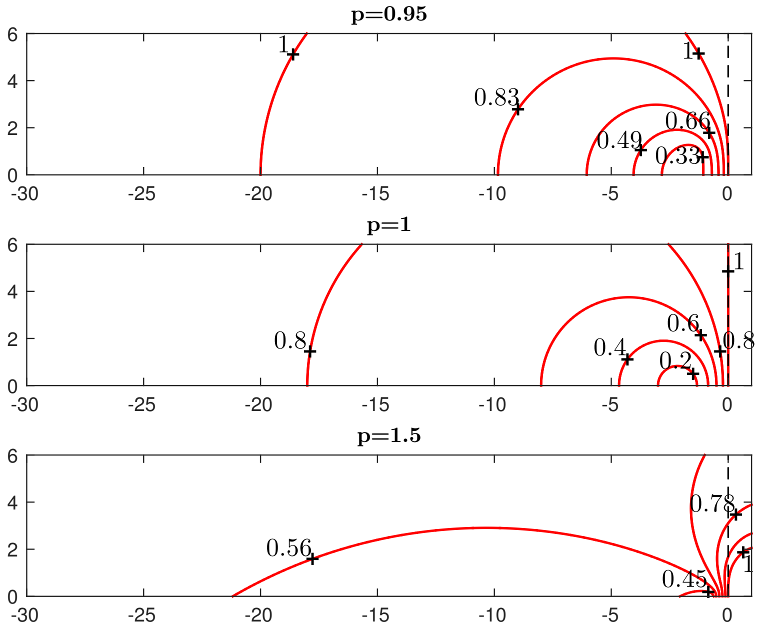

Figure 3.1: Stability contours for CNAB at various p.

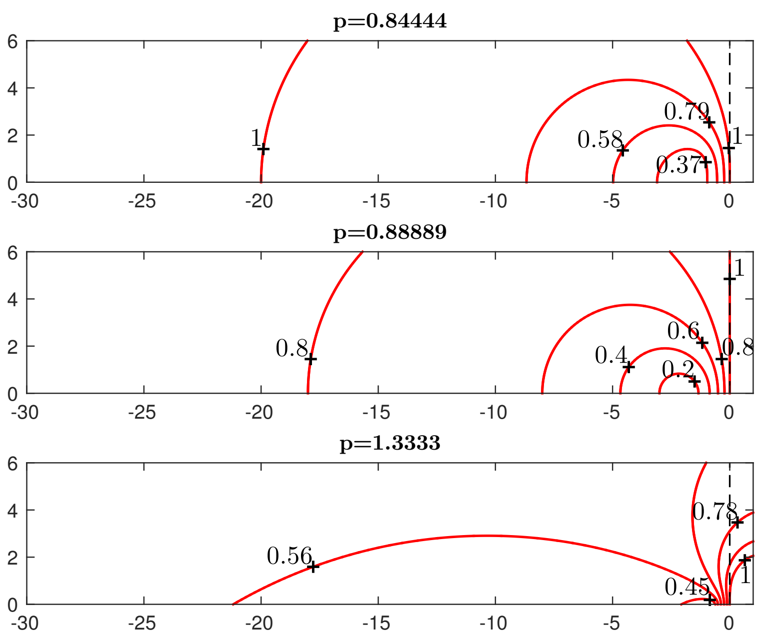

Figure 3.2: Stability contours for mCNAB at various p.

As in Section 2.3.2, we solve to time $T = 0.35$ with $N = 2048$ spatial grid nodes. A reference solution is generated using a third order Runge-Kutta method with time step-size $\Delta t = 1.46 \times 10^{-5}$, and a max norm relative error calculated. Starting values necessary for the multistep methods are found using a third order Runge-Kutta method with small time step. The values of p used for each scheme is as indicated in Fig. 3.7.

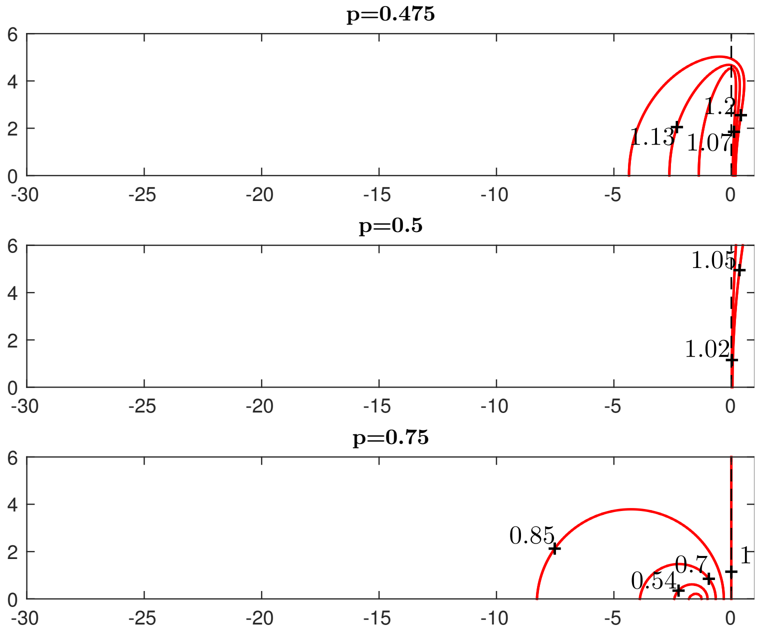

Figure 3.3: Stability contours for CNLF at various p.

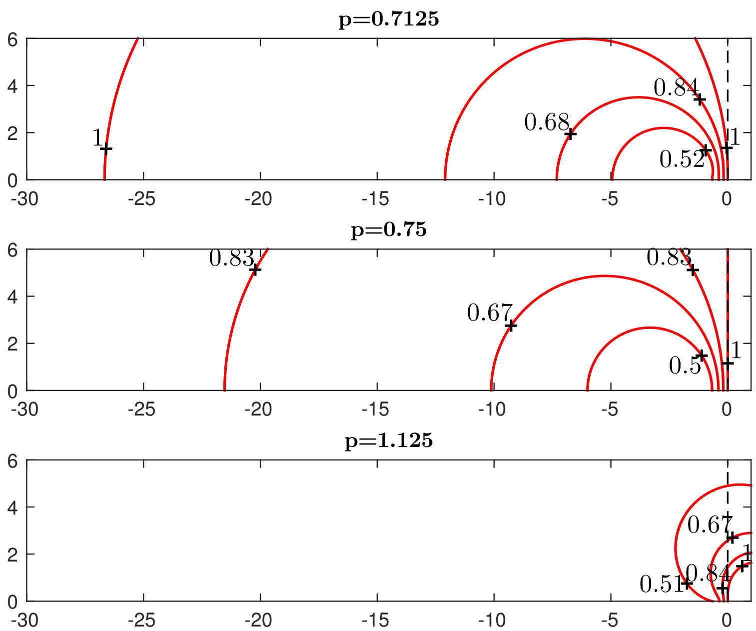

Figure 3.4: Stability contours for SBDF2 at various p.

Results of the numerical convergence study are shown in Fig. 3.7. Each of the second order methods converge with the order of accuracy as expected, with SBDF2 having the largest errors, followed by CNAB/mCNAB (identical performance), CNLF, and then EIN. SBDF3 converges nicely with $p = 0.875$. SBDF4 does not exhibit fourth order convergence and in fact fails at both $\Delta t = 6.84 \times 10^{-4}$ and $\Delta t = 3.42 \times 10^{-4}$. We discuss next why SBDF4 fails, and show also that SBDF3 suffers from the same defect.

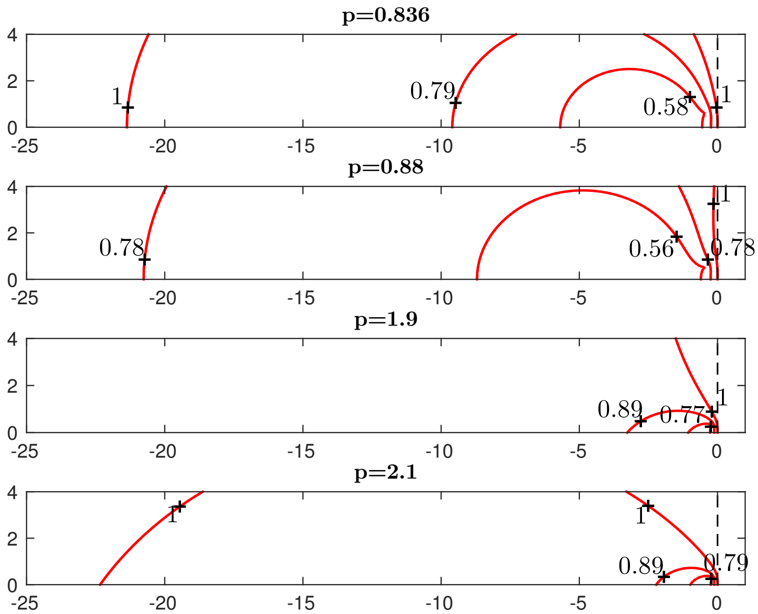

Figure 3.5: Stability contours for SBDF3 at various p.

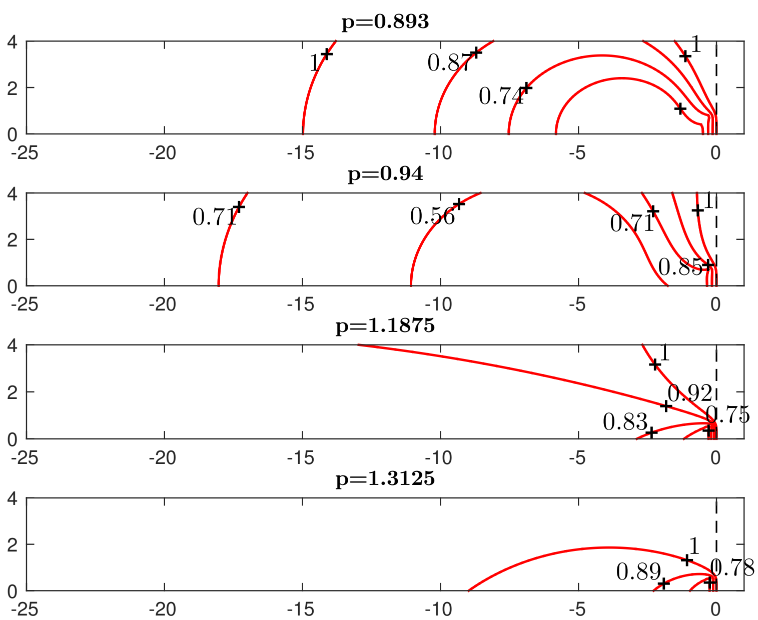

Figure 3.6: Stability contours for SBDF4 at various p.

3.3.3 Failure with high order IMEX methods

First, let us tabulate the observed (non)convergence of SBDF3 at various values of p. In Table 3.3, we document three cases. The first case is the one shown in Fig. 3.7. The second case exhibits a drastic drop in the observed convergence rate and and in the third case the method diverged as the time step-size is reduced. We attribute the divergence of SBDF3 and SBDF4 to the fact that their parameter restriction is a finite interval.

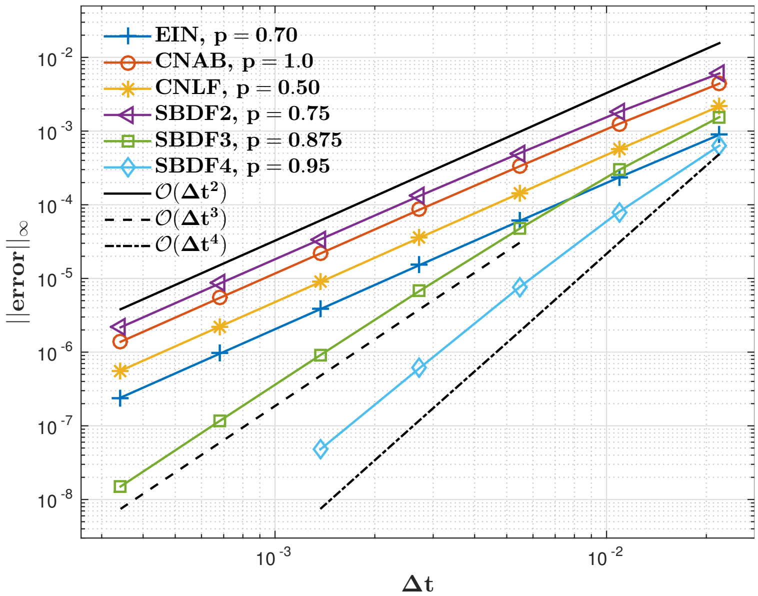

Figure 3.7: Numerical convergence study to (3.11) with IMEX methods. Convergence of EIN is also included for comparison. Test with mCNAB is omitted, but would otherwise overlap with CNAB.

Table 3.3: (Non)convergence of SBDF3 at various p.

	Observed convergence rate		
Δt	$p = 0.875$	$p = 1.475$	$p = 1.675$
2.19×10^{-2}	–	–	–
1.09×10^{-2}	2.39	2.39	2.39
5.47×10^{-3}	2.64	2.64	1.23
2.73×10^{-3}	2.80	2.80	−1.51
1.37×10^{-3}	2.90	2.90	−3.95
6.84×10^{-4}	2.95	2.95	diverge
3.42×10^{-4}	2.97	2.08	diverge

To see this, recall the parameter restrictions listed in Table 3.2 and the relation (2.19). Combining, we have for SBDF3 the parameter restriction

$$\frac{7}{8(1+(D_1\bar{u}_j^n)^2)} \leq p \leq \frac{2}{1+(D_1\bar{u}_j^n)^2}, \quad j = 1,\ldots,N, \tag{3.12}$$

or equivalently

$$\max_{1\leq j\leq N} \frac{7}{8(1+(D_1\bar{u}_j^n)^2)} \leq p \leq \min_{1\leq j\leq N} \frac{2}{1+(D_1\bar{u}_j^n)^2}. \tag{3.13}$$

Failure of the method is due to being unable to satisfy the parameter constraint at every grid node simultaneously. From Fig. 2.1, we see that $\max_j (D_1\bar{u}_j^n)^2$ is increasing as the solution evolves and is most extreme near the boundaries. Thus we expect instabilities to develop, and to develop in those regions first. This analysis is corroborated by Fig. 3.8, where we see instabilities developing near the righthand boundary.

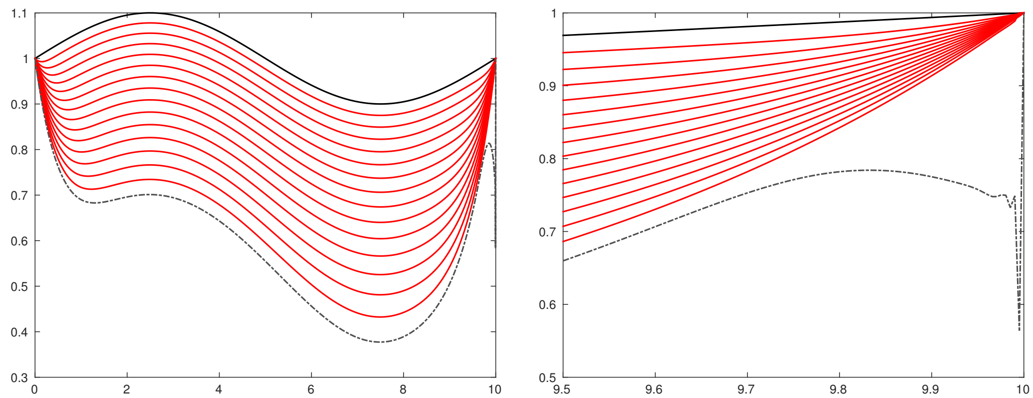

Figure 3.8: Instabilities using SBDF3. With $p = 1.675$ and $\Delta t = 9.2 \times 10^{-4}$, we observe instabilities developing near the righthand boundary of the gray dashed curve. The figure on the right is a zoom-in to the righthand boundary.

With SBDF4, the issue is only worse as the restriction is tighter. The results in Fig. 3.7 only appeared acceptable at coarse step-sizes because the low number of time steps did not allow for the instabilities to amplify and dominate the solution. We conclude that linear stabilization with SBDF3 and SBDF4 is not recommended.

A natural follow-up is then to ask if all third and fourth order IMEX schemes are unsuited for combination with linear stabilization. To this we provide a partial answer in the negative. Third order, three step schemes form a three parameter family, and likewise for the fourth order, four step schemes [1]. An extensive search through the parameter space so far has yielded no positives, nor is there any evidence suggesting we may find one with unbounded p-parameter restriction.

This leaves us a number of competing second order methods. Our next task is to compare the performance of our IMEX based schemes vs. the EIN method of [4].

3.4 Comparison of the Second Order Methods

Of the methods that we have proposed, only the second order variants remain. Along with the EIN method, that gives five second order linearly stabilized schemes. A performance evaluation is in order.

3.4.1 A 2D test problem

Let us use as a test problem, the following nonlinear PDE from [22]:

$$u_t = \Delta(u^5), \quad 0 \leq x, y \leq 1, \quad t > 0, \tag{3.14a}$$

with initial and boundary conditions set so that the exact solution is

$$u(x, y, t) = \left(\frac{4}{5}(2t + x + y)\right)^{1/4}. \tag{3.14b}$$

Further assuming a uniform grid and centred differences in space, the eigenvalues of the Jacobian of the linearization are estimated to be in the interval

$$\left(-\frac{64}{h^2}(1+t), -16\pi^2(t+h)\right). \tag{3.15}$$

To (3.14), we propose to stabilize with a $p\Delta u$ term, i.e., replace (3.14a) with

$$u_t = \Delta(u^5) - p\Delta u + p\Delta u, \quad 0 \leq x, y \leq 1, \quad t > 0. \tag{3.16}$$

The parameter p will then be chosen according to the ratio

$$\frac{p\lambda_{\mathcal{L}}}{\lambda_{\mathcal{N}}} \approx \frac{-8p/h^2}{-64(1+t)/h^2} = \frac{p}{8(1+t)}. \tag{3.17}$$

Let us remark on the importance of this test problem. In the analysis of (2.11), we performed a von Neumann analysis to obtain precise eigenvalue estimates. On the contrary, the interval (3.15) provides a bound that we do not expect to be sharp. We also never update p. It is chosen once and fixed at that value as we time step. Consequently, p may be substantially greater than necessary. Moreover, to compensate for the fact that the operator we introduced to stabilize is less stiff than the nonlinear term, we are forced to select a relatively large value for p. We expect the situations just described to be common in applications and thus it is of interest to investigate performance of the methods and behaviour of its discretization errors.

3.4.2 Loss of accuracy with EIN

We test our second order methods on (3.16) with initial and boundary conditions set by (3.14b). We solve to time $T = 0.40$ with spatial grid size $\Delta x = \Delta y = h = 0.015$. As a reference solution, we use an explicit third order Runge-Kutta method with time step-size $\Delta t = 6.25 \times 10^{-6}$. (Typical explicit methods demand step-sizes comparable for stability.) With the linearly stabilized schemes, we will solve (3.16) with time step-sizes as large as $\Delta t = 1.25 \times 10^{-2}$. We show the max norm relative error.

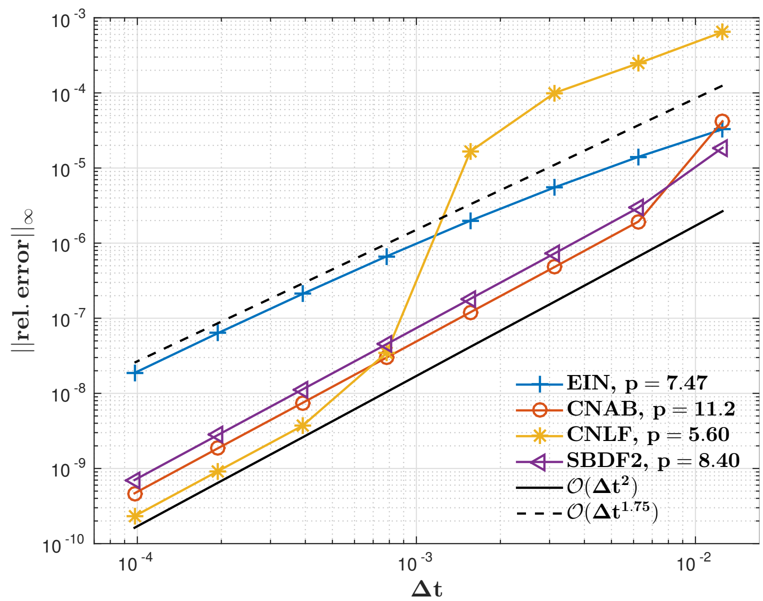

Figure 3.9: Numerical convergence study comparing second order methods. Test with mCNAB ($p = 9.97$) is omitted, but would otherwise overlap with CNAB.

Results of the numerical convergence test is plotted in Fig. 3.9 and it paints a grim picture for the EIN method and for CNLF. We will first discuss the horrific performance of the EIN method, after which we comment on the relative performance of the IMEX based schemes.

Contrasting the performance of EIN in Fig. 3.7 and Fig. 3.9, we observe a drastic reduction in the order of accuracy. In the former, EIN converged with second order accuracy and recorded, amongst the second order method, the lowest error at any fixed Δt (although an analysis of error vs. computing time would treat it less favorably). In our latest test problem, we do not (yet) observe second order convergence. Further testing shows that the EIN method only begins to demonstrate full second order rate of convergence as we dip below $\Delta t = 1 \times 10^{-5}$.

In fact, we argue that this behaviour can already be observed in the original paper by Duchemin and Eggers [4]. In their experiments with Hele-Shaw interface flow and with the Kuramoto-Sivashinsky equation, they fail to accurately reproduce the reference figures taken from prior publications [11, 12]. In both cases, a large value of p was necessary to obtain the desired stability.

We offer an explanation. In the simple case of the modified test equation (2.13), we can apply the EIN method and derive the amplification factor (2.16). A series expansion at $z = 0$ gives

$$\xi_{EIN} = 1 + z + \frac{z^2}{2} + \frac{1}{2}(p - p^2)z^3 + \mathcal{O}(p^3 z^4). \tag{3.18}$$

This expression deviates from the exact solution, $\exp(z)$, in the z^3 term and shows a quadratic dependence on p. That is to say, the leading error term may be large if p is large and Δt is not sufficiently small to control the error. That is exactly what is observed in Fig. 3.9.

On the other hand, the methods we have proposed all show only a linear dependence on p under the same analysis, and do not suffer catastropically when p is large.

3.4.3 Amplification factors at infinity

In the last section, we discovered a deficiency of the EIN method. The leading error term, which is $\mathcal{O}(\Delta t^3)$, has a coefficient that is quadratic in p, which we argued lead to a reduced observed order of accuracy. Thus it appears that an effective linearly stabilized time stepping scheme must have this error coefficient small with respect to p.

That, however, does not explain the equally horrific performance of CNLF or the sharp dip in the observed convergence of CNAB near $\Delta t = 1 \times 10^{-2}$ (see Fig. 3.9). To posit an explanation, we must think back to our discussion on stability and amplification factors. In that discussion, explicit schemes were said to be ill-suited to stiff problems because they are conditionally stable and this imposes a severe step-size restriction. Time steps violating this restriction will excite instabilities and energize high frequencies. A closer look at our schemes shows that we do exactly this. At each time step, there is an explicit calculation of a stiff nonlinear term and the other half of the "adding zero" term. Although we have guaranteed that our schemes are stable, the presence of slow decaying instabilities and persistent high frequency modes can drive up the error and force us to use smaller time steps to adequately damp and get the expected convergence order.

To explore this aspect, we consider the result of the method's amplification factor as $\lambda\Delta t \to -\infty$. For example, with the EIN method, we found the amplification factor in (2.16). As $z \to -\infty$, we find

$$\lim_{z\to-\infty} |\xi_{EIN}| = \left|\frac{p^2 - 3p + 2}{p^2}\right|. \tag{3.19}$$

For the multistep schemes, we first find the limiting expression of the amplification polynomial, and then take the larger of the absolute value of the two roots. For instance, with CNAB, we start from (3.8) and compute

$$\lim_{z\to-\infty} \frac{\Phi_2}{z} = -\frac{1}{2}p\xi^2 - \left(\frac{3}{2} - p\right)\xi - \frac{1}{2}p + \frac{1}{2}. \tag{3.20}$$

The amplification factor at infinity for CNAB is then

$$\max\left\{\frac{\left|p-1+\sqrt{-2p+1}\right|}{p}, \frac{\left|1-p+\sqrt{-2p+1}\right|}{p}\right\} \tag{3.21}$$

Fig. 3.10 shows the amplification factor at infinity for all of our second order schemes as well as SBDF1, and explains the behaviour of CNLF. It is known that CNLF is weakly damping at high frequencies and should not be used for diffusive problems [1]. This is equally true in the linear stabilization framework. Selecting coarse time steps with this scheme gives very poorly damping and it is necessary to take small time steps to combat this deficiency.

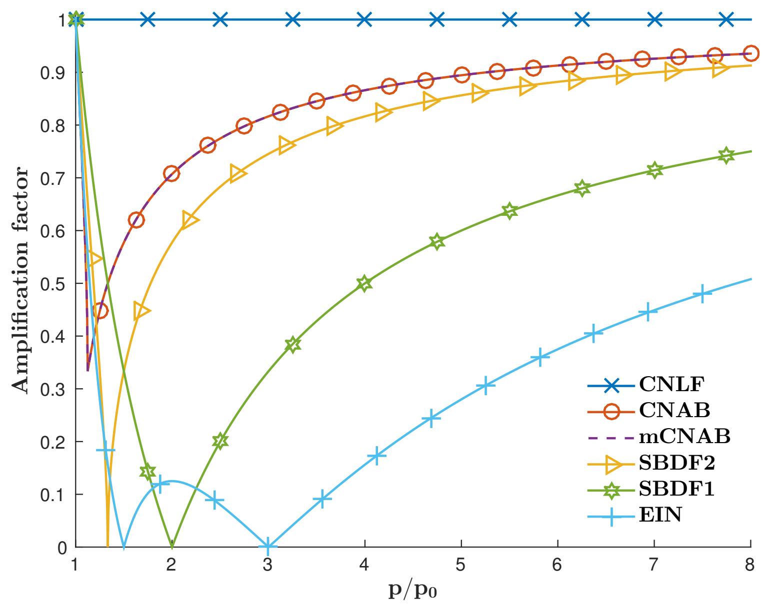

Figure 3.10: Amplification factors at infinity. The normalization along the horizontal axis is with respect to the lower limit of the parameter restriction of each scheme.

Out of our schemes, SBDF2 provides the most damping. CNAB (and mCNAB) may suffer from large errors at the coarsest time steps, but remains an useful alternative to SBDF2 as it exhibits lower errors as $\Delta t \to 0$. The scheme mCNAB was derived in [1] as an alternative to CNAB with stronger high frequency damping, however, in our framework, they perform identically. Thus there is little reason to consider mCNAB over CNAB.

Finally, let us remark on the plotting range along the horizontal axis and how it pertains to the implementation of linearly stabilized schemes. For one, we have chosen to plot in a normalized parameter space. There are many reasons for this, but the most compelling is from the point of view of how we use these schemes. Implementation of linearly stabilized schemes start by choosing p to satisfy the constraint

$$\frac{p\lambda_{\mathcal{L}}}{\lambda_{\mathcal{N}}} \geq p_0 \iff \frac{p}{p_0} \geq \frac{\lambda_{\mathcal{N}}}{\lambda_{\mathcal{L}}}, \tag{3.22}$$

where p_0 is the lower limit of the parameter restriction that guarantees unconditional stability, and where $\lambda_{\mathcal{L}}, \lambda_{\mathcal{N}}$ are the largest absolute eigenvalues of linear and nonlinear operators. The quantity $\lambda_{\mathcal{N}}/\lambda_{\mathcal{L}}$ is independent of the choice of time stepping method and if this quantity were overestimated in our solution procedure, the effect on p/p_0 is equal irrespective of the scheme. We have also plotted over a large range of parameter values. The reason is again rooted in the implementation. In our derivations, we operate as if we can access the exact eigenvalues at every time step. This is impractical and impossible. In practice, the ratio of eigenvalues will be an overestimate, and the value of p is fixed throughout the time evolution (or at least for a great number of time steps.) Therefore it necessary to chart the behaviour of the methods over a wide range of p.

Chapter 4

Higher Order with Exponential Integrators

The investigation with multistep schemes left us with a major question. Since the linearly stabilized schemes derived from SBDF3, SBDF4 were shown to be unsuitable for practical use, is it possible to construct practical high order linearly stabilized time stepping methods? In this chapter, we consider two methods coming from the class of exponential integrators. We will demonstrate that the second and fourth order exponential Runge-Kutta from Cox and Matthews [2] work well within our linear stabilization framework.

4.1 Exponential Runge-Kutta

Consider the ODE

$$\frac{du}{dt} = \mathcal{N}(u) + \mathcal{L}u. \tag{4.1}$$

Exponential time differencing, or exponential integrators, is a family of time stepping methods that treats the linear part exactly, and approximates the nonlinear part by some suitable quadrature formula. As an example, the exponential Euler has the formula

$$u^{n+1} = e^{\Delta t\mathcal{L}}u^n + \mathcal{L}^{-1}(e^{\Delta t\mathcal{L}} - 1)\mathcal{N}(u^n). \tag{4.2}$$

This is a first order accurate exponential integrator.

Our investigation deals with explicit exponential Runge-Kutta methods only. This family of one-step methods have the form

$$u^{n+1} = e^{\Delta t\mathcal{L}}u_n + \Delta t\sum_{i=1}^{s} b_i(\Delta t\mathcal{L})\mathcal{N}(U^{n,i}) \tag{4.3a}$$

$$U^{n,i} = e^{c_i\Delta t\mathcal{L}}u^n + \Delta t\sum_{j=1}^{i-1} a_{ij}(\Delta t\mathcal{L})\mathcal{N}(U^{n,j}), \tag{4.3b}$$

and can be presented in the familiar Butcher tableau:

$$\begin{array}{c|cccc} c_1 & & & & \\ c_2 & a_{21} & & & \\ \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & \cdots & a_{s,s-1} & \\ \hline & b_1 & \cdots & b_{s-1} & b_s \end{array}. \tag{4.4}$$

(Note that we have suppressed the argument, but these are indeed functions of $\Delta t\mathcal{L}$.) In particular, we focus on the second and fourth order exponential Runge-Kutta formulas of Cox and Matthews [2],

$$\begin{array}{c|cc} 0 & & \\ 1 & \varphi_{1,2} & \\ \hline & \varphi_1 - \varphi_2 & \varphi_2 \end{array}, \tag{4.5}$$

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & \frac{1}{2}\varphi_{1,2} & & & \\ 1/2 & 0 & \frac{1}{2}\varphi_{1,3} & & \\ 1 & \frac{1}{2}\varphi_{1,3}(\varphi_{0,3}-1) & 0 & \varphi_{1,3} & \\ \hline & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 4\varphi_3 - \varphi_2 \end{array}, \tag{4.6}$$

where

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - 1/k!}{z}, \quad \varphi_0(z) = \exp(z), \quad \text{and} \quad \varphi_{i,j}(z) = \varphi_i(c_j z). \tag{4.7}$$

We refer to this pair of exponential Runge-Kutta methods as ETDRK2 and ETDRK4.

4.2 Stability of ETDRK2 and ETDRK4

As before, we take the schemes (4.5), (4.6), and apply them to the modified test equation (2.13) to determined the parameter restriction over which the scheme is unconditionally stable. With the help of the computer algebra system, MAPLE™, the parameter restriction is found to be $[1/2, \infty)$ in both cases. We verify these findings with a series of stability contour plots in Figs. 4.1 and 4.2.

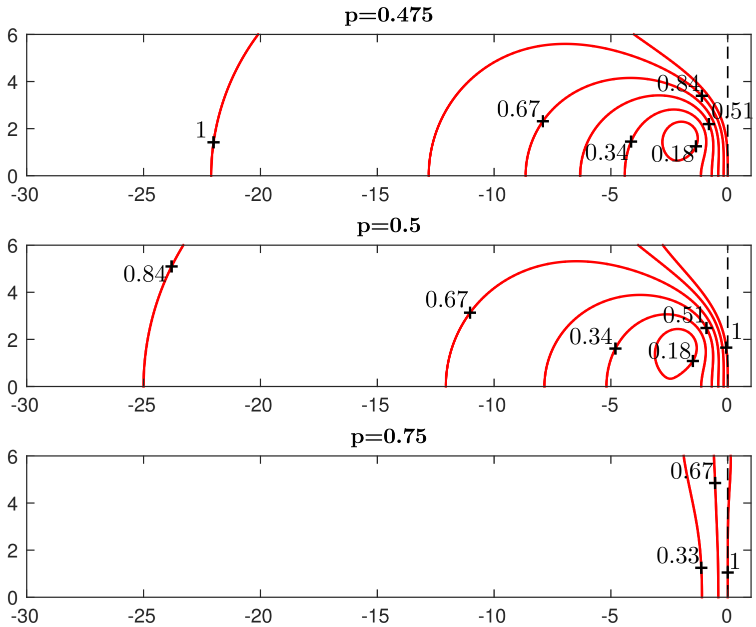

Figure 4.1: Stability contours for ETDRK2 at various p.

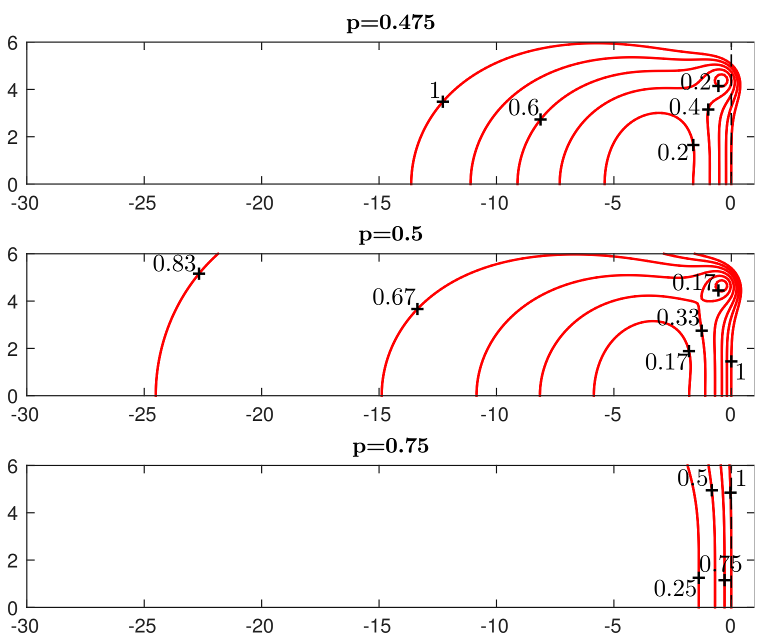

Figure 4.2: Stability contours for ETDRK4 at various p.

4.3 Numerical Results

In the last section, we verified that ETDRK2 and ETDRK4 was suitable for combination with linear stabilization on the modified test equation. The logical next step is to test on a more substantial problem and perform a numerical convergence study. However, the implementation of exponential integrators is not a trivial matter. We must first discuss potential issues and the method in which we choose to resolve them.

4.3.1 Stable evaluation of the matrix exponential and related functions

In Section 4.1, we have presented two exponential Runge-Kutta methods in tableau form and coefficients are combinations of functions of the operator, $\Delta t\mathcal{L}$. For example, to properly implement ETDRK2, we must compute the functions

$$\varphi_1(c\Delta\mathcal{L}) = (c\Delta t\mathcal{L})^{-1}(\exp(c\Delta t\mathcal{L}) - 1), \tag{4.8}$$

$$\varphi_2(c\Delta\mathcal{L}) = (c\Delta t\mathcal{L})^{-2}(\exp(c\Delta t\mathcal{L}) - 1 - c\Delta t\mathcal{L}), \tag{4.9}$$

for $c \in [0, 1]$, or be able to efficiently evaluate the related matrix-vector multiplications without explicit construction.

Difficulties with the evaluation of the matrix exponential and related functions of the form (4.7) are well-known and well-studied, eg. [14, 8, 9, 10]. In fact, struggles with the accurate and stable evaluation of the matrix exponential put a damper on the research into these methods and it was only with recent improvements and development of new techniques [10, 18, 12, 19, 9] that interest has revived and caught fire.

In all our examples, we follow the direction of Kassam and Trefethen [12]. They take a contour integral approach for the evaluation of functions in the form of (4.7) coupled with the trapezoidal rule for fast, accurate, and stable computations. Moreover, to avoid the unpleasantness of boundary conditions (and the subsequent complications), our domain will be periodic when using ETDRK2 or ETDRK4.

4.3.2 Numerical convergence test

4.4 Notes

Talk about parameter restriction for each scheme, stability plots, and maybe a little about damping at infinity.

These schemes also suck with large p. Can it be salvaged? What about exponential multistep methods? No, exponential multistep does not extend to high order.

We emphasize that our purpose is merely to demonstrate the potential of linear stabilization with ETD schemes and not a comprehensive survey.

+ Strengths of these schemes? Difficulties?

+ Modified test equation

+ Numerical Results with 2D periodic mean curvature motion problem

+ Compare with results from IMEX schemes

+ Computing time comparison

Chapter 5

Numerical Experiments

This chapter is entirely devoted to solving stiff PDEs prevalent in a number of fields. Through modelling of phenomenon in interfacial flows, demonstrating an application to image processing, and solving phase separation models on peculiar surfaces, we establish that our methods are easily applicable and can significantly reduce the runtime of 2D and 3D experiments.

5.1 Image Inpainting

The task in image inpainting is to repair corrupted images and damaged artwork. The restoration of artwork is the original motivation behind the development of the techniques in this area. Aged artwork would be restored to its former self by trained art restorers by extending the surrounding information into the missing/inpainting region. As an applied mathematician working on image inpainting, our goal is very much still to restore corrupted images, though not by hand, but in an automated fashion.

In [figure], we have a photograph of a sea turtle that has been vandalize and overlaid with text. Schoenlieb and friends [cite] proposed an assortment of fourth order PDE model for image restoration to images similar to this and compares her results to other state-of-the-art variational image inpainting models [cite]. One of the proposed is the so-called TV-H^{-1} inpainting:

$$u_t = -\Delta \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \epsilon^2}} \right) + \lambda_D (u_0 - u). \tag{5.1}$$

In here and what follows, u, the solution, is the inpainted image we seek. The quantity, u_0, is the initial corrupted image, and the parameter $\epsilon > 0$ is the necessary regularization. The symbol Ω denotes the image domain, D denotes the inpainting region and λ_D is defined as

$$\lambda_D(x) = \begin{cases} \lambda_0, & x \in \Omega \setminus D \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

Interestingly, her strategy for the time evolution of the PDE model is exactly the first order linearly stabilized SBDF1.

In [figure], we have an photograph of a bullfinch that has been vandalized.

5.2 Motion by Mean Curvature

5.3 Phase Separation Models on Fun Surfaces

Chapter 6

Conclusion

A careful study of the behaviour of the errors from choosing a (sub)optimal parameter, p.

+emphasize that the inversion is of the same operator at each time step

Bibliography

[1] Uri M Ascher, Steven J Ruuth, and Brian TR Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.

[2] Steven M Cox and Paul C Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2):430–455, 2002.

[3] Jim Douglas Jr and Todd Dupont. Alternating-direction Galerkin methods on rectangles. In Bert Hubbard, editor, *Numerical Solution of Partial Differential Equations II*, pages 133–214. Academic Press, 1971.

[4] Laurent Duchemin and Jens Eggers. The explicit–implicit–null method: Removing the numerical instability of PDEs. *Journal of Computational Physics*, 263:37–52, 2014.

[5] David J Eyre. Unconditionally gradient stable time marching the Cahn-Hilliard equation. In *MRS Proceedings*, volume 529, page 39. Cambridge Univ Press, 1998.

[6] David J Eyre. An unconditionally stable one-step scheme for gradient systems. *Unpublished article*, 1998.

[7] Karl Glasner. A diffuse interface approach to hele–shaw flow. *Nonlinearity*, 16(1):49, 2002.

[8] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. Siam, 2002.

[9] Nicholas J Higham. *Functions of matrices: theory and computation*. Siam, 2008.

[10] Marlis Hochbruck and Christian Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 34(5):1911–1925, 1997.

[11] Thomas Y Hou, John S Lowengrub, and Michael J Shelley. Removing the stiffness from interfacial flows with surface tension. *Journal of Computational Physics*, 114(2):312–338, 1994.

[12] Aly-Khan Kassam and Lloyd N Trefethen. Fourth-order time-stepping for stiff PDEs. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, 2005.

[13] Colin B Macdonald and Steven J Ruuth. The implicit closest point method for the numerical solution of partial differential equations on surfaces. *SIAM Journal on Scientific Computing*, 31(6):4330–4350, 2009.

[14] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1):3–49, 2003.

[15] David Salac and Wei Lu. A local semi-implicit level-set method for interface motion. *Journal of Scientific Computing*, 35(2-3):330–349, 2008.

[16] Carola-bibiane Schönlieb and Andrea Bertozzi. Unconditionally stable schemes for higher order inpainting. *Communications in Mathematical Sciences*, pages 413–457, 2011.

[17] Avram Sidi. *Practical extrapolation methods: Theory and applications.* Cambridge University Press, New York, NY, USA, 1st edition, 2003.

[18] Roger B Sidje. Expokit: a software package for computing matrix exponentials. *ACM Transactions on Mathematical Software (TOMS)*, 24(1):130–156, 1998.

[19] Valeria Simoncini and Daniel B Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications*, 14(1):1–59, 2007.

[20] Peter Smereka. Semi-implicit level set methods for curvature and surface diffusion motion. *Journal of Scientific Computing*, 19(1):439–456, 2003.

[21] J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations, Second Edition.* Society for Industrial and Applied Mathematics, 2004.

[22] P. J. van der Houwen. On the time integration of parabolic differential equations. In G. Alistair Watson, editor, *Numerical Analysis: Proceedings of the 9th Biennial Conference Held at Dundee, Scotland, June 23–26, 1981*, pages 157–168. Springer Berlin Heidelberg, Berlin, Heidelberg, 1982.

Appendix A

Code