

# Positivity Preserving Solutions of Partial Integro-Differential Equations

by

Alexander M. Lewis

Honors B.S. Chemical Engineering  
Oregon State University, 2004

M.S. Chemical Engineering Practice  
Massachusetts Institute of Technology, 2007

SUBMITTED TO THE DEPARTMENT OF CHEMICAL ENGINEERING IN  
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN CHEMICAL ENGINEERING PRACTICE  
AT THE  
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 15<sup>th</sup>, 2009

[JUNE]

© Massachusetts Institute of Technology

All rights reserved

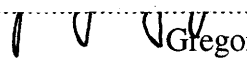
**ARCHIVES**

Signature of Author.....

Department of Chemical Engineering

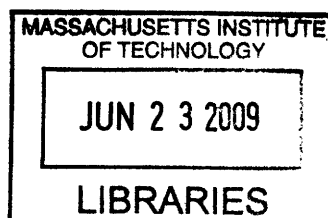
May 15<sup>th</sup>, 2009

Certified by.....

  
Gregory J. McRae  
Hoyt C. Hottel Professor of Chemical Engineering  
Thesis Supervisor

Accepted by.....

William M. Deen  
Professor of Chemical Engineering  
Chairman, Committee of Graduate Students



# Positivity Preserving Solutions of Partial Integro-Differential Equations

by  
Alexander M. Lewis

Submitted to the Department of Chemical Engineering on May 15<sup>th</sup>, 2009 in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Chemical Engineering Practice

## Abstract

Differential equations are one of the primary tools for modeling phenomena in chemical engineering. While solution methods for many of these types of problems are well-established, there is growing class of problems that lack standard solution methods: partial integro-differential equations. The primary challenges in solving these problems are due to several factors, such as large range of variables, non-local phenomena, multi-dimensionality, and physical constraints. All of these issues ultimately determine the accuracy and solution time for a given problem.

Typical solution techniques are designed to handle every system using the same methods. And often the physical constraints of the problem are not addressed until after the solution is completed if at all. In the worst case this can lead to some problems being over-simplified and results that provide little physical insight. The general concept of exploiting solution domain knowledge can address these issues.

Positivity and mass-conservation of certain quantities are two conditions that are difficult to achieve in standard numerical solution methods. However, careful design of the discretizations can achieve these properties with a negligible performance penalty. Another important consideration is the stability domain. The eigenvalues of the discretized problem put restrictions on the size of the time step. For “stiff” systems implicit methods are generally used but the necessary matrix inversions are costly, especially for equations with integral components. By better characterizing the system it is possible to use more efficient explicit methods. This work improves upon and combines several methods to develop more efficient methods.

There are a vast number of systems that be solved using the methods developed in this work. The examples considered include population balances, neural models, radiative heat transfer models, among others. For the capstone portion, financial option pricing models using “jump-diffusion” motion are considered. Overall, gains in accuracy and efficiency were demonstrated across many conditions.

Thesis Supervisor: Gregory J. McRae

Title: Hoyt C. Hottel Professor of Chemical Engineering

## **Acknowledgements**

Writing this thesis has been a challenging and rewarding process. Much like my time at MIT in general, it has had its ups and downs, but the end result was well worth the effort. I will certainly never forget these past five years; I have probably learned more over this period than any other in my life, both in and outside of school.

First and foremost, I would like to thank my Parents and Grandparents without whose support none of my education would have been possible.

The most important factor in completing my thesis has been my adviser, Greg McRae. The breadth and depth of his knowledge as well as his encouragement were equally important along the way. In addition, my Thesis Committee, Leonid Kogan, and Charles Cooney also provided valuable support.

Outside of school, an important part of my life was my home for five years, Sidney-Pacific. Working with people in the house government there were some of my best memories at MIT and I'd particularly like to acknowledge my friends from SPEC, Rob Wang, Ben Mares, Matt Eddy, and Jane Kim.

I'd also like to thank my lab mates past and present for their support and insight: Jeremy Johnson, Patrick de Man, Sarah Passone, Chuang-Chung Lee, Ingrid Berkelmans, Mihai Anton, Arman Haidari, Temi Popoola, Bo Gong, Anusha Kothandaraman, Ashleigh Hildebrand, Ken Hu, Liqiang Duan, Kunle Adeyemo, Carolyn Seto, and Rajat Suri. Having them around helped make the long days in the basement more pleasant.

## Table of Contents

1.0 Introduction.....	9
1.1 Thesis Statement.....	9
1.2 Rationale.....	9
1.2.1 Problem Domain Information.....	9
1.2.2 Solution Domain.....	11
1.2.3 Parameter Estimation.....	11
1.3 Overall Contribution.....	12
2.0 Background.....	13
2.1 Systems of Interest.....	13
2.2 Numerical Solution Methods.....	18
2.2.2 Mass Conservation.....	21
2.2.3 Time Stepping Methods.....	21
2.2.4 Convergence.....	22
2.2.5 Error Estimation.....	23
2.2.6 Stability Domain.....	25
2.2.7 Stiffness.....	29
2.2.8 Runge Kutta Methods.....	34
2.2.9 Linear Multistep Methods.....	37
2.2.10 Fourier Decompositions.....	38
2.2.11 Boundary Conditions.....	39
2.3 Transport Example Problem.....	41
3.0 Integral Equations.....	45
3.1 PIDE Solution Methods.....	46
3.2 Examples.....	47
3.2.1 Population Balance PIDEs.....	49
3.2.2 Other Examples.....	52
4.0 New Applications: Finance.....	55
4.1 Comparison with Chemical Engineering.....	55
4.2 Derivations of Fundamental Equations.....	59
4.3 Jump Processes.....	67
4.4 Option Example Problem.....	70
5.0 Novel Solution Techniques.....	73
5.1 Runge-Kutta Chebyshev.....	73
5.2 Positivity Preservation.....	76
5.3 Operator Splitting.....	81
5.4 High Dimensional Systems.....	86
6.0 Spatial Discretization Methods.....	88
6.1 Implementation of the Positivity Preserving Method.....	88
6.2 More Details of the Implementation.....	92
6.3 Results.....	94
7.0 Implementation of the Runge-Kutta Chebyshev Method.....	95
7.1 Detailed Development of the RKC Method.....	95
7.1.1 Standard RKC.....	95
7.1.2 RKC for Advection-Diffusion.....	99

7.1.3 Mapping Out Eigenvalues.....	105
7.1.4 Reaction Problems and the IMEX Method.....	110
7.1.5 Error Control.....	114
8.0 Actual Implementation and Results .....	120
8.1.1 Two-Step RKC IMEX Function .....	120
8.1.2 Problem Setup Function.....	123
8.2 Results from New Techniques .....	127
8.2.1 Two-Step RKC.....	127
8.2.2 Positivity Preservation .....	128
8.2.3 RKC IMEX .....	132
8.2.4 Error Correction.....	134
8.2.5 Other Problem Types .....	136
8.2.6 Comparison with Other Methods.....	140
8.2.7 Summary .....	145
9.0 Population Balance Systems .....	147
9.1 Notation.....	147
9.2 Basic Phenomena .....	149
9.2.1 Condensation/Evaporation.....	149
9.2.2 Coagulation .....	152
9.2.3 Other Phenomena.....	154
9.3 Equation Forms & Analytical Solutions.....	155
9.3.1 Condensation Equation .....	155
9.3.2 Coagulation Equation.....	161
9.4 Numerical Solutions.....	165
9.4.1 Condensation Equation .....	166
9.4.2 Coagulation Portion .....	170
9.5 Implementation and Results.....	173
9.5.1 Condensation Examples.....	173
9.5.2 Coagulation Examples .....	175
10.0 Other Examples.....	180
10.1 Neural Example .....	180
10.1.1 Problem Background and Setup.....	180
10.1.2 Problem Solution .....	185
10.2 Radiative Heat Transfer .....	188
10.2.1 Problem Background and Setup.....	188
10.2.2 Problem Solution .....	198
11.0 Conclusions and Directions for Future Research.....	202
11.1 Conclusions.....	202
11.2 Directions for Future Research .....	203
Appendix A: Capstone Paper.....	205
A.1 Executive Summary .....	205
A.2 Introduction .....	206
A.3 Background .....	206
A.4 Implementation .....	227
A.4.1 Numerical Set-Up .....	228
A.4.2 Actual Program .....	232

A.5 Results.....	234
A.6 Conclusions.....	244
A.7 Capstone Paper Works Cited .....	245
Appendix B: References .....	246

## List of Figures

Figure 1.1: Problem Domain.....	10
Figure 1.2: Modified Stability Domain.....	11
Figure 2.1: Stability Domains for Explicit Euler (left) and Trapezoid Rule (right) .....	27
Figure 2.2: A-Stability; A( $\alpha$ )-Stability; Stiff Stability.....	28
Figure 2.3: Instability of Explicit Euler Method Solutions, Concentration vs. Time. ....	31
Figure 2.4: Transport Example, initial condition (dashed line) and approximation.....	43
Figure 4.1: Value of a European call option .....	59
Figure 4.2: Payoff function (dashed line) and approximate option value vs. asset price .	72
Figure 5.1: Shifted Chebyshev polynomial, $s=4$ .....	74
Figure 5.2: Stability domain for shifted Chebyshev polynomial, $s=4$ .....	75
Figure 5.3: Stability domain for damped, shifted Chebyshev polynomial, $s=4$ , $\varepsilon=2/13$ ..	75
Figure 5.4: Convection Equation, Initial Condition.....	77
Figure 5.5: Concentration vs. Position, Smooth Profile .....	78
Figure 5.6: Concentration vs. Position, 3 profiles .....	81
Figure 7.1: Eigenvalues for typical diffusion (left) and advection problems .....	100
Figure 7.2: Effect of increasing $\varepsilon$ on stability domain; 5 stages, $\varepsilon = 2/13, 5, 10, 100$ ....	101
Figure 7.3: Stability domains for 2 step RKC; 3, 4, 7, 10 stages.....	104
Figure 7.4: Example initial conditions, conc. vs. position.....	109
Figure 8.1: Typical Initial Conditions; concentration vs. position .....	127
Figure 8.2: Eigenvalues for 3rd order upwind biased discretization .....	128
Figure 8.3: Advection with 2 <sup>nd</sup> order central discretization.....	129
Figure 8.4: Advection with 2 <sup>nd</sup> order upwind discretization.....	130
Figure 8.5: Advection with 3 <sup>rd</sup> order upwind biased discretization.....	130
Figure 8.6: Dirichlet Boundary Conditions, $t=100$ (left) & $t=1000$ (right).....	137
Figure 8.7: Neumann Boundary Conditions, $t=1000$ .....	137
Figure 8.8: Advection with no flux BC; Initial condition (solid line) and solution.....	138
Figure 8.9: Advection with no flux BC, corrected solution and solution with diffusion	138
Figure 8.10: 2-D Advection .....	139
Figure 8.11: 3-D Advection .....	139
Figure 8.12: Time Steps vs. Diffusion Coef .....	140
Figure 8.13: Function Evaluations vs. Diffusion Coef .....	141
Figure 8.14: Time Steps vs. Velocity.....	141
Figure 8.15: Function Evaluations vs. Velocity.....	142
Figure 8.16: Time Steps vs. $k_2$ .....	142
Figure 8.17: Function Evaluations vs. $k_2$ .....	143
Figure 8.18: Time Steps vs. $k_2$ .....	143
Figure 8.19: Function Evaluations vs. $k_2$ .....	144
Figure 9.1: Condensation Growth, Mass Distribution, with closeup.....	158
Figure 9.2: Eigenvalues for Condensation Equation, "Advective" Portion.....	168
Figure 9.3: Eigenvalues for Condensation Equation, Both Portions .....	168
Figure 9.4: Eigenvalues, Coagulation Eqn.; Exponential (left) and Square Pulse IC ....	172
Figure 9.5: Condensation equation example results .....	174
Figure 9.6: Condensation equation, non-filtered version.....	175
Figure 9.7: Condensation and coagulation, 2 species system.....	176

Figure 9.8: Coagulation only, uniform .....	177
Figure 9.9: Coagulation only, similar-size dominated.....	178
Figure 9.10: and coagulation, 2 species system, variable parameters .....	179
Figure 10.1: Rate of change of the firing rate.....	181
Figure 10.2: Symmetric activation kernel.....	181
Figure 10.3: Kernel function (left) and activation function; $b=0.25$ , $r=0.1$ , $\theta=1.5$ .....	183
Figure 10.4: One and two bump solutions to the stationary problem.....	184
Figure 10.5: Eigenvalues of integral portion .....	185
Figure 10.6: Long-time solution, 201 grid points, $L=6$ .....	186
Figure 10.7: Radiative example .....	198
Figure 10.8: Temperature profile for $\tau_L=0.1$ (left), and $\tau_L=1.0$ , and $\tau_L=10$ (bottom) ....	200
Figure A.1: Discrete price tree.....	208
Figure A.2: Different probability measures, discrete case.....	209
Figure A.3: Discrete tree with cumulative probabilities .....	212
Figure A.4: Cadlag function .....	219
Figure A.5: Levy distribution for Gamma case .....	225
Figure A.6: <i>xfull</i> vector .....	230
Figure A.7: <i>G</i> matrix.....	230
Figure A.8: Eigenvalues for the first (left) and second derivative components .....	232
Figure A.9: Merton Gaussian jump model for European Call (left) and Put options .....	234
Figure A.10: Volatility surface, Put option with Merton Gaussian jumps .....	235
Figure A.11: Approximate and Exact call (left) and put option value solutions .....	237
Figure A.12: Volatility surface, Put option with Merton Gaussian jumps .....	238
Figure A.13: Gamma model value of call (left) and put options; 2 cases .....	239
Figure A.14: Volatility surface for Put option, Gamma model, Cases 1 (top) and 2.....	240
Figure A.15: Variable parameter case, Merton model for call (left) and put options.....	242
Figure A.16: Variable parameter case, Gamma model for call (left) and put options....	242
Figure A.17: Volatility surfaces, variable cases for Merton (top) and Gamma models .	243



## List of Tables

Table 4.1: Financial Assumptions.....	57
Table 7.1: Eigenvalues for common advection schemes .....	107
Table 7.2: Eigenvalues for common diffusion schemes .....	107
Table 7.3: Eigenvalues due to positivity filter .....	109
Table 8.1: Inputs for rkc2stepIMEX5.m.....	120
Table 8.2: Optimal stability boundary for a given number of stages.....	121
Table 8.3: Inputs for advDiffRxnMultiDim1.m .....	124
Table 8.4: Advection Test Error Results.....	131
Table 8.5: Standard RKC vs. IMEX RKC; reaction-diffusion problem.....	133
Table 8.6: Error Correction Results .....	135
Table 9.1: Standard Distributions .....	148
Table 9.2: Solutions for basic forms of the Condensation Equation .....	158
Table 9.3: Condensation Equation Forms.....	167
Table 9.4: Approximate eigenvalue bounds for discretized variable velocity operator .	170
Table 9.5: Basic variables for MATLAB Population Balance function.....	173
Table 9.6: Condensation equation error comparison .....	175
Table 9.7: Solution efficiency results from Condensation and Coagulation examples ..	176
Table 10.1: Error evaluation, neuron base case .....	186
Table 10.2: Solution efficiency results from neural example .....	187
Table 10.3: Solution efficiency results from radiative heat transfer example .....	201
Table A.1: Inputs for finOpt_jumpPIDE8.m .....	232
Table A.2: Performance of Methods, Marton Gaussian jump case .....	236
Table A.3: Performance of methods, Gamma case.....	238
Table A.4: Performance of Methods, Merton case .....	241

## **1.0 Introduction**

### **1.1 Thesis Statement**

Modeling natural phenomena as systems is one of the primary aspects of chemical engineering. Differential equations have been one of the primary tools and their usage in chemical engineering has led to not only to better understanding of physical systems but also to advancements in mathematics. Indeed, much of the work of chemical engineering can be applied in a diverse array of fields.

The challenge of modeling systems ultimately comes down to solving systems of some class of differential equations. And of course, the systems that are of interest today are the most challenging. There are many characteristics that cause these challenges:

- Large number of variables: Interesting systems are described by many interacting factors
- Non-local phenomena: Some physics require considering effects over the whole solution space, requiring integral equations
- Wide range of space and time scales: The variables can take values over many orders of magnitude
- Physical constraints: The results must be bounded by known constraints and the solution method must capture these facts
- Solved numerous times: Parameter estimation and optimization may involve solving the same system with slightly perturbed values repeatedly

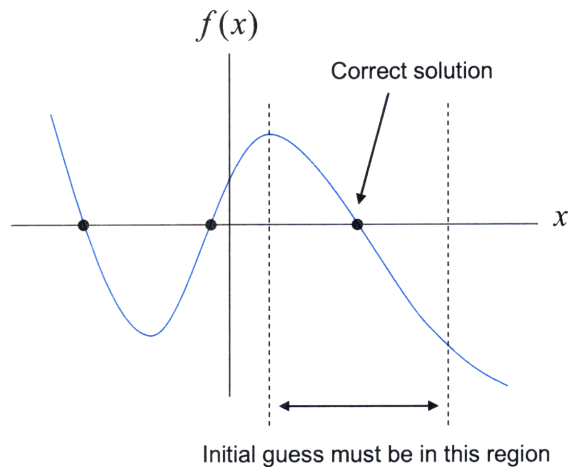
While a great body of work already exists on solving differential equations, many of the above issues have not been fully addressed. For example, most solution techniques do not exploit all available information about the problem domain. If as much as possible is known about this, then a solution method can be developed that is as efficient as possible for the situation at hand. Of course, determining all of the relevant information is not available without computational cost. But this up-front cost can become more than worthwhile in situations where the same system is solved a large number of times.

This illustrates the main focus of this work. This thesis demonstrates the development of solution techniques that exploit problem domain knowledge for the solution of estimation and optimization problems, particularly in partial integro-differential equation systems.

### **1.2 Rationale**

#### **1.2.1 Problem Domain Information**

The concept of problem domain information can be explained with a simple example. Consider a third degree polynomial, shown in Figure 1.1 below. Determining the zeros of the equation can be found using a standard Newton-Rhapson type method.



**Figure 1.1: Problem Domain**

However, it may be the case that physical reality constrains the solution to be positive. So a good initial guess is important not only in obtaining a quick convergence but in assuring convergence to the correct solution.

Clearly there is a benefit to taking a more advanced approach to solving these types of problems. First off one can use information about the physical situation the problem represents. As mentioned above, positivity is a common physical property. Knowing that fact immediately allows for a better initial guess to be selected. Knowledge of the problem domain is also useful. In this case, the problem domain amounts to recognizing the function as a third order polynomial the roots of which are determined by the coefficients.

All of this information is essentially represented in the figure above. The solution can be achieved much faster with the information than simply plugging into a Newton-Raphson solver with an arbitrary guess. And if this problem were repeatedly solved with slightly different parameters, the final answer from the first solution could be used as an initial guess for the next solution, likely leading to a very fast solution.

Now if this were a vector problem, a matrix inversion would be required.

$$\underline{x}^{(n+1)} = \underline{x}^{(n)} - \left( \frac{\partial f}{\partial \underline{x}} \bigg|_{\underline{x}^{(n)}} \right)^{-1} f(\underline{x}^{(n)})$$

The numerical iteration used to find this inverse could also be employed in the next solution since the matrix structure would be fairly similar. Overall, using information about the previous solution in each succeeding solution could make the entire process of optimization or estimation much more efficient.

While this is a trivial example, the very same concepts can be applied to differential equation systems. As will be shown throughout, problem domain information is one of the primary tools employed in this thesis.

### 1.2.2 Solution Domain

The location of the eigenvalues of a generalized system of differential equations

$$\frac{d}{dt} w = Aw$$

are essential to determining a practical solution technique. Effectively, the eigenvalues determine the maximum step size that can be used to obtain a convergent solution. Solution methods can be categorized by region in the complex domain to which the product of the timestep and the eigenvalue must be constrained. Clearly, maximizing this region for a given configuration of eigenvalues is desirable. The conventional route is to use implicit methods which have an effectively infinite stability domain. However, they are very computationally intensive. Explicit methods have a limited stability domain, but recent methods, e.g. [Hundsdorfer & Verwer, 2003], have shown that these domains can be modified to contain more of the negative real line.

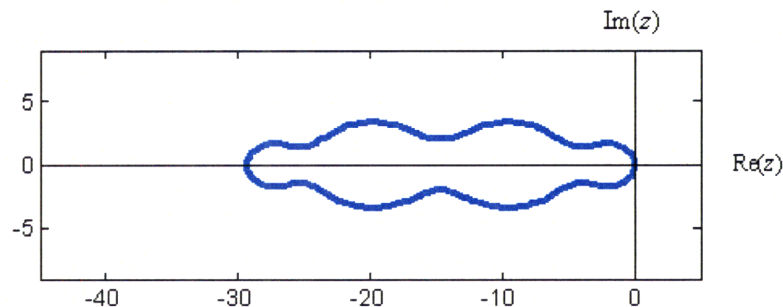


Figure 1.2: Modified Stability Domain

By solving the eigenvalue problem for the system initially, a solution method can be developed that efficiently captures the eigenvalues and will remain stable to small changes in the problem structure at each iteration of different parameters. This would lead to a very efficient solution once the up-front cost of finding the eigenvalues has been completed.

### 1.2.3 Parameter Estimation

Parameter estimation is characterized by solving similar systems a multitude of times. In control theory tools such as Kalman filters use previous measurements to estimate current parameter values. To be effective for control the problems need to be solved in real time. Thus the system needs to be solved as efficiently as possible.

Using standard methods would require solving the entire system each time a new estimate is needed. However, it is generally the case that the parameters only change slightly as time progresses. Such information effectively constitutes the problem domain. A solution method that takes advantage of information about the solution domain can solve the similar problem very quickly once the problem domain is initially calculated.

This same concept can be applied to any situation that needs to be solved repeatedly with similar parameters. The only major cost is the initially time and effort to fully characterize the initial problem domain.

### **1.3 Overall Contribution**

This thesis has developed several important techniques that can be applied to a wide range of systems of both partial differential and partial integro-differential equations. Methods from several sources have been enhanced and combined and applied to a variety of examples. The end result is a numerical method that exploits problem knowledge to achieve solutions that are accurate, preserve positivity, and perform in a computationally efficient manner.

Specifically, the methods developed in this thesis were applied to systems in population balances, neuron impulses, and radiative heat transfer, as well as option-pricing models. The results show significant speed advantages over standard Runge-Kutta and BDF methods. Additionally, the methods can be scaled up to  $n$ -dimensional systems.

The thesis follows the following outline:

- Background, covering the basics of numerical solution techniques
- Overview of integral equations
- A brief discussion of the links to equations used in finance
- Overview of novel solution techniques serving as the basis for the methods developed in this thesis
- Development of spatial discretization and time integration methods
- Implementation and results of methods
- Population balance example
- Neural example
- Radiation example
- Capstone Paper: Financial examples

## 2.0 Background

### 2.1 Systems of Interest

There are a plethora of systems that can be modeled with some class of differential equations. For most of the development of this thesis the focus will be on transport equations, but several examples will go beyond this. The underlying structures of many systems are quite similar when arranged properly, so the strategies discussed throughout can be readily applied to many diverse situations.

The transport equations are one of the most important concepts in the study of chemical engineering (and in the physical sciences in general). The basic relationships apply to the transfer of heat (temperature), mass (concentration), and momentum (velocity). For the purposes of this section, the equations are developed such that they can be applied directly to heat or mass transfer. This form will elucidate the analogy that is being put forth in the main body.

In heat transfer, temperature ( $T$ ) is the measured variable of interest and in mass transfer, it is species concentration ( $c$ ). These two phenomena are similar enough that they can be developed simultaneously using a common variable,  $b$ , which represents the “concentration” of either energy or a species. The total flux (flow per unit area) of  $b$  is represented by  $\underline{F}$ . Note that  $b$  and  $\underline{F}$  are both functions of position and time and that  $\underline{F}$  is a vector quantity.

It can be observed that heat flows from regions with high temperatures to regions of low temperatures and mass flows from regions of high to low concentrations of a given species. This is known as diffusive flux,  $\underline{f}$ , (and generally referred to as conduction for heat transfer and diffusion for mass transfer). These phenomena occur on the molecular level, and required empirical observation for the development of equations. This was first done by Fourier for heat transfer and later by Fick for mass transfer. In general terms, the relationship between flux and concentration is

$$\underline{f} = -a\nabla b$$

Where  $\nabla$  is the gradient operator, and  $a$  is either the thermal conductivity of a material ( $k$ ) or the diffusion coefficient of a given species in a medium ( $D_i$ ). Note that this assumes the conductivity or diffusion is the isotropic. If not,  $a$  must be represented by a tensor. Note also the negative relationship, to account for the transfer from high to low concentration regions. Finally note that the above  $D$  represents a tensor in general; in the one-dimensional case it is replaced by  $d$  hereafter.

Using the concepts of diffusive flux, the general conservation equations can be developed from first principles. This development is the most general possible, and is based on that of [Deen, 1998]. Consider a control volume of arbitrary shape with volume  $V$  and surface area  $S$ . This control volume could change with time. Any point can be defined in

terms of a position vector,  $\underline{r}$ , and time,  $t$ . Define  $\underline{n}$  as a vector normal to the surface and pointing outward. The velocity at a given point is  $\underline{v}(\underline{r}, t)$  and the velocity of the surface is  $\underline{v}_s(\underline{r}, t)$ . Note that if  $\underline{v} = \underline{v}_s$ , there is no fluid flow across the surface. The term  $B(\underline{r}, t)$  represents generation of  $b$  within the control volume.  $\underline{F}(\underline{r}, t)$  is the total flux of  $b$  at a given point.

To maintain the conservation of energy and mass,

$$\text{Accumulation} = \text{Input} - \text{Output} + \text{Generation}$$

must be maintained. For a fixed control volume ( $\underline{v}_s = 0$ ), this gives

$$\frac{d}{dt} \iiint_V b \, dV = - \iint_S \underline{F} \cdot \underline{n} \, dS + \iiint_V B \, dV$$

The negative sign on the middle term is necessary since  $\underline{n}$  points outward. Since the integral is over the whole surface, it encompasses input and output. To account for the volume swept out by the surface if the control volume is changing with time, one more term must be added

$$\frac{d}{dt} \iiint_{V(t)} b \, dV = - \iint_{S(t)} \underline{F} \cdot \underline{n} \, dS + \iiint_{V(t)} B \, dV + \iint_{S(t)} b \underline{v}_s \cdot \underline{n} \, dS \quad (2.1)$$

To simplify this expression, the Leibniz rule is employed

$$\frac{d}{dt} \int_{A(t)}^{B(t)} f(x, t) \, dx = \int_{A(t)}^{B(t)} \frac{\partial}{\partial t} f(x, t) \, dx + \frac{dB}{dt} f(B(t), t) - \frac{dA}{dt} f(A(t), t).$$

Taking  $A$  and  $B$  to be the outer positions of the surface, the difference of their derivatives amounts to an integration of surface velocities (in a direction normal to the surface) over the surface. This gives

$$\frac{d}{dt} \iiint_{V(t)} b \, dV = \iiint_{V(t)} \frac{\partial b}{\partial t} \, dV + \iint_{S(t)} b \underline{v}_s \cdot \underline{n} \, dS$$

Now (2.1) can be simplified to

$$\iiint_{V(t)} \frac{\partial b}{\partial t} \, dV = - \iint_{S(t)} \underline{F} \cdot \underline{n} \, dS + \iiint_{V(t)} B \, dV.$$

To evaluate this equation at a point, the limit  $V \rightarrow 0$  must be taken. First, though, the surface integral must be transformed into a volume integral. This is accomplished using the divergence theorem,

$$\iiint_{V(t)} \nabla \cdot \underline{F} \, dV = \iint_{S(t)} \underline{n} \cdot \underline{F} \, dS$$

to give

$$\iiint_{V(t)} \left( \frac{\partial b}{\partial t} + \nabla \cdot \underline{F} - B \right) dV = 0.$$

Any integral must equal the average value of the integrand times the value of the region of integration. Therefore,

$$\left\langle \frac{\partial b}{\partial t} + \nabla \cdot \underline{F} - B \right\rangle_{avg} V = 0$$

For this to be true for any volume, the bracketed value must always be equal to zero. In the limit as  $V \rightarrow 0$ , the bracketed value must equal its value at the point on which it is centered. This gives the general conservation equation,

$$\frac{\partial b}{\partial t} = -\nabla \cdot \underline{F} + B .$$

The total flux,  $\underline{F}$ , is the sum of the diffusive and convective fluxes. The diffusive flux was defined above. Convective flux is transport of a given quantity due to bulk motion of a fluid. If the mass-average velocity is used, the total flux is

$$\underline{F} = \underline{f} + b\underline{v} .$$

With this, the conservation equation is now

$$\frac{\partial b}{\partial t} + \nabla \cdot (b\underline{v}) = -\nabla \cdot \underline{f} + B . \quad (2.2)$$

While this equation seem fairly innocuous, its actual solution can prove quite challenging when all of the dependencies and interactions of the variables are considered. This equation can be directly applied to situations involving heat or mass transfer. In heat transfer,  $b$  becomes energy,  $\rho c_p T$ , where  $\rho$  is mass density,  $c_p$  is constant pressure heat capacity per unit mass, and  $T$  is absolute temperature and  $B$  becomes heat generation per time,  $H$ . If  $\rho$  and  $c_p$  are constant in space and time, the one-dimensional energy conservation equation is

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p v_x \frac{\partial T}{\partial x} = k \frac{\partial^2 T}{\partial x^2} + H$$



When considering a stagnant medium (bulk velocity is zero) and there is no energy generation, this simplifies to the heat equation,

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (2.3)$$

Where  $\alpha = k/\rho c_p$  is the thermal diffusivity.

In mass transfer,  $b$  becomes molar concentration of a species  $c_i$  and  $B$  becomes the rate of reaction,  $R$ . For a first order consumption of species  $i$ , with rate constant  $k_i$ , the one-dimensional mass conservation equation is

$$\frac{\partial c_i}{\partial t} + v_x \frac{\partial c_i}{\partial x} = d_i \frac{\partial^2 c_i}{\partial x^2} - k_i c_i \quad (2.4)$$

These are both linear forward parabolic partial differential equations. They are second order in space and first order in time.

To be unique, these equations need boundary conditions. Specifically, they need two boundary conditions in space and one initial condition in time. The spatial boundary conditions generally refer to either a concentration, flux, or both of a given conserved quantity. Defining  $b$  as the conserved quantity,  $\underline{r}_s$  as a position vector of the boundary,  $\underline{n}$  as the vector normal to the surface, and  $t$  as time, three common types are as follows:

Dirichlet:	$b = f(\underline{r}_s, t)$
Neumann:	$\underline{n} \cdot \nabla b = g(\underline{r}_s, t)$
Robin:	$\underline{n} \cdot \nabla b + h_1(\underline{r}_s, t) b = h_0(\underline{r}_s, t)$

Initial conditions take the form

$$b = f(\underline{r}, t_0)$$

where  $\underline{r}$  is any position vector and  $t_0$  is the initial time.

As mentioned above, the generic transport variable will be  $b(t, x; \theta)$ ; here  $\theta$  represents the parameters of the equation. From this point on, the one-dimensional case will be considered with expansion to multiple dimensions for cases of particular interest (where  $d$  replaces  $D$  since it is no longer a matrix quantity). A generic reaction term,  $r(\cdot)$  is also used, giving the default equation

$$\frac{\partial}{\partial t} b + \frac{\partial}{\partial x} v b = \frac{\partial}{\partial x} \left( d \frac{\partial}{\partial x} b \right) + r(b)$$

The simplest case has constant coefficients (i.e.  $\nu$  and  $d$  are invariant with respect to time and position). In such cases it is often advantageous to non-dimensionalize the equation using the Peclet and Damköhler numbers. Note that the bars over variables indicate that they are the dimensionless counterparts of the original variables.

$$\frac{\partial \bar{b}}{\partial \bar{t}} = \frac{\partial^2 \bar{b}}{\partial \bar{x}^2} - Pe \cdot \frac{\partial \bar{b}}{\partial \bar{x}} + Da \cdot \bar{b}.$$

Even this simple equation presents challenges to solve. The basic structure will serve as the base case for many of the concepts developed in this thesis.

For the sake of comparison, consider the Black-Scholes equation used in financial option price modeling

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0$$

which is described in more detail in Chapter 0. Note the similar structure. As will be seen later the techniques developed in this thesis can be applied to these types of problems as well.

Of course the more interesting problems can become much more complicated as well. For example, consider the number distribution equation for a population balance

$$\frac{\partial n(\underline{m})}{\partial t} + \nabla_{\underline{x}} \cdot \underline{\nu} n(\underline{m}) = \frac{\partial I}{\partial \underline{m}} \frac{n(\underline{m})}{n(\underline{m})} + \int \dots \int \beta(\underline{m} - \underline{m}', \underline{m}') n(\underline{m} - \underline{m}') n(\underline{m}') d\underline{m}' + \dots$$

This will be explained more fully in Chapter 9.0 but it is readily apparent that standard solution techniques are unlikely to work on such equations. The two most relevant points about this equation are the integral terms and the dimensionality. The integral terms, discussed in the next chapter, present a departure from normal chemical engineering problems. Most problems only involve phenomena that are influenced by local perturbations; but the integral terms represent effects that can occur over the entire solution domain. The dimensions over which the dependent variable can vary is of an arbitrarily high number. This is because the dependent variables, besides space and time, can also include size and other physical characteristics. Overall this results in more dimensions than the three spatial ones seen in standard problems and adds to the problem's complexity.

The techniques developed in this thesis can and will be applied to many different systems, but it will all start from the basic equation structure mentioned above.

## 2.2 Numerical Solution Methods

In broad terms, the solution of partial (integro)-differential equations occurs in two (generally interrelated) parts: The spatial derivative (and integral) terms can be discretized to give a first order ordinary differential equation. This can then be integrated through time to obtain the variable of interest as function of the temporal and spatial variables.

However, each type of numerical method has its own drawbacks; they must be chosen to meet predetermined criteria. For this thesis, the criteria are physical correctness, computational efficiency, and numerical accuracy. Of course, these factors are often at odds with one another. And overlying all of these is robustness. The systems mentioned in the previous section are fairly disparate in terms of the range of inputs and outputs, but the methods developed should work on both of them; focusing on the underlying structure of the methods in a ground-up approach is the key here.

Partial differential equations (PDEs) are a standard modeling tool. However, the inclusion of integral terms (resulting in partial integro-differential equations, PIDEs) allows for more correct models in many situations. But including them will require forging new ground. While there is a great deal of theory dedicated to PDE solutions and several software packages designed for solving such systems, PIDE solution methods still do not have a common framework. There are many promising approaches, but they are often situation-specific, limiting their overall utility.

For clarity, the one-dimensional, one-species system with constant coefficients is considered here,

$$\frac{\partial}{\partial t} b + v \frac{\partial}{\partial x} b = d \frac{\partial^2}{\partial x^2} b + r(b) \quad (2.5)$$

where  $x$  is the dimension of interest and  $b(t, x)$  is the transported quantity (e.g. mass, heat, momentum). Note that diffusion coefficient is noted with a lowercase  $d$  here to avoid confusion with the diffusion matrix  $D$ .

The technique of discretizing spatial terms first and then integrating over the time variable is known as the method of lines (MOL) approach. The ODE system resulting from the spatial discretization can be written as

$$w'(t) = F(t, w(t)) \quad w(t_0) = w_0 \quad (2.6)$$

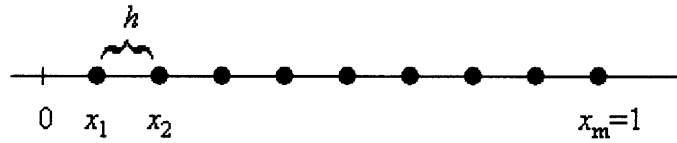
where  $w$  is a vector representing the spatially discretized concentration and  $w_0$  is the initial value and the tick indicates the first derivative in time. Note that  $w$  is a vector here. (Throughout this thesis vectors are not written as bold or underlined to make them easier to read. A note will be made if there is some potential for ambiguity.) It is important to point out that  $b$  represents the true quantity of interest and  $w$  is the spatially discretized approximation. If the system is linear it can be written as

$$w'(t) = Aw(t) + g(t), \quad w(t_0) = w_0 \quad (2.7)$$

with  $A$  some matrix which is generally square and non-singular, with the same dimensions as  $w$ .  $g$  is some source term which is absent in many examples.

### 2.2.1 Spatial Discretization

Spatial discretizations involve approximating the first and second order derivative terms on a grid separated by a distance  $h$ .



For the first derivative term, one simple method is the first-order upwind

$$\frac{\partial b}{\partial x}(x) = \frac{1}{h} b(x) - b(x-h) + O(h). \quad (2.8)$$

Note that this equation is a first order approximation meaning that the error is order  $h$  (see Section 2.2.5). Also note that this approximation requires velocity to be in the positive  $x$ -direction (the point  $x+h$  is used if the velocity is negative). For the basic advection equation,

$$\frac{\partial b}{\partial t} + v \frac{\partial b}{\partial x} = 0$$

with periodic boundary conditions  $b(x \pm 1, t) = b(x, t)$ , the semi-discrete system is

$$w'_j(t) = \frac{v}{h} w_{j-1}(t) - w_j(t), \quad j = 1, 2, \dots, m, \quad w_0(t) = w_m(t).$$

The matrix  $A$  for this method is

$$A = \frac{v}{h} \begin{pmatrix} -1 & & & & 1 \\ 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & 1 & -1 & \\ & & & 1 & 0 \end{pmatrix}.$$

For the second-order central scheme the approximation is

$$\frac{\partial b}{\partial x}(x) = \frac{1}{2h} b(x+h) - b(x-h) + O(h^2). \quad (2.9)$$

The corresponding semi-discrete system is

$$w'_j(t) = \frac{v}{2h} w_{j-1}(t) - w_{j+1}(t), \quad j=1,2,\dots,m, \quad w_0(t) = w_m(t), \quad w_{m+1}(t) = w_1(t)$$

with matrix

$$A = \frac{v}{h} \begin{pmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & \vdots \\ & \ddots & \ddots & \ddots \\ & & 1 & 0 & -1 \\ -1 & & 0 & 1 & 0 \end{pmatrix}.$$

For the diffusion term, a second-order discretization is

$$\frac{\partial^2 b}{\partial x^2}(x) = \frac{1}{h^2} b(x-h) - 2b(x) + b(x+h) + O(h^2). \quad (2.10)$$

For the diffusion problem

$$\frac{\partial b}{\partial t} = d \frac{\partial^2 b}{\partial x^2}$$

with periodic boundary conditions  $b(x \pm 1, t) = b(x, t)$ , the semi-discrete system is

$$w'_j(t) = \frac{d}{h^2} w_{j-1}(t) - 2w_j(t) + w_{j+1}(t), \quad j=1,2,\dots,m, \quad w_0(t) = w_m(t), \quad w_{m+1}(t) = w_1(t)$$

and the matrix is

$$A = \frac{d}{h^2} \begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix}.$$

There are several other higher order schemes that exist, using more grid points to obtain a more accurate approximation, but the basic idea is the same.

### 2.2.2 Mass Conservation

When considering the conservation of mass (or other quantity that can be represented by  $b$ ) it is useful to think in terms of the values midway between the points in space, i.e.  $w_{j-1/2}$  and  $w_{j+1/2}$ . This allows the consideration of a cell centered at the  $j^{\text{th}}$  point (where a “cell” can be one, two, or three dimensions). All of the above forms can be written in this manner, but when both concentration and velocity are variable with position there is the issue of whether to evaluate the velocity with the velocity at the middle points (the flux form) or evaluate it at the standard points (the advective form). In the upwind scheme these two forms are, respectively

$$w'_j(t) = \frac{1}{h} v(x_{j-\frac{1}{2}})w_{j-1}(t) - v(x_{j+\frac{1}{2}})w_j(t) , \quad \forall v(x)$$

$$w'_j(t) = \begin{cases} \frac{1}{h} v(x_j) w_{j-1}(t) - w_j(t) , & \text{if } v(x_j) \geq 0 \\ \frac{1}{h} v(x_j) w_j(t) - w_{j+1}(t) , & \text{if } v(x_j) \leq 0 \end{cases} .$$

Both forms have different properties, but only the flux form conserves mass. This is so because the flux form effectively balances the fluxes into and out of the  $j^{\text{th}}$  cell, maintaining the requirement

$$M(t) = \int_0^1 b(x,t) dx = \text{constant} .$$

This can be tested by evaluating the  $m$ -length vector at each time step and ensuring that

$$h \sum_{j=1}^m w_j(t) = \text{constant} .$$

Throughout this thesis, mass conservation will be considered an essential requirement.

### 2.2.3 Time Stepping Methods

With systems of equations discretized and in the form of (2.6) a method must be used to evaluate the solution through time. The basic concept is similar to the spatial discretizations: evaluate the derivative using information adjacent to the point of interest.

Time is divided into segments  $\tau$  units of time apart. The numerical approximations  $w_n$  estimate the exact values,  $w(t_n)$  at the points  $t_n = n\tau$ . Methods can be categorized as either implicit or explicit, depending on whether they require an evaluation of the function,  $F$ , at the point of interest or not, respectively. For a gross generalization, explicit methods are less expensive to solve at each time step whereas implicit methods are more useful for stiff systems (see below). The simplest method is the (forward) Euler method,

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

This is a first-order explicit method. The notion of order can be understood by considering the Taylor series expansion,

$$w(t_{n+1}) - w(t_n) = \tau w'(t_n) + O(\tau^2).$$

The approximation is accurate up to  $\tau^1$  so it is first order. (See also Section 2.2.5.)

The simplest implicit method is the backward Euler method,

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

which is also first order. Note that determining  $w_{n+1}$  in general involves solving a potentially nonlinear system of equations.

### 2.2.4 Convergence

The only useful methods are those that converge to the correct solution. This requires both consistency and stability.

Consistency essentially means that the error of a given method with go to zero as the time step size goes to zero. If  $\rho$  is defined as the local truncation error (see Section 2.2.5), consistency requires that

$$\lim_{\substack{\tau \rightarrow 0 \\ t=t_0+n\tau}} \frac{\rho_n}{\tau} = 0.$$

The limit requires that  $t = t_0 + n\tau$  remain constant, i.e. it must be that  $n \rightarrow \infty$ . Every solution method must be formulated in such a way that it fulfills this criterion. For more detail, see [Lambert, 1991].

Stability essentially means that the overall global error will not “blow up.” Consider the ODE problem

$$w'(t) = F(t, w(t)) \quad w(t_0) = w_0.$$

Also, consider  $\|\cdot\|$  to be some type of vector norm. The Lipschitz condition requires that

$$\|F(t, \tilde{w}) - F(t, w)\| \leq L \|\tilde{w} - w\|$$

for all  $t, \tilde{w}, w$  on some well defined space,  $D$ ,

$$D = (t, w) \in \mathbb{R} \times \mathbb{R}^m : 0 \leq t \leq T, \|w - w_0\| \leq K_0, \quad K_0 > 0.$$

While meeting the Lipschitz condition ensures stability, in practice the solution may require prohibitively small timesteps.

### 2.2.5 Error Estimation

Quantifying the error is important both in testing the accuracy of a method and optimizing the step sizes. For the purposes of comparison to a known solution there are many choices. In this thesis, three types are employed. The first two types are termed the absolute and relative errors. Note that the names of these error types given here are by no means standard, but the underlying method can be defined in terms of vector norms, where the  $p$ -norm is defined by

$$p\text{-norm} \equiv \left( \sum_{i=1}^m |w_i|^p \right)^{1/p} \equiv \|w\|_p$$

where  $w$  is an  $m$ -length vector. The errors are then defined as

$$\begin{aligned} \text{absolute error} &\equiv \frac{1}{m} \|w - w_{\text{exact}}\|_1 \\ \text{relative error} &\equiv \frac{1}{m} \frac{\|w - w_{\text{exact}}\|_2}{\|w_{\text{exact}}\|_2} \end{aligned}$$

The other type of error used herein is the peak error, which is useful since peaks are typically difficult to reproduce numerically. It is defined as the difference between the highest point of the exact solution and the corresponding approximate point, divided by the exact value.

A more rigorous definition of order and error is as follows. If an exact solution can be represented as a polynomial, the order of a numerical method is essentially the highest degree of polynomial for which the method obtains the exact answer. For a simple example, consider the general ODE system

$$w' = F(t, w), \quad w(t_0) = w_0, \quad F : \mathbb{R} \times \mathbb{R}^m \rightarrow \mathbb{R}^m \quad (2.11)$$

and define  $w(t_n)$  as an  $m$ -length vector representing the exact solution at time  $t_n$  and  $w_n$  as the corresponding approximate solution.

Note that before any numerical method can be attempted, the system itself must meet certain general criteria, the most important of which being the Lipschitz condition, as discussed above.



Every time step  $\tau$  results in some amount of error. For example, the explicit Euler scheme gives

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

with  $\rho_n$  as the local truncation error. By comparing this to a Taylor series expansion

$$w(t_{n+1}) = w(t_n) + \tau w'(t_n) + \frac{1}{2} \tau^2 w''(t_n) + O(\tau^2),$$

the error is clearly

$$\rho_n = \frac{1}{2} \tau w''(t_n) + O(\tau^2).$$

Alternately, if  $\|\cdot\|$  is some vector norm, it can be shown using the mean value theorem that the local error can be defined by

$$\|\rho_n\| \leq \frac{1}{2} \tau \max_{t_n \leq s \leq t_{n+1}} \|w''(s)\|.$$

The error term can be more complex, but often it is the order of the method that is of greater interest. In general, if the exact solution and the approximate solution differ only by terms of order greater than  $\tau^p$  then the method is considered to be of order  $p$ . So the explicit Euler is clearly a first order method. Usually, the higher the order, the greater the accuracy, but this is by no means rigorously true.

Thus far only the error between steps has been considered. The global error up to the  $n^{\text{th}}$  time step

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

is what determines the true accuracy of the solution. This error can be expressed in terms of the Lipschitz constant by

$$\|\varepsilon_n\| \leq e^{L t_n} \|\varepsilon_0\| + \frac{1}{L} e^{L t_n} - 1 \max_{0 \leq j < n} \|\rho_j\|.$$

However, this bound can often be so large as to be meaningless; in practical error control techniques it is the local truncation error that is the basis for estimation.

Error control is an important part of any algorithm. The ideal way to measure error is to compare the solution from a given approximation with the exact solution. But of course, this is impossible when solving systems where the exact solution is unknown, which are the only solutions of interest. As such, various approximations of error are necessary.

One such method is Richardson extrapolation. The basic idea is to perform an evaluation with two different time steps and use the difference between them as an estimate of the local truncation error.

Consider,  $w_{n+1}$  to be an estimate with step size  $\tau$  and  $z_{n+1}$  to be an estimate with step size  $2\tau$ . Then, for some constant  $C$ , we have

$$\begin{aligned}w(t_{n+1}) &= w_{n+1} + C\tau^p + O(\tau^{p+1}) \\z(t_{n+1}) &= z_{n+1} + C(2\tau)^p + O(\tau^{p+1})\end{aligned}$$

Note that the two  $O(\tau^{p+1})$  terms are different. Solving for the two unknowns, the exact solution and  $C$  gives

$$\begin{aligned}w(t_{n+1}) &= \frac{2^p w_{n+1} - z_{n+1}}{2^p - 1} + O(\tau^{p+1}) \\C &= \frac{w_{n+1} - z_{n+1}}{(2\tau)^p - \tau^p} + O(\tau^{p+1})\end{aligned}$$

So the local truncation error is

$$\rho_n = w(t_{n+1}) - w_{n+1} = \frac{w_{n+1} - z_{n+1}}{2^{p+1} - 1}.$$

This error estimate is accurate to order  $\tau^p$ , which is the same as the order of the method. In practice more advanced error estimation techniques can be employed, but the basic concepts are similar.

Richardson extrapolation (or some similar technique) serves as the basis for most error correction methods used in numerical solution programs. The most important feature is that they do not require any knowledge of the true solution. They simply compare the error between two different estimates of the solution.

These error estimation techniques do not give a true estimate of the total error but rather a practical value that can be used within an algorithm itself. In general, the most important goal of the error estimation is to control the time step size when variable time steps are used.

### 2.2.6 Stability Domain

The stability provides a region where the solution will converge in terms of the time step size. Consider the scalar test equation

$$w'(t) = \lambda w(t), \quad w(0) = 1$$

with solution

$$w(t) = e^{\lambda t}.$$

Any one-step method (a method that estimates the next value  $(n+1)$  from function evaluations at the previous point and the point itself) can be written in terms of a recursion involving a stability function  $R(z)$  where for the simple scalar case  $z = \tau\lambda$ , and we have

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

Since this recursion is repeated  $n$  times, stability requires that

$$|R(z)| \leq 1.$$

Formally, this is

$$D_{\text{Euler}} = \{z \in \mathbb{C} : |1+z| < 1\}. \quad (2.12)$$

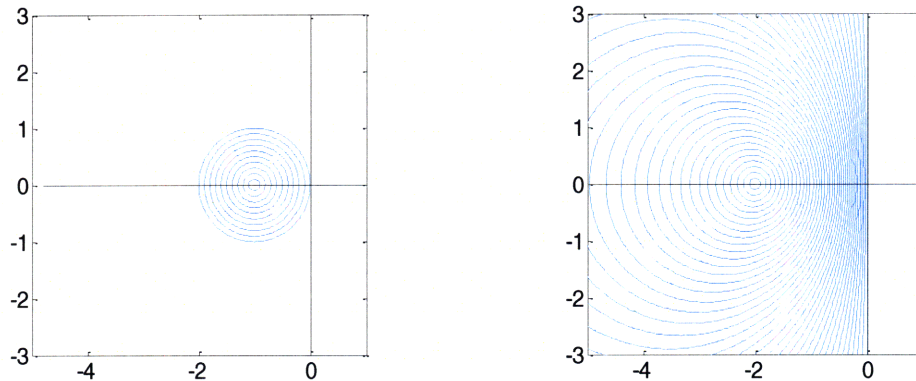
The values for  $z$  for which this inequality holds are mapped out on the complex plane. For two examples, consider the explicit Euler method and the implicit trapezoid rule (also called the Crank-Nicolson method)

$$w_{n+1} = w_n + \tau F(t_n, w_n) \quad \text{and} \quad w_{n+1} = w_n + \frac{1}{2} \tau F(t_n, w_n) + \frac{1}{2} \tau F(t_{n+1}, w_{n+1})$$

with respective stability functions

$$R(z) = 1+z \quad \text{and} \quad R(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}.$$

The stability regions for these methods are shown in the following figure.



**Figure 2.1: Stability Domains for Explicit Euler (left) and Trapezoid Rule (right)**

Note that for the trapezoid rule, the stability region is the entire left hand side of the complex plane. Methods that contain this region are referred to as  $A$ -stable. From a practical standpoint, this means that there is no limitation on the time step imposed by the stability requirement. More types of stability regions are discussed below.

For values of  $z$  that fall outside of a method's stability region, the solution will diverge (often strongly) as the time steps progress, even if the ODE satisfies the Lipschitz condition. For an example, see the next section.

The stability domains remain the same for systems of equations as well. For the linear system

$$w'(t) = Aw(t) + g(t), \quad w(t_0) = w_0,$$

the same criterion for the stability function  $R(z)$  applies, but now with  $z = \tau A$ . If  $A$  is a normal matrix, the stability requirement amounts to setting  $z = \tau \lambda_i$  for all  $i$ , where  $\lambda_i$  represents the  $i^{\text{th}}$  eigenvalue of  $A$ . Nonlinear systems can be analyzed similarly with the Jacobian of  $F(t, w(t))$  replacing  $A$ . However, the same guarantees of stability often do not apply and any stability analysis with non-linear systems must be done with care.

There are several different types of stability beyond those discussed above. The most desirable type is clearly one with a stability domain that encompasses the entire left had side of the complex plane. As mentioned before, this is called  $A$ -stability and is formally defined by

$$D \supseteq \{ z \mid \text{Re}(z) < 0 \} .$$

Since the eigenvalues must be negative for convergence of the system in general,  $A$ -stability imposes no restriction on  $\tau$ . Of course, this stability requirement also places the most restriction on the method. (As can be seen in above, Euler's method is not  $A$ -stable.)

Another type of stability is  $A(\alpha)$ -stability, which is defined by

$$D \supseteq \{ z \mid -\alpha < \pi - \arg(z) < \alpha, \quad \alpha \in (0, \pi/2) \}.$$

To restrict the domain to only the real negative axis,  $A_0$ -stability is used:

$$D \supseteq \{ z \mid \operatorname{Re}(z) < 0, \operatorname{Im}(z) = 0 \}.$$

When the eigenvalues responsible for the slower transients are clustered close to the origin, stiff-stability may be useful,

$$D \supseteq \{ z \mid \operatorname{Re}(z) < -a \cup \{ z \mid -a \leq \operatorname{Re}(z) < 0, -c \leq \operatorname{Im}(z) \leq c \}, \quad a, c \in \mathbb{R}^+ \}.$$

The following figure shows these stability domains.

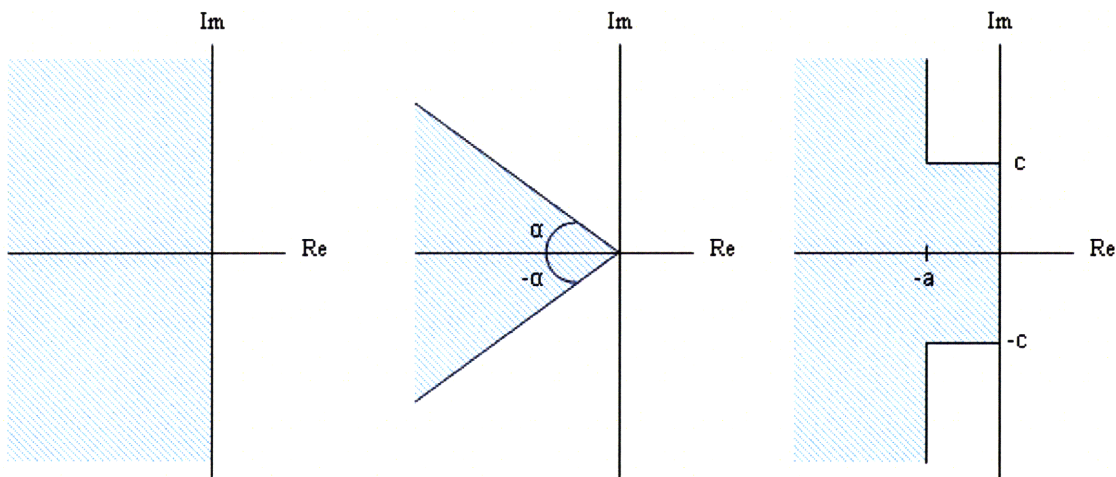


Figure 2.2: A-Stability;  $A(\alpha)$ -Stability; Stiff Stability

Strong A-stability requires that for a rational function,  $R(z)$ ,  $\lim_{z \rightarrow \infty} |R(z)| < 1$ . L-stability is an even more restrictive type that requires  $\lim_{z \rightarrow \infty} |R(z)| = 0$ . Stability on the imaginary axis,  $\lim_{z \rightarrow \infty} |R(iz)| \leq 1$  is known as I-stability.

There are several other types of stability that are more specific to different methods. For example implicit Runge-Kutta methods (see Section 2.2.8) require that, for some perturbation of the system, the condition

$$\|\tilde{w}_{n+1} - w_{n+1}\| \leq \|\tilde{w}_n - w_n\|$$

is met. This applies to nonlinear methods. G-stability is similar but applies to one-leg methods.

Not all authors use exactly the same name for all of these stability types, but the basic concepts allow for a classification of numerical methods into categories based on these definitions.

The key point of all these stability domains is allowing the largest time step possible for the given structure of the problem. A special challenge arises in the vector case when there is a large spread in the eigenvalues. This leads to the concept of stiffness.

### 2.2.7 Stiffness

Stiffness manifests itself by forcing a numerical method to use a very small time step despite the relative smoothness of the solution. A good definition is the following:

If a numerical method with a finite region of absolute stability, applied to a system with any initial conditions, is forced to use in a certain interval of integration a step length which is excessively small in relation to the smoothness of the exact solution in that interval, then the system is said to be stiff in that interval.

[Lambert, 1991]

It is somewhat unsatisfactory that there exists no rigorous mathematical definition to determine if a system is stiff, but there are certain features that can be indicative of stiffness. These can often enable the prediction of a stiff system *ab initio*.

Consider a general, linear, constant coefficient initial value problem

$$w' = Aw + g(t), \quad w(t_0) = w_0 \quad (2.13)$$

where, as before,  $w$  is an  $m$ -length vector,  $A$  is an  $m \times m$  matrix, and  $g$  is some function of  $t$ . Define  $w_p(t)$  as the particular integral and  $w_c(t)$  as the complementary function of (2.13). In general, this complementary solution has the form

$$w_c(t) = \sum_{i=1}^m k_i \exp(\lambda_i t) r_i$$

where  $k_i$  are arbitrary constants,  $\lambda_i$  are eigenvalues of  $A$  and  $r_i$  are the corresponding eigenvectors. The solution to (2.13) is then

$$w(t) = w_c(t) + w_p(t) = \sum_{i=1}^m k_i \exp(\lambda_i t) r_i + w_p(t).$$

If all of the eigenvalues have negative real parts, the solution will go to  $w_p$  as  $t \rightarrow \infty$ , so this can be thought of as a steady state solution. The decay of the transient portion is determined by the eigenvalues. Large negative eigenvalues will decay quickly, while smaller values will persist as time progresses. Consider the  $|\operatorname{Re} \lambda|_{\max}$  and  $|\operatorname{Re} \lambda|_{\min}$  as the eigenvalues of the largest and smallest real parts, respectively. A small value for  $|\operatorname{Re} \lambda|_{\min}$  will cause a large number integration steps to be taken until its effects are negligible. However, the stability region is determined by the magnitude of the eigenvalues, i.e. we must have  $\tau \lambda_i \in D_{\text{Stable}}$ . Therefore, a large  $|\operatorname{Re} \lambda|_{\max}$  may require a very small  $\tau$  depending on the size of the stability domain. This naturally gives rise to the stiffness ratio

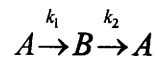
$$\text{SR} = \frac{|\operatorname{Re} \lambda|_{\max}}{|\operatorname{Re} \lambda|_{\min}}$$

as a potential indicator of stiffness. It is neither necessary nor sufficient for a system to exhibit stiff behavior, but it provides an indication that stiffness may be an issue.

Another way to consider the cause of stiffness is to consider the fact that any numerical method has some error, and therefore relies on function evaluations of the integral curves surrounding the true solution. If the function is evaluated on a nearby curve with a gradient that differs significantly, it is easy to see how errors could quickly accumulate. In such a situation, only by taking a large number of time steps is it possible to remain faithful to the real solution.

Both of these ways of considering stiffness obviate the superiority of implicit methods in their solution. In the first case, they generally have larger stability domains, allowing for a larger value of  $\tau$ . In the second case, implicit methods do not rely entirely on previous evaluations of the function. By using evaluations at the new solution point as well, the error of the widely different gradient can be mitigated more easily.

It's worth pointing out here that most systems of interest in engineering exhibit stiffness to some extent. This can occur due to the physics of the problem having parameters that are far apart or from the spatial discretization and desired accuracy. For a simple example of stiffness that arises purely from the physics of the problem, consider the chemical reaction



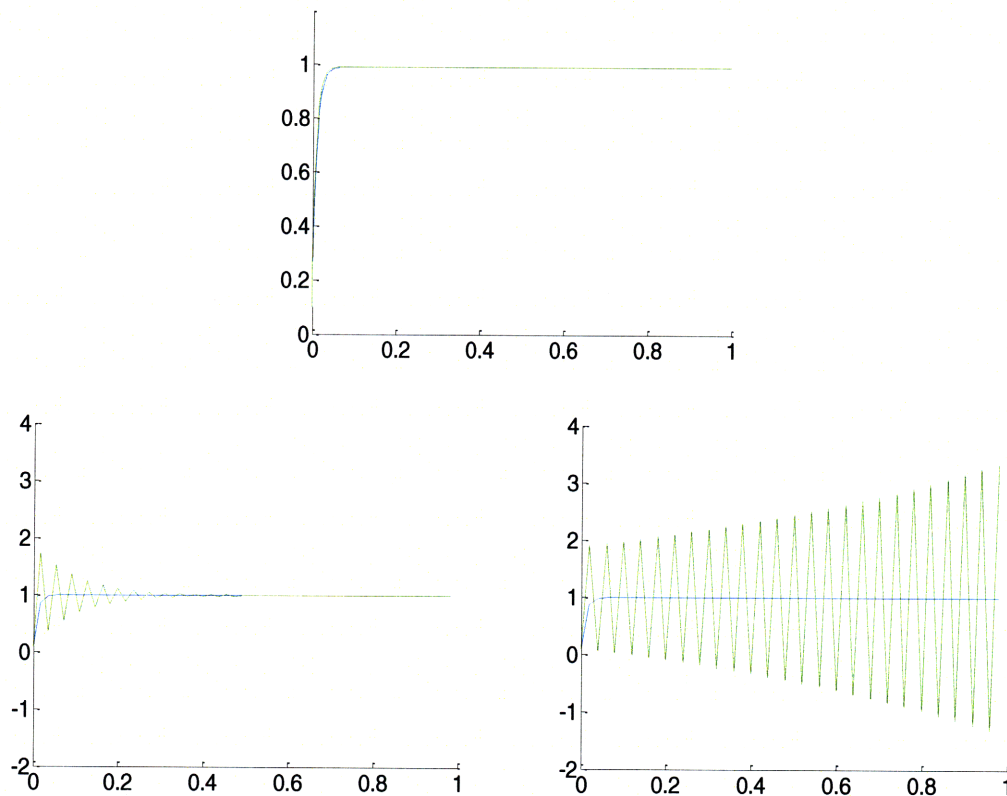
which gives the ODE system

$$\begin{aligned} w_1' &= -k_1 w_1(t) + k_2 w_2(t) \\ w_2' &= k_1 w_1(t) - k_2 w_2(t) \end{aligned}$$

The Jacobian matrix of this system is

$$A = \begin{pmatrix} -k_1 & k_2 \\ k_1 & -k_2 \end{pmatrix}.$$

If  $k_1$  is 1 and  $k_2$  is 100, the eigenvalues,  $\lambda$ , are -101 and 0, which means the stiffness ratio is effectively infinite. The real solution to this system is two well-behaved curves. However, when the explicit Euler method is used, the time steps,  $\tau$ , need to be quite small to obtain a useful solution. For large enough time steps, the solution actually diverges completely.



**Figure 2.3: Instability of Explicit Euler Method Solutions, Concentration vs. Time.**

The analytical solution is dashed, the approximate solution is solid. The time steps,  $\tau$ , are 1/100, 1/55, 1/50, respectively, while all other conditions are the same

The divergence can be understood when considering the stability domain of the explicit Euler method (see Figure 2.1). Recall that the  $\tau\lambda$  needs to be in this region for all of the eigenvalues. Clearly, the zero eigenvalue is acceptable for any  $\tau$ , but the larger-magnitude eigenvalue needs a value of  $\tau \leq -2/-101 = 1/50.5$  to remain stable. Implicit methods such as the trapezoid rule attain stable results for any time step and can achieve good accuracy with larger time steps than corresponding explicit methods.



In terms of spatial discretizations, stiffness is almost always an issue. Consider the heat equation

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2} \quad \text{with } w(t, 0) = 0, w(t, 1) = 0, w(0, x) = 1.$$

Using central finite differences on the second derivative to construct a grid of  $N$  points yields the following equation:

$$w' = -h^{-2} A w$$

where  $h$  is the discretization size in the spatial grid and  $A$  is an  $N \times N$  matrix defined by

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

The eigenvalues and eigenvectors of  $A$  must satisfy

$$A r_k = \lambda_k r_k \tag{2.14}$$

Also, noting that the second derivative operator is an eigenfunction, we have

$$-\frac{d^2}{dx^2} r(x) = \lambda r(x) \quad \text{with } r(0) = 0, r(1) = 0.$$

The solution to this differential equation yields

$$r(x) = \sin(k\pi x) \quad \text{with } \lambda = \pi^2 k^2, k = 1, 2, \dots$$

since  $A$  is just a discretization of the operator  $-\frac{d^2}{dx^2}$ , the elements of the eigenvectors can be represented as follows. Note that the subscripts indicate the  $j^{\text{th}}$  element of the  $k^{\text{th}}$  eigenvector and  $x_j = j \cdot h$  and  $h = 1/(N+1)$ .

$$r_{k,j} = \sin\left(\frac{k\pi j}{N+1}\right) \quad j = 1, 2, \dots, N.$$

(2.14) can then be written as

$$-r_{k,j-1} + 2r_{k,j} - r_{k,j+1} = \lambda_k r_{k,j}$$

and after substitution as

$$-\sin\left(\frac{k\pi(j-1)}{N+1}\right) + 2\sin\left(\frac{k\pi j}{N+1}\right) - \sin\left(\frac{k\pi(j+1)}{N+1}\right) = \lambda_k \sin\left(\frac{k\pi j}{N+1}\right).$$

Now define

$$\alpha = \left(\frac{k\pi j}{N+1}\right) \text{ and } \beta = \left(\frac{k\pi}{N+1}\right)$$

to obtain

$$-\sin \alpha - \beta + 2\sin \alpha - \sin \alpha + \beta = \lambda_k \sin \alpha .$$

Using the trigonometric identity  $\sin \alpha \pm \beta = \sin \alpha \cos \beta \pm \cos \alpha \sin \beta$  and dividing by  $\sin(\alpha)$  gives

$$-\cos \beta + \sin \beta + 2 - \cos \beta + \sin \beta = 2 \frac{1 - \cos \beta}{\sin \alpha} = \lambda_k .$$

The trigonometric identity  $\cos 2\theta = 1 - \sin^2 \theta$  results in

$$\lambda_k = 4 \sin^2 \left( \frac{k\pi}{2(N+1)} \right).$$

The largest eigenvalue clearly occurs for  $k = N$  and the smallest for  $k = 1$ ,

$$|\operatorname{Re} \lambda|_{\max} = 4 \sin^2 \left( \frac{N\pi}{2(N+1)} \right) \approx 4 \sin^2 \left( \frac{\pi}{2} \right) = 4$$

$$|\operatorname{Re} \lambda|_{\min} = 4 \sin^2 \left( \frac{\pi}{2(N+1)} \right) \approx 4 \left( \frac{\pi}{2(N+1)} \right)^2 \approx \frac{\pi^2}{N^2}$$

So the stiffness ratio is

$$\text{SR} \approx \frac{4N^2}{\pi^2}$$

indicating that systems with a large number of grid points are likely to be quite stiff. This becomes increasingly problematic as greater accuracy is required. This is just one area in which the tradeoff between accuracy and solution speed is readily apparent.

### 2.2.8 Runge Kutta Methods

Thus far it seems that implicit methods have some advantages in terms of stability and explicit methods in terms of speed. However little has been said about accuracy of solution methods. Indeed, the two mentioned above are only of first order accuracy. It is therefore worthwhile to consider more advanced methods. These methods not only change the order of accuracy but also the solution domain.

Runge-Kutta time integration methods are one of the most frequently used tools in evaluating ODEs. As their basic structure is used in many portions of this thesis, a brief overview is worthwhile.

The basic idea behind Runge-Kutta methods is to gather information about the family of integral curves that surround the unique solution to an initial value problem. The unique solution (for an  $m$ -length vector) is represented by a single solution curve in  $\mathbb{R}^{m+1}$ . But truncation and round-off errors will result in adjacent integral curves affecting any numerical solution. By considering the effects of these curves, a more accurate solution can often be achieved. The downside of Runge-Kutta methods is that the error structure can become somewhat complex.

Runge-Kutta methods are one-step methods, meaning that a numerical estimation depends only on the value of the one previous numerical estimation. Each step uses values from  $s$  stages, which generally are not on the solution curve, to estimate the value of the next step.

Consider the general initial value problem of the form

$$w' = F(t, w), \quad w(t_0) = w_0, \quad F: \mathbb{R} \times \mathbb{R}^m \rightarrow \mathbb{R}^m \quad (2.15)$$

where  $w(t)$  is an  $m$ -length vector and  $w'$  is its first derivative with respect to  $t$ . The general Runge-Kutta solution has the form

$$w_{n+1} = w_n + \tau \sum_{i=1}^s b_i k_i \quad (2.16)$$

$$k_i = F \left( t_n + c_i \tau, w_n + \tau \sum_{j=1}^s a_{ij} k_j \right)$$

where  $w_n$  is the  $n^{\text{th}}$  step,  $\tau$  is the step length, and the coefficients satisfy the following conditions:

$$c_i = \sum_{j=1}^s a_{ij}, \quad i = 1, 2, \dots, s$$

$$1 = \sum_{i=1}^s b_i$$

A Butcher array can be used to keep track of the coefficients. It has the form

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & & a_{2s} \\ \vdots & \vdots & & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

which can be abbreviated as

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

Note that in (2.16) if the coefficients  $a_{ij}$  are zero for all  $j \geq i$ , then each  $k_i$  can be solved explicitly with respect to previously calculated  $k_j$ 's. For such methods the Butcher array is strictly lower triangular. When this is the case, the system is said to be an explicit Runge-Kutta method. If some of the diagonal elements are non-zero the method is semi-implicit, and if there are non-zero elements in the upper triangle the method is implicit.

The coefficients of a Runge-Kutta method cannot be chosen arbitrarily. For scalar problems, the basic derivation consists of matching up terms of the expansion of the Runge-Kutta method with a Taylor series expansion of the exact solution. This, however, can become quite tedious for orders higher than two. Also, there is a loss in generality in assuming that a scalar initial value problem is autonomous.

To determine the coefficients, the methods of Butcher [Butcher, 2003] are the best resource. That work also determines the order conditions for Runge-Kutta methods. The order conditions establish the maximum order that can be attained for a method with a given number of stages. The most important result is that for order greater than four, the number of stages must be greater than the order.

One way to think about (some) Runge-Kutta methods is in terms of interpolation polynomials. Consider  $P$  to be a polynomial of degree  $s$  with real coefficients and distinct collocation points  $\{t_n + c_i\tau, i = 1, 2, \dots, s\}$ . Determine  $P$  such that it satisfies

$$\begin{aligned}
P(t_n) &= w_n \\
P'(t_n + c_i\tau) &= F(t_n + c_i\tau), \quad i = 1, 2, \dots, s.
\end{aligned} \tag{2.17}$$

This can be seen to be in the form of a Lagrange polynomial by defining  $k_i = P'(t_n + c_i\tau)$ ,  $i = 1, 2, \dots, s$  and  $t = t_n + x\tau$  and writing

$$\begin{aligned}
P'(t_n + x\tau) &= \sum_{j=1}^s L_j(x)k_j \quad \text{where} \\
L_j(x) &= \prod_{\substack{i=1 \\ i \neq j}}^s \frac{x - c_i}{c_j - c_i}
\end{aligned}$$

Integrate (2.17) with respect to  $t$  from  $t = t_n$  to  $t = t_n + c_i\tau$  to get

$$P(t_n + c_i\tau) - P(t_n) = \tau \sum_{j=1}^s \int_0^{c_i} L_j(x) dx k_j, \quad i = 1, 2, \dots, s$$

and integrate (2.17) from  $t = t_n$  to  $t = t_n + \tau = t_{n+1}$  to get

$$P(t_n + \tau) - P(t_n) = \tau \sum_{j=1}^s \int_0^1 L_j(x) dx k_j.$$

Now define

$$a_{ij} = \int_0^{c_i} L_j(x) dx, \quad i = 1, 2, \dots, s \quad \text{and} \quad b_j = \int_0^1 L_j(x) dx. \tag{2.18}$$

This gives

$$\begin{aligned}
k_i &= P'(t_n + c_i\tau) = F(t_n + c_i\tau) = F\left(t_n + c_i\tau, w_n + \tau \sum_{j=1}^s a_{ij}k_j\right) \\
w_{n+1} - w_n &= P(t_n + \tau) - P(t_n) = \tau \sum_{i=1}^s b_i k_i
\end{aligned}$$

which is exactly (2.16).

Not all Runge-Kutta schemes satisfy (2.18); those that do are referred to as collocation methods. The determination of Runge-Kutta coefficients via orthogonal polynomials is the basis for the explicit methods with enhanced stability domains (see Section 5.1).

## 2.2.9 Linear Multistep Methods

In contrast to Runge-Kutta methods which use only the values at the previous time step (and possibly the current one), multistep methods can use many values from both past up to the current time step. They are usually expressed as

$$\sum_{j=0}^k \alpha_j w_{n+j} = \tau \sum_{j=0}^k \beta_j F(t_{n+j}, w_{n+j}). \quad (2.19)$$

which indicates that there are  $k$  past values used in the computation of the current value. If the coefficient  $\beta_k$  is zero then the method is explicit.

The procedure for designing a method relies on choosing the coefficients to obtain as high an order as possible. This can be demonstrated by inserting an exact value into all of the  $w_k$  values and determining how the error term,  $\rho_{n+k-1}$  is propagated. Solving this linear system gives the standard order conditions to obtain order  $p$ ,

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

as long as  $\tau$  is constant.

The stability of multistep methods depends on considering the characteristic polynomial defined by

$$\pi(\zeta) = \sum_{j=0}^k \gamma_j \zeta^j$$

where the  $\gamma$ s are determined by the scalar linear recursion formula

$$\sum_{j=0}^k \gamma_j w_{n+j} = 0.$$

If  $\zeta_1, \zeta_2, \dots, \zeta_k$  refer to the  $k$  zeros of the characteristic polynomial, then the root condition is satisfied if

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

This effectively bounds the solution for  $n \rightarrow \infty$  steps. For the general formula (2.19) first test for stability is called zero-stability which tests for the simple case when  $F=0$ . Then the characteristic polynomial is

$$\pi(\zeta) = \sum_{j=0}^k \alpha_j \zeta^j$$

and the root condition for this must be satisfied for any method (it holds trivially for all one-step methods). After this the root condition for the general characteristic polynomial can be considered.

There are two primary challenges of multistep methods. First of all, when taking the first few time steps, there clearly need to be  $k-1$  values available in addition to the initial conditions for the method to begin working. Additionally, allowing for variable time step sizes will require changing the  $\alpha$  and  $\beta$  coefficients at each time step.

Common multistep methods can be grouped into two broad categories. Adams methods define the  $\alpha$  coefficients by

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

and choose the  $\beta$  coefficients to maximize the order. In contrast backward differentiation formulae (BDF) define the  $\beta$  coefficients by

$$\beta_k = 1, \quad \beta_j = 0, \quad j = 0 \dots k-1$$

and choose the  $\alpha$  coefficients to maximize the order. There are many other methods that do not fit into these categories and some are even defined differently than (2.19), but the same underlying theory applies for the most part.

### 2.2.10 Fourier Decompositions

There is one more topic that requires a brief overview in this section. Fourier decompositions are useful not only in solving many PDE systems but also in determining where the difficulties will arise in numerical solution methods. Define the Fourier series as

$$f(x) = \sum_{k=-\infty}^{\infty} \alpha_k e^{2\pi i k x}$$

with Fourier modes,  $\varphi_k(x)$ , defined by

$$\varphi_k(x) = e^{2\pi i k x}$$

and Fourier coefficients defined by

$$\alpha_k = \int_0^1 f(x) e^{-2\pi i k x} dx.$$

Since (almost) any function can be represented as an infinite sum of Fourier modes it often suffices to look at just one mode and then expand the results to the general case. For example, consider the advection equation with an initial condition defined by a single Fourier mode,

$$\frac{\partial}{\partial t} b(x, t) = -v \frac{\partial}{\partial x} b(x, t), \quad b(x, 0) = \varphi_k(x).$$

By making the assumption up front that this PDE is separable it can be solved to give

$$b(x, t) = e^{-2\pi i k v t} \varphi_k(x) = \varphi_k(x - vt).$$

Similarly, the diffusion PDE,

$$\frac{\partial}{\partial t} b(x, t) = d \frac{\partial^2}{\partial x^2} b(x, t), \quad b(x, 0) = \varphi_k(x)$$

can be solved to give

$$b(x, t) = e^{-4\pi^2 k^2 d t} \varphi_k(x).$$

Note that the Fourier decomposition has revealed the key characteristics of each equation. Thinking of the Fourier mode as a wave, it is seen that pure advection leads to a shift in space proportional to velocity and that pure diffusion leads to dissipation of the wave. Both of these results agree with the physical interpretation of the equations. And this can be extended to the general case by considering an infinite series of Fourier modes.

### 2.2.11 Boundary Conditions

Boundary conditions (BCs) are often overlooked in the development of numerical techniques. However, they often present significant challenges when attempting to design and implement a solution method.

The theory that is developed regarding the order of accuracy, convergence, and other properties often breaks down when actual boundary conditions are implemented. Most of the time the reduction is not too great but it is always important to realize that there is the potential for unexpected results when applying boundary conditions. The key, as always, is to analyze a given system of interest thoroughly.

A few commonly used boundary conditions are discussed in terms of their implementation in standard finite difference discretizations. As in the previous sections,  $w$  is the vector of the discretized values of the quantities of interest,  $j$  is a given grid point, and  $A$  is the matrix that contains the coefficients that approximate the functions. First off are the so called periodic boundary conditions first mentioned in Section 2.2.1. They



amount to setting the value at one end to the solution domain to be equivalent to the value at the other end. Numerically this results in applying the coefficients for a given discretization that would extend beyond the one end point to be at the other end point. To clarify this, consider a matrix that results from a discretization that spans three grid points with coefficients  $a_{-1}$ ,  $a_0$ , and  $a_1$ . In the periodic BC case the  $A$  matrix then has the form

$$A = \begin{pmatrix} a_0 & a_1 & & & a_{-1} \\ a_{-1} & a_0 & a_1 & & \\ & \ddots & \ddots & & \\ & & a_{-1} & a_0 & a_1 \\ a_1 & & & a_{-1} & a_0 \end{pmatrix}.$$

The important feature of this matrix is that it is now circulant. This fact allows for much of the theory regarding the standard discretizations to be developed and allows for the exact characterization of the eigenvalues by Fourier analysis.

Of course, these periodic boundary conditions do not correspond to any actual physical situations. In real situations the most common types of boundary conditions are Diriclet which specifies the value at the boundaries and Neumann which specifies the value of the gradient at the boundaries. One other type of BC, the open boundary condition, becomes important in numerical methods. If the function being approximated is only a first derivative then only one spatial BC can be specified. The other boundary cannot be specified which presents a challenge since the numerical approximation must have some value at the other point.

The boundary conditions can be specified in some vector,  $g(w)$ , that is added to the product of  $w$  and  $A$ . The boundary conditions are denoted  $BC_l$  and  $BC_u$  for the lower and upper boundary conditions. To keep things simple consider a discretization that spans three grid points with coefficients specified as above. The grid runs from 1 to  $m$  so the boundary values correspond to points  $w_0$  and  $w_{m+1}$ . The Diriclet conditions are the easiest to handle since they specify these two values,

$$w_0 = BC_l \quad \text{and} \quad w_{m+1} = BC_u$$

The equation for the  $m^{\text{th}}$  discrete approximation is

$$a_{-1}w_{m-1} + a_0w_m + a_1w_{m+1}.$$

Since the  $A$  matrix is only  $m \times m$  the rightmost portion must be handled in the boundary vector,  $g$ . The value for the 1<sup>st</sup> row can be handled similarly finally giving us

$$g_1 = a_{-1}BC_l \quad \text{and} \quad g_m = a_1BC_u$$

For Neumann conditions the situation is a bit more challenging as we need to consider how to handle the gradient discretely. In this case the boundary values must be specified

indirectly by the choice of discretization of the 1<sup>st</sup> derivative. Take the upper boundary and consider the upwind discretization,

$$\frac{1}{h} w_{m+1} - w_m = BC_u$$

where  $h$  is the grid spacing. Rearranging this gives a value for  $w_{m+1}$  which can then be multiplied by the  $a_1$  coefficient. Handling the lower bound similarly we have

$$g_1 = -a_{-1} hBC_l - w_1 \quad \text{and} \quad g_m = a_1 hBC_u - w_m .$$

The open BC can be considered as extrapolation to a virtual point. We basically want the value at  $w_{m+1}$  to be the same as the values preceding it. The basic extrapolation is

$$w_{m+1} = \theta w_m + (1 - \theta) w_{m-1}$$

where  $\theta$  can be 1 or 2. The simplest way to incorporate this is to treat  $w_m$  as the virtual point and just modify the last row of the  $A$  matrix so that the last two entries of the row are  $\theta$  and  $-\theta$ .

Most of the boundary conditions used in this work are based on one of the above types. The actual implementation varies depending on the discretization method and other factors but most of the concepts remain the same.

### 2.3 Transport Example Problem

As a summary of this section a basic transport example is solved. This serves to demonstrate some of the principles described above and justify the need for new techniques.

Consider transport in a very long tubular reactor of length with narrow aspect ratio (e.g. it can be approximated as one dimensional) and length  $L = 2.0$  m. The species of interest,  $i$ , has a diffusion coefficient of  $d = 1\text{E-}4$  m<sup>2</sup>/s in the surrounding fluid. The fluid is moving through the reactor with a mass-average velocity of 0.3 m/s. In the reactor,  $i$  is created by a first order reaction with rate constant  $k_1 = 0.2$  s<sup>-1</sup> in the presence of a homogeneous catalyst. The transport is thus described by the familiar equation

$$\frac{\partial c}{\partial t} + v \frac{\partial c}{\partial x} = d \frac{\partial^2 c}{\partial x^2} + k_1 c .$$

The initial condition defined such that there is a spike in the concentration of  $i$  at  $l = 1.0$  m.

$$c_i(x, 0) = \begin{cases} \max(0, 0.4 \cdot x - 3) & x \leq \ell \\ \max(0, -0.4 \cdot x + 5), & x > \ell \end{cases}$$

The boundary conditions are no-flux at the outlet and a zero concentration at the inlet,

$$c_i(0, t) = 0 \frac{\text{mol}}{\text{m}^3}$$

$$-d \frac{\partial c_i}{\partial x}(\infty, t) = 0 \frac{\text{mol}}{\text{m}^2 \text{s}}$$

The variables can be non-dimensionalized as follows:

$$\text{dimensionless position: } \bar{x} = \frac{x}{L}$$

$$\text{dimensionless time: } \bar{t} = \frac{td}{L^2}$$

$$\text{dimensionless concentration: } b(\bar{x}, \bar{t}) = \frac{c(x, t)}{c_{ref}}$$

$$\text{Damköhler number: } Da = \frac{k_1 L^2}{d}$$

$$\text{Peclet number: } Pe = \frac{vL}{d}$$

With these transformations, the equation is now

$$\frac{\partial b}{\partial \bar{t}} = \frac{\partial^2 b}{\partial \bar{x}^2} - Pe \cdot \frac{\partial b}{\partial \bar{x}} + Da \cdot b \quad (2.20)$$

with initial and boundary conditions

$$b(\bar{x}, 0) = \begin{cases} (1/c_{ref}) \max(0, 0.4 \cdot \bar{x}L - 3) & \bar{x} \leq \ell/L \\ (1/c_{ref}) \max(0, -0.4 \cdot \bar{x}L + 5), & \bar{x} > \ell/L \end{cases}$$

$$b(0, \bar{t}) = 0$$

$$\frac{\partial b}{\partial \bar{x}}(1, \bar{t}) = 0$$

To solve this numerically, the second derivative is approximated with the second order central scheme and the first derivative with the first order upwind scheme (see Section 2.2.1). This leads to a right hand side of equation (2.20) that is approximated with the matrix (note the approximated values for  $b$  are represented by  $w_j$ ).

$$A = \begin{pmatrix} -\frac{2}{h^2} + Da & \frac{1}{h^2} - \frac{Pe}{2h} & & & & & \\ \frac{1}{h^2} + \frac{Pe}{2h} & -\frac{2}{h^2} + Da & \frac{1}{h^2} - \frac{Pe}{h} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \frac{1}{h^2} + \frac{Pe}{2h} & -\frac{2}{h^2} + Da & \frac{1}{h^2} - \frac{Pe}{h} & & \\ & & & \frac{1}{h^2} + \frac{Pe}{2h} & -\frac{2}{h^2} + Da & & \\ & & & & & \frac{1}{h^2} + \frac{Pe}{2h} & -\frac{2}{h^2} + Da \end{pmatrix}$$

The boundary conditions can be handled via the vector

$$g = \left[ \left( \frac{1}{h^2} + \frac{Pe}{2h} \right) \cdot b(0, \bar{t}), 0, \dots, 0, \left( \frac{1}{h^2} - \frac{Pe}{h} \right) \cdot \left( 2h \cdot \frac{\partial b}{\partial x}(1, \bar{t}) + w_{M-1} \right) \right]^T$$

to give an equation of the form

$$w'(t) = Aw(t) + g(t), \quad w(t_0) = w_0.$$

The solution was integrated via a fourth order Runge-Kutta method over a time of 10 seconds (corresponding to a span of  $\bar{t} = [0, 0.00025]$ ) and transformed back into the original variables.

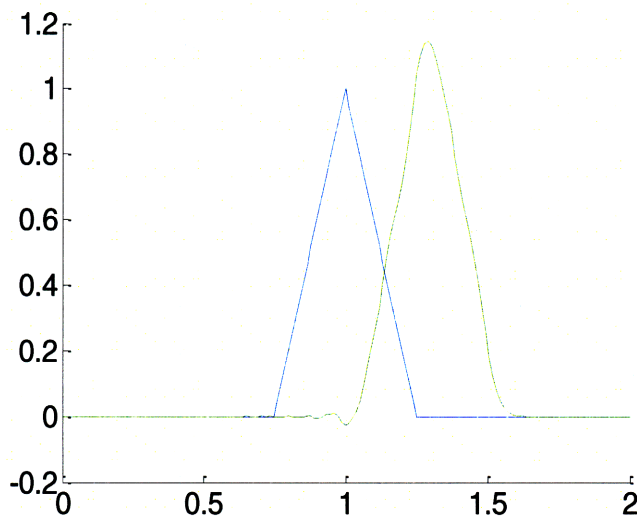


Figure 2.4: Transport Example, initial condition (dashed line) and apporximation

This figure plots concentration against position for both the initial condition and the final approximate solution. The results of this simple example largely agree with intuition. The peak moves through the reactor and spreads slightly due to diffusion. Also, there is a slight increase in concentration due to the first order reaction. However, note that there are also some oscillations in the middle that result in negative values for concentration.

This is clearly unphysical and demonstrates some of the shortcomings of these numerical approximations.

### 3.0 Integral Equations

Integral equations are a powerful but often overlooked tool. Though they have been around in some form for hundreds of years, their solution still represent a field of very active research, with new techniques being developed in many publications. Integral equations are very useful when describing phenomena of a hereditary nature or considering quantities that must be averaged over some space. In actuality, many boundary value problems based on differential equations can be formed in terms of integral equations where the integral itself naturally includes the boundary conditions without them needing to be imposed as separate equations. And when integral equations are mated with differential equations, forming partial integro-differential equations (PIDEs), they allow for the inclusion of history dependent factors, uncertainty in parameters, and other considerations that allow for more advanced physical and chemical models.

What defines an integral equation is an unknown function,  $g(x)$ , that appears under integral sign

$$g(x) = f(x) + \int K(x,u)g(u)du ,$$

where  $K(x,u)$  is called the kernel of the integral equation. This can also be written as

$$g(x) = f(x) + (Kg)(x) ,$$

where  $K$  is an integral operator that maps the input function  $g$  to an output over some given range.

Integral equations are often divided into two categories. Volterra integral equations take the form

$$h(x)g(x) = f(x) + \int_a^x K(x,u)g(u)du .$$

They are called Volterra integrals of the first kind or second kind if  $h(x)$  is zero or one, respectively. Fredholm integral equations have a constant limit of integration,

$$h(x)g(x) = f(x) + \int_a^b K(x,u)g(u)du ,$$

and are referred to as first or second kind if  $h(x)$  is zero or one, respectively. In both classes of integral equations, they are called homogeneous if  $f(x) = 0$ .

Integral transforms are a simple example of integral equations. For the Laplace transform,

$$\mathcal{L} f(t) = F(s)$$

the function  $f(t)$  is known, while for the inverse transform,  $\mathcal{L}^{-1} F(s)$ ,  $F(s)$  is known. So the inverse transform requires solving the integral equation

$$F(s) = \int_0^{\infty} e^{-st} f(t) dt$$

for  $f(t)$ . Fourier transforms can be considered in a similar way. Using these and other transforms allows the solution of some types of integral equations.

Overall, though, there are only a small number of problems that can be solved analytically. For problems that are currently of interest, numerical methods must almost always be used. In this thesis analytical solutions do serve as a point of comparison for numerical methods.

### **3.1 PIDE Solution Methods**

Even for this small sampling of examples, the solutions are handled using various numerical techniques. Unfortunately, these techniques for solving PIDEs do not fit into nearly as well a defined framework as PDE techniques discussed throughout this document. Each new situation can involve a new technique to obtain useful results within the constraints of the system. Indeed, novel methods are being published all the time. This is both the challenge and excitement of PIDEs: They represent a field of research that is very much at the frontier.

While outlining the solution methods employed even in this small sampling would take many more pages, there are a few common elements that are taken from basic integral equation methods. Consider the Volterra integral equation

$$g(x) = x - \int_0^x (x-u)g(u)du$$

The integral can be represented as a sum of  $N$  terms via some basic method such as a midpoint rule or Simpson's rule. Taking sample values for  $x$  and these points gives  $N$  equations for the unknown function  $g(x)$ . Using an interpolation (e.g. Lagrange interpolation) can then provide a functional approximation for  $g(x)$ . Note that in this simple example, Laplace transforms can show that the true solution is in fact  $g(x) = \sin(x)$ . These types of integral approximation and collocation methods represent one way to attack integral equations and can often be meshed with time-stepping PDE methods when the integral is taken with respect to the time variable. As mentioned above, there are many solution methods in use, but collocation of some form does often serve as a basis.

This basic solution technique works well in many cases, but one of the main challenges in solving integral equations is that the integral is often taken over a large range. This means that the values of the solution at a given grid point depend on the values of many

surrounding points. This definitely is the case in population balances where the particle size can form from the fragmentation of particles up to an arbitrarily large size, resulting in an integral over a very large range of masses. Consider the generic integral equation

$$w(x) = \int_a^b K(x, x') w(x') dx'.$$

If the integral is approximated via some type of quadrature rule,  $x$  will then be discretized over the range of the integral with the number of points,  $M$ , dependent on the level of detail required. The function is now

$$w(x_i) = \sum_{j=1}^M r_j K_{ij} w(x_j), \quad i = 1, 2, \dots, M$$

where the  $r_j$ 's are the coefficients due to the quadrature rule and the  $K_{ij}$ 's are the approximations of the kernel at various  $x$ -values. Now  $w(x)$  is a vector and the value at each point,  $w(x_i)$  may depend on any number of the surrounding values of  $w(x_{i-1})$ ,  $w(x_{i+1})$ , etc. depending on the form of the kernel. When put in matrix form,

$$w(\underline{x}) = A(\underline{x})\underline{x} \Rightarrow \begin{bmatrix} w(x_0) \\ \vdots \\ w(x_M) \end{bmatrix} = \begin{bmatrix} r_1 K_{11} & \dots & \\ & \dots & \\ & & r_M K_{MM} \end{bmatrix} \begin{bmatrix} w(x_0) \\ \vdots \\ w(x_M) \end{bmatrix},$$

it becomes clear that a matrix inversion will be necessary to solve for  $w(x)$ . The problem is that  $A$  may be very dense depending on how far the non-local effects are significant.

Now consider a transient version of this problem resulting in a PIDE. The resulting temporal integration method will require solving the above equations at every time step. Recall that the matrices resulting from spatial discretization are rather sparse but still presented challenges to invert; the dense matrices of integral equations may be impractical to invert in the same context. This makes explicit methods all the more attractive in this context.

There are two common issues with the majority of PIDE solution methods. The first is that they often do not preserve the positivity of the underlying physical process. The second is that the time integration is not very efficient. These two issues can be addressed by the methods developed in this thesis.

### 3.2 Examples

There are many examples of integral and partial integro-differential equations in the literature. Presented below are several interesting examples which emphasize interesting aspects of these classes of equations. In Chapter 10.0 some of them are explained in more detail and solved using the methods developed in this thesis. First off, a basic integral equation is explored.



As mentioned above, there is a natural link between integral and differential equations. Consider the steady-state convection-diffusion-reaction equation in dimensionless form,

$$\frac{\partial^2 \bar{b}}{\partial \bar{x}^2} - Pe \cdot \frac{\partial \bar{b}}{\partial \bar{x}} + Da \cdot \bar{b} = 0$$

with boundary conditions

$$\bar{b}(a) = 0, \quad \bar{b}(z) = 0.$$

Integrating the equation gives

$$\frac{\partial \bar{b}}{\partial \bar{x}} - Pe \cdot \bar{b}(\bar{x}) + Da \cdot \int_a^{\bar{x}} \bar{b}(u) du + C_1 = 0$$

and integrating again gives

$$\bar{b}(\bar{x}) - Pe \cdot \int_a^{\bar{x}} \bar{b}(u) du + Da \cdot \int_a^{\bar{x}} \int_a^{\bar{x}} \bar{b}(u) du d\xi + C_1 \bar{x} + C_2 = 0.$$

Using the identity

$$\int_a^x \int_a^\xi f(u) du d\xi = \left[ \int_a^\xi f(u) du \right]_a^x - \int_a^x \xi f(\xi) d\xi = x \int_a^x f(u) du - 0 - \int_a^x \xi f(\xi) d\xi = \int_a^x (x-u) f(u) du$$

allows the simplifications

$$\bar{b}(\bar{x}) - Pe \cdot \int_a^{\bar{x}} \bar{b}(u) du + Da \cdot \int_a^{\bar{x}} (\bar{x} - u) \bar{b}(u) du + C_1 \bar{x} + C_2 = 0$$

and

$$b(\bar{x}) = \int_a^{\bar{x}} -Da(\bar{x} - u) + Pe \ b(u) du - C_1 \bar{x} - C_2. \quad (3.1)$$

Applying the first boundary condition gives

$$\bar{b}(a) = 0 = 0 - C_1 a - C_2 \quad \Rightarrow \quad C_2 = -C_1 a,$$

while applying the second boundary condition gives

$$\begin{aligned}\bar{b}(z) = 0 &= \int_a^z -Da(z-u) + Pe \bar{b}(u) du - C_1 z + C_1 a \\ \Rightarrow C_1 &= \frac{1}{z-a} \int_a^z -Da(z-u) + Pe \bar{b}(u) du\end{aligned}$$

Putting these values for the constants of integration into equation (3.1) gives

$$\bar{b}(\bar{x}) = \int_a^{\bar{x}} -Da(\bar{x}-u) + Pe \bar{b}(u) du - \frac{\bar{x}-a}{z-a} \int_a^z -Da(z-u) + Pe \bar{b}(u) du.$$

Splitting the last integral into two parts allows further simplification,

$$\begin{aligned}\bar{b}(\bar{x}) &= \int_a^{\bar{x}} -Da(\bar{x}-u) + Pe \bar{b}(u) du - \frac{\bar{x}-a}{z-a} \int_a^{\bar{x}} -Da(z-u) + Pe \bar{b}(u) du \\ &\quad - \frac{\bar{x}-a}{z-a} \int_{\bar{x}}^z -Da(z-u) + Pe \bar{b}(u) du \\ &= \int_a^{\bar{x}} \left( \frac{\bar{x}-a}{z-a} - 1 \right) Da(\bar{x}-u) - Pe \bar{b}(u) du + \int_{\bar{x}}^z \left( \frac{\bar{x}-a}{z-a} \right) Da(z-u) - Pe \bar{b}(u) du\end{aligned}$$

and if the kernel is written as

$$K(\bar{x}, u) = \begin{cases} \left( \frac{\bar{x}-a}{z-a} - 1 \right) Da(\bar{x}-u) - Pe, & a \leq u \leq \bar{x} \\ \left( \frac{\bar{x}-a}{z-a} \right) Da(z-u) - Pe, & \bar{x} \leq u \leq z \end{cases},$$

the equation for  $\bar{b}$  can be written as

$$\bar{b}(\bar{x}) = \int_a^{\bar{x}} K(\bar{x}, u) \bar{b}(u) du,$$

which is the form of a homogeneous Fredholm integral equation.

The real potential utility of integral equations comes when they are combined with differential equations to form PIDEs. This allows for an extension of standard transport models to more complicated systems and overcome certain shortcomings in the standard models.

### 3.2.1 Population Balance PIDEs

One of the most interesting type of systems that can be handled with PIDEs are those of agglomerating particles. These so called population balance systems are characterized as follows.

There are many systems where particles interact with each other to form larger agglomerates and degrade to form smaller ones. Such interactions may or may not be the result of chemical reactions, but they can often be treated in a common manner. For example, consider the coalescence of water droplets or the formation of polymers. Population balances are a tool that allows for the analysis of these effects.

Often times the most natural variable to use to describe these phenomena is mass, since it is conserved, while particle volume, diameter, etc. are not. With this, the variable  $n_N(m)$  can describe the number density of particles over a distribution of masses. Generally, this term is considered to have units of #/mass. If a constant volume system is being considered then the units could be #/mass/volume. Integrating this term would then give the concentration of particles in a given mass range. There are several phenomena that are of interest in population balance systems. The explanation given here follows that of [Obrigkeit, 2001]. This explanation is elaborated upon in Chapter 9.0.

First considered is coagulation. This refers to two particles colliding to form a larger member of the same species. In a process with a large number of species, such interactions are continuously occurring. It is also generally assumed that ternary collisions are relatively rare so that only binary interactions need to be considered. For such binary interactions, the rate is similar to a first order reaction process, but with the reaction constant being a function of particle sizes. Consider two particles of sizes  $a$  and  $b$ . The rate of their coagulation is

$$\text{rate} = \beta(a,b)c_a c_b,$$

where the  $c$ 's represent concentrations and  $\beta(a,b)$  is known as the coagulation kernel. It will soon become obvious that this is an integration kernel as defined in the previous section.

If a particle of mass  $m$  is considered, the particles that create it can be of sizes  $a$  and  $m - a$  where  $a$  can range from 0 to  $m$ . The rate of increase of particles of mass  $m$  due to coagulation is then the product of the kernel and the concentrations of particles. To properly consider the concentration of particles, the range of masses,  $\Delta m$  must be multiplied by the number density,  $n_N(m)$ . The product  $n_N(m)\Delta m$  then represents the number of particles between size  $m$  and  $m + \Delta m$ . This gives the following rate equation:

$$\frac{dn_N(m)}{dt} \Delta m = \beta(m-a,a)n_N(m-a)\Delta(m-a)n_N(a)\Delta a.$$

Taking the delta values down to infinitesimal size results in  $da$  and  $d(m-a)$ . Dividing the whole equation by  $dm$  and recognizing that  $d(m-a)/dm = 1$  gives

$$\frac{dn_N(m)}{dt} = \beta(m-a,a)n_N(m-a)n_N(a)da.$$

To account for all of the particles up to size  $m$ , the right hand side must be integrated. However, note that this basic form would double count all of the interactions so the final expression should be

$$\frac{dn_N(m)}{dt} = \frac{1}{2} \int_0^m \beta(m-a, a) n_N(m-a) n_N(a) da .$$

Note that this is an integro-differential equation of Volterra type.

Particles of a given size also can also be decreased due to coagulation with other particles. In this case, there is a collision of a particle of size  $m$  with a particle of size  $a$  where  $a$  can now be arbitrarily large. This results in the equation

$$\frac{dn_N(m)}{dt} \Delta m = -\beta(m, a) n_N(m) \Delta(m) n_N(a) \Delta a$$

which for infinitesimal deltas can be simplified to

$$\frac{dn_N(m)}{dt} = -\beta(m, a) n_N(m) n_N(a) da .$$

This can be integrated of all possible a values to give

$$\frac{dn_N(m)}{dt} = n_N(m) \int_0^{\infty} \beta(m, a) n_N(a) da .$$

Note that infinite particle size can be interpreted as representing a distinct phase.

Fragmentation represents the reverse process of coagulation. A particle splits into an arbitrary number of smaller particles. The loss of particles of a given size is a first order process that depends upon the removal rate expression  $g(m)$ . Loss of particles in this manner is described by a simple first order differential equation,

$$\frac{dn_N(m)}{dt} = -g(m) n_N(m) .$$

Increase in particles of a given size due to fragmentation is more complicated. The rate of fragmentation, the number of particles created and the distribution of particle sizes all factor into the equation. The rate of fragmentation is  $g(a)$  as described above. The amount of particles generated is described by  $v(a)$ . The distribution of sizes depends on both the fragmented particle size and the size of interest. Another kernel, the fragmentation kernel,  $\gamma(a, m)$ , is used to describe this. Similarly to above, the product of all these factors and the respective number densities and mass ranges. So to describe the

increase in particles of size  $m$  due to the fragmentation of particles of an arbitrary size  $a$  the expression is

$$\frac{dn_N(m)}{dt} \Delta m = g(a)n_N(a)\Delta a \cdot v(a) \cdot \gamma(a, m)n_N(m-a)\Delta m .$$

Taking infinitesimal deltas and simplifying results in

$$\frac{dn_N(m)}{dt} = g(a)v(a)\gamma(a, m)n_N(a)da .$$

To consider all sizes  $a$  that could result in a particle of size  $m$ , the right hand side must be integrated to give

$$\frac{dn_N(m)}{dt} = \int_m^{\infty} g(a)v(a)\gamma(a, m)n_N(a)da .$$

Any of the above phenomena could occur in a given system possibly leading to several different integral terms in the same expression. And the above expressions consider only one species with only one attribute of interest (mass in this case). Adding species would require additional equations and adding attributes would require multiple integrals, both of which would add to the challenge of solving these already challenging systems.

### 3.2.2 Other Examples

The standard derivation of the transport equations effectively reduces any dependencies down to the immediately surrounding points in space and time. This works well when particles are very small and can interact only with their immediate neighbors, but this doesn't allow for more complicated interactions that may depend on the surrounding elements. Also, there is an inherent lack of time delay between cause and effect that can lead to some incorrect behavior when the standard equations are solved. A few examples are given here that exemplify some of these issues.

The standard transient reaction-diffusion equation,

$$\frac{\partial c}{\partial t} = d \frac{\partial^2 c}{\partial x^2} + f(c)$$

is known as the Fisher equation when the reaction term is given by

$$f(c) = k(c - c^2).$$

However, when there is an initial condition in the form of a step function, there are some unphysicalities that arise in the solution. In the solution, it is known that a traveling wave

will result with velocity equivalent to  $\sqrt{4dk}$ , but for very fast chemical reactions, this wave can propagate faster than the transport processes. To remedy this discrepancy with reality, a flux with memory term is introduced,

$$J = -\frac{d}{\tau} \int_0^t e^{-\frac{t-s}{\tau}} \frac{\partial c}{\partial x}(x,s) ds,$$

to give an integro-differential equation for transport,

$$\frac{\partial c}{\partial t} = \frac{d}{\tau} \int_0^t e^{-\frac{t-s}{\tau}} \frac{\partial^2 c}{\partial x^2}(x,s) ds + k(c - c^2).$$

This is known as the generalized Fisher-Kolmogorov-Petrovski-Piskunov equation. Its solution is discussed in, for example, [Araujo *et al.*, 2004].

Radiative heat transport can also involve PIDEs. A problem considered in [Frankel & Osborne, 2000], among others is a fairly common example. Consider a one-dimensional absorbing region, bounded by black walls. The cooling has the familiar fourth-power of temperature relationship. When the terms are all non-dimensionalized, the partial integro-differential equation for dimensionless temperature,  $\theta$ , is

$$\frac{\partial \theta}{\partial t}(x,t) = \frac{1}{2} E_2(x) + \theta_0^4 E_2(L-x) - \theta^4(x,t) + \frac{1}{2} \int_0^L E_1(|x' - x|) \theta^4(x',t) dx'$$

where  $\theta_0$  is the initial condition and  $E_n(\cdot)$  is the  $n^{\text{th}}$  exponential integral function,

$$E_n(x) = \int_1^\infty \frac{e^{-xt}}{t^n} dt.$$

Examples exist in biological systems as well. In [Mogilner & Gueron, 2000], a model is developed that considers pattern formation in bacterial swarms. The basic formulation considers the cell density,  $C(x,t)$  and a chemoattractant and chemorepellent ( $S_1(x,t)$  and  $S_2(x,t)$ , respectively). The transport equations describing this system are

$$\begin{aligned} \frac{\partial C}{\partial t} &= d_b \frac{\partial^2 C}{\partial x^2} - A_1 \frac{\partial}{\partial x} \left( \frac{\partial S_1}{\partial x} C \right) + A_2 \frac{\partial}{\partial x} \left( \frac{\partial S_2}{\partial x} C \right) \\ \frac{\partial S_i}{\partial t} &= d_i \frac{\partial^2 S_i}{\partial x^2} - a_i C - b_i S_i, \quad i = 1, 2 \end{aligned}$$

where  $A_i$  represents the magnitude of the chemotactic responses,  $a_i$  represents the secretion of chemical by the bacteria, and  $b_i$  is the decay of the chemicals. When the diffusion of the chemicals is much faster than that of the bacteria, a quasi-stationary

distribution of chemicals in space develops. This simplification allows a solution for the chemical concentrations,

$$S_i = \frac{L}{L_i} \frac{a_i}{2b_i} \int_{-\infty}^{+\infty} \exp\left(\frac{L}{L_i}|x-x'|\right) C(x') dx',$$

where

$$L = L_2 = \sqrt{\frac{d_2}{b_2}}, \quad L_1 = \sqrt{\frac{d_1}{b_1}},$$

and  $x$  and  $t$  are now non-dimensionalized. Finally, defining

$$\hat{C} = (2d_2b_2)/(A_2a_2), \quad A = \frac{A_1a_1b_2}{A_2a_2b_1}, \quad a = \frac{L_2}{L_1},$$

and defining the kernel

$$K(x, x') = \text{sgn}(x) A e^{-a|x-x'|} - e^{-|x-x'|},$$

the entire system can now be represented by the PIDE

$$\frac{\partial \hat{C}}{\partial t} = \frac{\partial^2 \hat{C}}{\partial x^2} - \frac{\partial}{\partial x} \hat{C} \int_{-\infty}^{+\infty} K(x, x') \hat{C}(x') dx'.$$

These are just a few examples of problems represented by PIDEs. Other examples will be explored more fully in Chapter 10.0.

## **4.0 New Applications: Finance**

### ***4.1 Comparison with Chemical Engineering***

Chemical engineers develop skills that can be readily applied across many scientific fields. It is well known that problems in physics, chemistry, biology, and other natural sciences often solved by people in the field of chemical engineering. Financial engineering is in its infancy compared to other scientific and engineering disciplines. As such, there is tremendous opportunity to apply the capabilities of chemical engineering to this area. This provides the opportunity to merge the main portion of this work with finance as the capstone portion of this PhDCEP thesis.

Though transfer of heat and mass through a medium and the rise and fall of option prices seem to be quite disparate phenomena, the mathematics that describe them are quite analogous. Even in following the development of the transport equation and the Black-Scholes equation, the similitude is not evinced until the dénouement of the derivations (see below). It is this relationship that forms a basis for much of this thesis.

A quantitative treatment of transport phenomena has existed almost as long as calculus itself. Beginning with Newton's Law of cooling in 1701 and continuing with works by Fourier in 1822 and countless others, there is an enormous body of work dedicated to this topic. Despite existing in a traded form since 1848, financial derivatives received rigorous quantitative treatment only in 1973 with the celebrated papers of Merton and of Black and Scholes. Their theories stimulated a great increase in the trading of options and led to the development of increasingly complicated financial instruments.

The great interest in financial derivatives today notwithstanding, the body of work of solution methods developed for transport equations is both larger and more advanced. With this being the case, the field of chemical engineering has a great deal to offer to the study of financial derivatives.

At its heart, trading involves the exchange of a financial risk for a specified payment either now or at some future date. These transactions can take place explicitly between two parties or on an exchange.

There are two broad categories of derivatives relevant to this thesis. The first category encompasses forward and futures contracts. In a forward contract, one party agrees to buy an asset from another party at some future date for a specified amount. An important feature is that it requires no money to exchange hands initially. The same basic principles apply to futures as well, but they are usually traded on exchanges and have certain standardized features. In addition, they require a margin to protect both parties from default.

Option contracts are also based on the purchase (or sale) of an underlying asset at some future date for a set exercise or strike price. The key difference is that there is a right rather than an obligation to buy (or sell) an asset. With this flexibility, there must of



course be a premium to enter into such a contract, which is paid up front. Some basic terminology is necessary to describe options. The terms call and put refer to options to buy or sell an underlying asset, respectively. An option that is “in the money” is favorable to exercise. For a call option, this means that the asset price is above the exercise price. The option allows the holder to buy the asset at the exercise price and immediately sell it at the market price for a profit. A put option requires the asset price to be less than the exercise price to be in the money. The two simplest option types are European and American. European options can only be exercised at the expiration date whereas American options can be exercised any time prior to expiration. Beyond these two types, there are a large number of options with increasingly complex structures.

Clearly, financial derivatives allow for speculation. For the same amount invested, derivatives allow a holder to make greater profits than he would by buying the underlying asset itself. Of course, they also expose the holder to a greater loss potential as well. More importantly for the purposes of this thesis, though, financial instruments enable the mitigation or transfer of risk. For example, consider a portfolio of an asset and a put option on that asset. Clearly as the value of one increases, the value of the other decreases. At a given instant, some specific portfolio ration ensures that small unpredictable movements in asset price do not result in unpredictable changes in the portfolio price. Determining this relationship and adjusting the portfolio ratio accordingly is called hedging. By selling this portfolio for more than it is worth and hedging away risk throughout the life of the option, a risk-free profit can be made.

Undoubtedly, being able to quantify the relationship between asset price and derivative value is of great importance. The study and development of these models are central aims of this thesis.

Like any other model, the development of the mathematics necessary to analyze financial derivatives necessitates some assumptions. For many of these assumptions, analogies can be drawn with transport phenomena.

**Table 4.1: Financial Assumptions**

<b>Financial Assumptions</b>	<b>Transport Analog</b>
Investors always prefer the trading strategy earning the greatest profit	Heat flows from hot to cold regions
No arbitrage opportunities: If such opportunities existed, they would be exploited until a new market equilibrium was established	A system with no net external influences returns to a single equilibrium state
Infinite liquidity of market	Species transport are considered continuous despite discrete nature of molecules
Continuous trading is possible	
No counterparty risk: The instrument's seller is not relevant to the price	Indistinguishability of particles
No transaction costs	No friction, e.g. inviscid flow
The following parameters are deterministic and constant: interest rates, volatilities, dividend yields	Constant physical properties
The relationship between risk factors are not stochastic	

The fundamental model for determining the value of a European option is the Black-Scholes differential equation.

$$\frac{\partial V(S,t)}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,t)}{\partial S^2} + rS \frac{\partial V(S,t)}{\partial S} - rV(S,t) = 0 \quad (4.1)$$

where  $V$  represents option value,  $t$  is time,  $S$  is underlying asset price,  $\sigma$  is the volatility of the asset, and  $r$  is the risk-free interest rate. It was developed in a 1973 paper by Fisher Black and Myron Scholes. Its derivation and solution are attached in Section 4.2.

It is worth pointing out the similarities between (4.1) and the transport equations discussed in the previous section. After a change of variables, (4.1) becomes

$$\frac{\partial v}{\partial \tau} = \frac{\partial^2 v}{\partial x^2} + (k-1) \frac{\partial v}{\partial x} - kv$$

which is in the same form as

$$\frac{\partial \bar{b}}{\partial \bar{t}} = \frac{\partial^2 \bar{b}}{\partial \bar{x}^2} - Pe \cdot \frac{\partial \bar{b}}{\partial \bar{x}} + Da \cdot \bar{b}.$$

It is this similarity that will allow the methods developed in this thesis to later be applied to financial equations.

As with any differential equation, the key to a unique solution is the boundary conditions. Note that since the value is known exactly at expiry, final conditions are used instead of initial conditions. For example, a European call option has the following conditions:

$$\begin{aligned} V(0, t) &= 0 \\ V(S, t) &\underset{S \rightarrow \infty}{\sim} S \\ V(S, T) &= \max(S - E, 0) \end{aligned}$$

where  $T$  is the expiration time and  $K$  is the exercise or strike price. Alternately, Neumann boundary conditions can be used to give

$$\begin{aligned} \left. \frac{\partial V(S, t)}{\partial S} \right|_{S=0} &= 0 \\ \left. \frac{\partial V(S, t)}{\partial S} \right|_{S \rightarrow \infty} &= 1 \end{aligned}$$

European puts have similar boundary conditions.

An analytical solution of this differential equation is possible, as shown below. The final result is

$$\begin{aligned} V(S, t) &= SN(d_1) - Ke^{-r(T-t)}N(d_2) \quad \text{with} \\ d_1 &= \frac{\ln(S/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \\ d_2 &= \frac{\ln(S/K) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \end{aligned} \tag{4.2}$$

As the financial derivatives become increasingly complex, their descriptive equations become more and more difficult to solve analytically. This leads to the development of numerical solutions. Numerical solutions can be advantageous even when analytical solutions exist since these solutions can involve infinite series or other functions that are difficult to manipulate. Also, numerical solutions are much more useful when considering boundary conditions that need to be considered at each time step (such as American options). Overall, the discrete nature of price quotes makes numerical solutions a natural choice.

MATLAB code was written for the numerical solution of the Black-Scholes equation for European options. An example of the output is given below for a European call option. The intrinsic value (which is also the final condition) is plotted as well. Note that the option has a value still has a small region where it is still greater than zero even if when the asset value is below the exercise price. This represents the fact that there is

uncertainty about the future price of the underlying. As the time gets closer to expiry, this region becomes smaller and smaller.

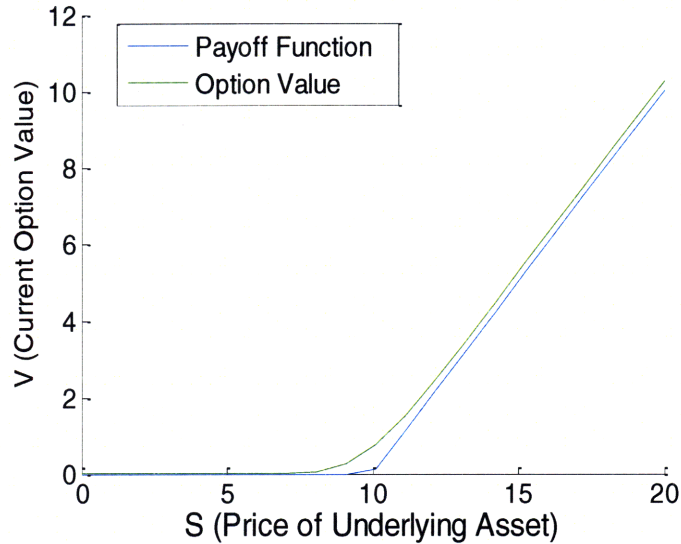


Figure 4.1: Value of a European call option

For this figure, the strike price is 10, the interest rate is 5%, the volatility is 20%, and there are 6 months to expiry.

## 4.2 Derivations of Fundamental Equations

When following this derivation it is worthwhile for the reader to refer back to Section 2.1 to consider the derivation and final form of the transport equation.

The Black-Scholes option pricing model determines the value of an option as a function of the underlying asset and time. By making a few assumptions (see Table 4.1) about stock price movements and market efficiency, a model is obtained that accurately represents the option price movements that are actually observed. The equations developed here are based on a single underlying asset, but the concepts can be readily extended to multiple assets (see equation (4.12)). The development here is based on European options. To price more exotic options generally involves a change of boundary conditions, but the solution becomes more difficult.

The return on an asset is the change in price divided by the original value,  $dS/S$ . This return can be modeled as a random walk with drift. The drift is the deterministic portion of the return. It is simply the average rate of growth of the asset and is denoted  $\mu$ . The stochastic portion of the return is based on the volatility of the returns,  $\sigma$ . The drift is multiplied by an infinitesimal time,  $dt$ , while the volatility is multiplied by a Wiener process,  $dX$ . This gives a stochastic differential equation for the asset value,

$$\frac{dS}{S} = \mu dt + \sigma dX \quad (4.3)$$

The Wiener process is defined such that for the time interval  $t-s$ ,  $X(t)-X(s)$  is a random variable normally distributed with zero mean and variance  $t-s$ . Each step is independent of the previous steps. In the infinitesimal case,  $dX$ , the variance becomes  $dt$ . Note that since  $X$  has a mean of zero,

$$E[dS] = E[\mu S dt + \sigma S dX] = \mu S dt .$$

The variance of a random variable is the accumulation of independent effects over an interval of time, which in this case is  $dt$ . The standard deviation is thus proportional to  $\sqrt{dt}$ . This allows  $dX$  to be written as

$$dX = \Phi \sqrt{dt} \quad (4.4)$$

where  $\Phi$  is a standard normal random variable.

The next result relies on Itô's Lemma. Consider some variable  $z$  described by the stochastic differential equation

$$dz = a(z,t)dX + b(z,t)dt .$$

Itô's Lemma states that, for some function  $f(z)$ ,

$$df = a \frac{df}{dz} dX + \left( b \frac{df}{dz} + \frac{1}{2} a^2 \frac{d^2 f}{dz^2} \right) dt .$$

For the case at hand, define  $f(S, t)$  as some smooth function of  $S$ . A Taylor series expansion of  $df$  gives

$$\begin{aligned} f(S, t) = & f(a, b) + \frac{\partial f(S, b)}{\partial t} (t - b) + \frac{\partial f(a, t)}{\partial S} (S - a) + \frac{\partial^2 f(a, b)}{\partial S \partial t} (S - a)(t - b) + \frac{1}{2} \frac{\partial^2 f(S, b)}{\partial t^2} (t - b)^2 \\ & + \frac{1}{2} \frac{\partial^2 f(a, t)}{\partial S^2} (S - a)^2 + \dots \end{aligned}$$

As the change in  $S$  and  $t$  (and therefore  $f$ ) becomes infinitesimal, we have

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial S} dS + \frac{\partial^2 f}{\partial S \partial t} dS dt + \frac{1}{2} \frac{\partial^2 f}{\partial t^2} dt^2 + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} dS^2 + \dots$$

Inserting (4.3) for  $dS$  and (4.4) for  $dX$  gives

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial S} \mu S dt + \sigma S \Phi \sqrt{dt} + \frac{\partial^2 f}{\partial S \partial t} \mu S dt + \sigma S \Phi \sqrt{dt} dt + \frac{1}{2} \frac{\partial^2 f}{\partial t^2} dt^2 + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \mu^2 S^2 dt^2 + \dots$$

and further expansion results in

$$df = \frac{\partial f}{\partial t} dt + S \frac{\partial f}{\partial S} \mu dt + \sigma \Phi dt^{1/2} + S \frac{\partial^2 f}{\partial S \partial t} \mu dt^2 + \sigma \Phi dt^{3/2} + \frac{1}{2} \frac{\partial^2 f}{\partial t^2} dt^2 + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \mu^2 dt^2 + 2\mu S \sigma \Phi dt^{3/2} + \sigma^2 S^2 \Phi^2 dt + \dots$$

Since powers of  $dt$  higher than unity will vanish in the limit of infinitesimal  $dt$ , this equation can be simplified to

$$df = \frac{\partial f}{\partial t} dt + S \frac{\partial f}{\partial S} \mu dt + \sigma \Phi dt^{1/2} + \frac{1}{2} S^2 \frac{\partial^2 f}{\partial S^2} \sigma^2 \Phi^2 dt .$$

Since  $E[\Phi]$  is 0, and  $E[\Phi^2]$  is  $(E[\Phi])^2 + \text{var}(\Phi) = 1$ , the deterministic portion of  $df$  is

$$E[df] = E \left[ \frac{\partial f}{\partial t} dt + S \frac{\partial f}{\partial S} \mu dt + \sigma \Phi dt^{1/2} + \frac{1}{2} S^2 \frac{\partial^2 f}{\partial S^2} \sigma^2 \Phi^2 dt + \dots \right] = \frac{\partial f}{\partial t} dt + \mu S \frac{\partial f}{\partial S} dt + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} dt$$

The stochastic component of  $df$  is given by

$$\frac{\partial f}{\partial S} \sigma S \Phi dt^{1/2} + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 \Phi^2 dt$$

The  $\Phi \sqrt{dt}$  term is simply  $dX$ , and the  $\Phi^2 dt$  term is  $dX^2$ . However,  $dX^2$  would have a variance proportional to  $dt^2$ , which means that it would disappear in the limit of infinitesimal  $dt$ . This means that, for infinitesimal  $dt$ ,  $dX^2$  is equal only to its deterministic portion and thus is not stochastic. This gives the final expression for  $df$ ,

$$df = \frac{\partial f}{\partial t} dt + \mu S \frac{\partial f}{\partial S} dt + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} dt + \sigma S \frac{\partial f}{\partial S} dX . \quad (4.5)$$

Consider now  $V(S, t)$  to be some option based on an underlying asset  $S$ . Also consider  $\Pi$  to be a portfolio consisting of the option and the underlying asset. Since  $V$  is a function of  $S$ , they are correlated, and their random components are proportional. Therefore, some linear combination of  $V$  and  $S$  will allow this randomness to be eliminated. Write  $\Pi$  as

$$\Pi = V - \Delta S .$$

For constant  $\Delta$  we have

$$\begin{aligned}
d\Pi &= dV - \Delta dS \\
&= \frac{\partial V}{\partial t} dt + \mu S \frac{\partial V}{\partial S} dt + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} dt + \sigma S \frac{\partial V}{\partial S} dX - \Delta \mu S dt - \Delta \sigma S dX \\
&= \frac{\partial V}{\partial t} dt + \mu S \left( \frac{\partial V}{\partial S} - \Delta \right) dt + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} dt + \sigma S \left( \frac{\partial V}{\partial S} - \Delta \right) dX
\end{aligned}$$

Since this equation resulted from a Taylor series expansion, all of the differentials are evaluated at a point. Therefore, they are constant for a given  $t$  and  $\Delta$  can be set equal to  $\partial V / \partial S$ . This will then eliminate the randomness from the equation and give

$$d\Pi = \left( \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt. \quad (4.6)$$

The return on an amount  $\Pi$  invested at the risk free rate,  $r$ , is given by the differential equation

$$\frac{d\Pi}{dt} = r\Pi,$$

so  $d\Pi$  would grow by an amount  $r\Pi dt$  in a time  $dt$ . If no arbitrage opportunities exist, then the right hand side of equation (4.6) must equal  $r\Pi dt$ . This is because the right hand side of (4.6) is deterministic so it should not be any greater (or less) than an investment at the risk free rate. If it were greater than  $r\Pi dt$ , an investor could make a risk-free profit by borrowing an amount  $\Pi$  and investing it in the portfolio. If the right hand side were less than  $r\Pi dt$ , an investor could short the portfolio and invest the  $\Pi$  in the bank. Therefore, equation (4.6) can be written as

$$r V - \Delta S dt = \left( \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt$$

with  $\Delta = \partial V / \partial S$ . Dividing by  $dt$  and rearranging gives

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0. \quad (4.7)$$

This is the Black-Scholes partial differential equation. Note that it is a linear backward parabolic partial differential equation.

The boundary conditions depend on the type of derivative (or combination thereof) that  $V$  represents. Note that there are final conditions rather than initial conditions since the equation is a backward parabolic equation. With  $E$  representing the exercise price,  $T$  representing expiry date, and  $r$  representing the risk free interest rate, the boundary and initial conditions for European options are as follows:

For a European call option:

$$C(0, t) = 0$$

$$C(S, t) \underset{S \rightarrow \infty}{\sim} S$$

$$C(S, T) = \max(S - K, 0)$$

For a European put option:

$$P(0, t) = Ke^{-r(T-t)}$$

$$\lim_{S \rightarrow \infty} P(S, t) = 0$$

$$P(S, T) = \max(E - S, 0)$$

Note that these boundary conditions are Dirichlet. Neumann boundary conditions can be used as well:

$$\left. \frac{\partial C(S, t)}{\partial S} \right|_{S=0} = 0$$

$$\left. \frac{\partial C(S, t)}{\partial S} \right|_{S \rightarrow \infty} = 1$$

for a European call option and

$$\left. \frac{\partial P(S, t)}{\partial S} \right|_{S=0} = -1$$

$$\left. \frac{\partial P(S, t)}{\partial S} \right|_{S \rightarrow \infty} = 0$$

for a European put option.

Further mathematical manipulation can make the differential equation simpler. Consider a European call option,  $C(S, t)$ , with boundary conditions as defined above. Define new variables by

$$S = Ke^x$$

$$t = T - \tau / \frac{1}{2} \sigma^2$$

$$C = Kv(x, \tau)$$

Where  $x$  is dimensionless asset price,  $\tau$  is dimensionless time, and  $v$  is dimensionless option value. Substituting these new variables in gives



$$\begin{aligned}\frac{\partial C(S,t)}{\partial t} &= \frac{\partial \tau}{\partial t} \frac{\partial C}{\partial \tau} \frac{\partial v}{\partial \tau} = -\frac{1}{2} \sigma^2 K \frac{\partial v}{\partial \tau} \\ \frac{\partial C(S,t)}{\partial S} &= \frac{\partial x}{\partial S} \frac{\partial C}{\partial v} \frac{\partial v}{\partial x} = \frac{1}{S} K \frac{\partial v}{\partial x} \\ \frac{\partial^2 C(S,t)}{\partial S^2} &= \frac{\partial}{\partial S} \left( \frac{\partial C(S,t)}{\partial S} \right) = \frac{\partial}{\partial S} \left( \frac{1}{S} K \frac{\partial v}{\partial x} \right) = K \left( \frac{1}{S} \frac{\partial^2 v}{\partial S \partial x} - \frac{1}{S^2} \frac{\partial v}{\partial x} \right) = K \left( \frac{1}{S} \frac{\partial^2 v}{\partial x^2 \partial S / \partial x} - \frac{1}{S^2} \frac{\partial v}{\partial x} \right) \\ &= \frac{K}{S^2} \left( \frac{\partial^2 v}{\partial x^2} - \frac{\partial v}{\partial x} \right)\end{aligned}$$

With these new values, equation (4.7) becomes, after rearrangement

$$\frac{\partial v}{\partial \tau} = \frac{\partial^2 v}{\partial x^2} + (k-1) \frac{\partial v}{\partial x} - kv$$

where  $k$  is defined as  $r / \frac{1}{2} \sigma^2$ . Note that this equation is very similar to the convection-diffusion-reaction equation (2.4).

Defining  $u(x, \tau) = ve^{-\alpha x - \beta \tau}$  with  $\alpha = -\frac{1}{2}(k-1)$  and  $\beta = -\frac{1}{4}(k+1)^2$  gives

$$\begin{aligned}\frac{\partial u}{\partial \tau} &= \frac{\partial^2 u}{\partial x^2} \quad -\infty < x < \infty, \tau > 0 \quad \text{with} \\ u(x, 0) &= \max e^{\frac{1}{2}(k+1)x} - e^{\frac{1}{2}(k-1)x}, 0, \quad \lim_{x \rightarrow \pm\infty} u(x, t) = 0\end{aligned}\tag{4.8}$$

Note that this equation is very similar to the heat equation (2.3).

This equation can be solved using a similarity solution. Such solutions are possible when the equation depends on  $x$  only through the combination  $\xi = x / \sqrt{\tau}$ . The fundamental solution is of the form

$$u_\delta(x, \tau) = \tau^{-1/2} U_\delta(\xi).\tag{4.9}$$

Differentiation of (4.9) shows that

$$\begin{aligned}\frac{\partial u_\delta}{\partial \tau} &= \frac{\partial}{\partial \tau} \tau^{-1/2} U_\delta(\xi) = -\frac{1}{2} \tau^{-3/2} U_\delta + \tau^{-1/2} U'_\delta \cdot \frac{-1}{2} x \tau^{-3/2} = -\frac{1}{2} \tau^{-3/2} U_\delta + \xi U'_\delta \\ \frac{\partial^2 u_\delta}{\partial x^2} &= \frac{\partial^2}{\partial x^2} \tau^{-1/2} U_\delta(\xi) = \tau^{-1/2} \frac{\partial}{\partial x} U'_\delta(\xi) \cdot \tau^{-1/2} = \tau^{-3/2} U''_\delta\end{aligned}$$

where primes denote differentiation with respect to  $\xi$ . The  $\tau$ 's cancel and equation (4.9) now satisfies the ordinary differential equation

$$U_{\delta}'' + \frac{1}{2}\xi U_{\delta}' = 0$$

which has the solution

$$U_{\delta}(\xi) = C_1 e^{-\xi^2/4} + C_2.$$

Setting  $C_2 = 0$  and normalizing such that  $\int_{-\infty}^{\infty} u dx = 1$  gives

$$u_{\delta}(x, \tau) = \frac{1}{2\sqrt{\pi\tau}} e^{-x^2/4\tau}.$$

Note that the fundamental solution has the initial value

$$u_{\delta}(x, 0) = \delta(x)$$

where  $\delta$  is the Dirac delta function.

A general initial value problem for equation (4.8) can be solved using the fundamental solution. The initial value for  $u$  can be written as

$$u(x, 0) = u_0(x) = \int_{-\infty}^{\infty} u_0(s) \delta(s-x) ds.$$

Due to symmetry, we also have

$$u_{\delta}(s-x, \tau) = \frac{1}{2\sqrt{\pi\tau}} e^{-(s-x)^2/4\tau}$$

with initial value

$$u_{\delta}(s-x, 0) = \delta(s-x).$$

So, for a fixed  $s$ , the function

$$u_0(s) u_{\delta}(s-x, \tau)$$

satisfies equation (4.8). Integrating over all values for  $s$  gives the general solution to equation (4.8),

$$u(x, \tau) = \frac{1}{2\sqrt{\pi\tau}} \int_{-\infty}^{\infty} u_0(s) e^{-(x-s)^2/4\tau} ds.$$

To apply this equation, it is convenient to define the new variable  $x' = (s-x)/\sqrt{2\tau} \Rightarrow dx' = ds/\sqrt{2\tau}$ . This gives

$$u(x, \tau) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} u_0(x'\sqrt{2\tau} + x) e^{-\frac{1}{2}x'^2/4\tau} dx'.$$

Since the initial condition is given by (4.8), this can be plugged in for  $u_0$ . Note that the lower bound on the initial value is zero, which corresponds to  $x = 0$ . Then we have

$$u(x, \tau) = \frac{1}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau}}^{\infty} e^{-\frac{1}{2}(k+1)(x'\sqrt{2\tau}+x)} e^{-\frac{1}{2}x'^2/4\tau} dx' - \frac{1}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau}}^{\infty} e^{-\frac{1}{2}(k-1)(x'\sqrt{2\tau}+x)} e^{-\frac{1}{2}x'^2/4\tau} dx'.$$

Each integral can be evaluated by completing the square in the exponent and defining  $\rho_1 = x' - \frac{1}{2}(k+1)\sqrt{2\tau}$  and  $\rho_2 = x' - \frac{1}{2}(k-1)\sqrt{2\tau}$  to give

$$u(x, \tau) = \frac{e^{\frac{1}{2}(k+1)x + \frac{1}{4}(k+1)^2\tau}}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau} - \frac{1}{2}(k+1)\sqrt{2\tau}}^{\infty} e^{-\frac{1}{2}\rho_1^2} d\rho_1 - \frac{e^{\frac{1}{2}(k-1)x + \frac{1}{4}(k-1)^2\tau}}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau} - \frac{1}{2}(k-1)\sqrt{2\tau}}^{\infty} e^{-\frac{1}{2}\rho_2^2} d\rho_2$$

Now define  $d_1 = \frac{x}{\sqrt{2\tau}} + \frac{1}{2}(k+1)\sqrt{2\tau}$  and  $d_2 = \frac{x}{\sqrt{2\tau}} + \frac{1}{2}(k-1)\sqrt{2\tau}$ . Due to the symmetry of the normal distribution, it is clear that the integrals are simply  $N(d_1)$  and  $N(d_2)$ , where  $N$  is the cumulative distribution function of the normal distribution.

Reverting back to the original dimensional variables and parameters gives

$$\begin{aligned} C(S, t) &= SN(d_1) - Ke^{-r(T-t)}N(d_2) \quad \text{with} \\ d_1 &= \frac{\ln(S/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \\ d_2 &= \frac{\ln(S/K) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \end{aligned} \tag{4.10}$$

This is the Black Scholes equation for a European call option. Due to put-call parity, a European put has the form

$$P(S, t) = Ke^{-r(T-t)}N(-d_2) - SN(-d_1). \tag{4.11}$$

Note that the equations thus far apply to a single underlying asset. Multiple assets would result a (integro-)differential equations with many more terms. The most important change is that the correlation between each asset must be considered. For example, the Black-Scholes PDE for  $n$  assets is

$$\frac{\partial V(S,t)}{\partial t} + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial^2 V(S,t)}{\partial S_i \partial S_j} + r \sum_{i=1}^n S_i \frac{\partial V(S,t)}{\partial S_i} - rV(S,t) = 0 \quad (4.12)$$

where  $S$  is now an  $n$ -length vector and  $\rho_{ij}$ , is the correlation coefficient between assets  $i$  and  $j$ . This is similar to an increase in dimensionality in a transport PDE but there can be many more than three dimensions in this case. The solution of these multi-asset equations is generally handled via some sort of simulation, such as Monte Carlo, as PDE/finite difference approaches have often proven impractical in such situations [Pacelli *et al.*, 1999]. If the approaches developed in this thesis for chemical engineering problems have difficulties when applied to more advanced finance problems, these simulation techniques can be considered more thoroughly. This may even allow methods developed for financial systems to be applied to chemical engineering systems.

### 4.3 Jump Processes

The story does not end with Black-Scholes. There are several underlying problems that make the model incomplete and can lead to difficulties when actually applied. Many of these can be addressed by jump processes.

Jump processes allow for a more accurate model of the movements in asset price. At the most basic level, asset prices are quoted discretely. Naturally, where considering a long enough time interval, such a process has the appearance of a continuous phenomenon, and this is the basis for many of the financial models that are used in derivative pricing. However, there are several problems that cannot be well incorporated into continuous diffusion models. For example, large sudden price movements can occur and losses are generally concentrated in a few large downward moves. Another issue is that perfect hedges do not exist in reality and some hedging strategies are better than others, but in the model, all hedging strategies lead to zero risk. These flaws in the models can lead to large problems when they are put into use.

These shortcomings led to the development of new types of models, such as jump processes. Jump processes can be divided into two main categories. In infinite activity models, every movement in time is essentially a jump. These give the most accurate representations of how prices actually move, but can be difficult to use in practice. Jump-diffusion models use an underlying Brownian model with jumps at random intervals. Such models can be incorporated well with the standard Black-Scholes model.

The Lévy process is the basis for these jump models. In essence, they are the discrete time analog of random walks. Also, they can be written as a superposition of a Wiener process and an infinite number of independent Poisson processes; indeed, Wiener and

Poisson processes are themselves Lévy processes. Lévy processes,  $(X_t)_{t \geq 0}$ , have the following properties:

- Independent increments: for every increasing sequence of times  $t_0, \dots, t_n$ , the random variables  $X_{t_0}, X_{t_1} - X_{t_0}, \dots, X_{t_n} - X_{t_{n-1}}$  are independent.
- Stationary increments: the probability law of  $X_{t+h} - X_t$  does not depend on  $t$ .
- Stochastic continuity:  $\forall \varepsilon > 0, \lim_{h \rightarrow 0} \mathbb{P} |X_{t+h} - X_t| \geq \varepsilon = 0$ .

A simple example of a Lévy process is a compound Poisson process.

$$X_t = \sum_{i=1}^{N_t} Y_i$$

where the  $Y_i$  are independent and identically distributed (i.i.d.) random variables and  $N_t$  is a Poisson process with intensity  $\lambda$ . The probability law for  $Y_i$  determines the jump size and the jump times are determined by the Poisson process,  $N_t$ .

The Lévy measure,  $\nu(A)$ , is the expected number of jumps per unit time whose step size belongs to  $A$ .

$$\nu(A) = \mathbb{E} \left[ \# \text{ } t \in [0,1]: \Delta X_t \neq 0, \Delta X_t \in A \right]$$

where  $(X_t)_{t \geq 0}$  is a Lévy process,  $\Delta X_t$  means jumps size, and # means “number of elements.” Using  $\nu$  provides the most general way to characterize Lévy processes.

The Lévy-Itô decomposition allows a Lévy process to be represented in terms of a covariance matrix,  $A$ , and drift vector,  $\gamma$ . This gives rise to a characteristic triplet  $(A, \nu, \gamma)$  as a way to characterize a Lévy process. The Lévy-Khinchin representation is a compact way to represent a Lévy-process in terms of its characteristic function and characteristic triplet. For one dimension, this is

$$\mathbb{E} \left[ e^{izX_t} \right] = e^{t\psi(z)} \quad \text{with}$$

$$\psi(z) = -\frac{1}{2} Az^2 + i\gamma z + \int_{-\infty}^{\infty} e^{izx} - 1 - izx \mathbf{1}_{|x| \leq \varepsilon} \nu(dx)$$

where  $\varepsilon$  is some arbitrary step size limitation.

An abbreviated derivation gives a partial integro-differential equation describing option prices that is similar to the Black-Scholes equation. The main difference is that the Lévy process replaces the random walk. Using the risk-neutral assumption, the asset price can be represented by

$$S_t = S_0 e^{r+X_t}$$

where  $r$  is the risk-free interest rate and  $X_t$  is a Lévy process with characteristic triplet  $(\sigma^2, \nu, \gamma)$ .

The value of a European option can be defined as a discounted conditional expectation of its terminal payoff,  $H(S_T)$ . Representing the value of the option by  $V(S, t)$ ,

$$V(S, t) = \mathbb{E} \left[ e^{-r(T-t)} H(S_T) \mid S_t = S \right]. \quad (4.13)$$

Using the following change of variables

$$\begin{aligned} x &= \ln(S/K) + r\tau \\ \tau &= T - t \end{aligned}$$

where  $K$  is the strike price, and defining

$$\begin{aligned} h(x) &= H(Ke^x) / K \\ u(x, \tau) &= e^{r(T-t)} V(S, t) / K \end{aligned}$$

allows equation (4.13) to be written as

$$u(x, \tau) = \mathbb{E} h(x + X_\tau)$$

Using the concept of the infinitesimal generator,  $L$ , (and assuming that  $h$  is in the domain of  $L^X$ ), we have

$$\frac{\partial u}{\partial \tau} = L^X u, \quad u(x, 0) = h(x)$$

and  $L^X$  is defined as

$$L^X f(x) = \gamma \frac{\partial f}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 f}{\partial x^2} + \int \left( f(x+y) - f(x) - y \mathbf{1}_{|y|<1} \frac{\partial f(x)}{\partial x} \right) \nu(dy).$$

Changing the variables back gives

$$\begin{aligned} \frac{\partial V(S, t)}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V(S, t)}{\partial S^2} + rS \frac{\partial V(S, t)}{\partial S} - rV(S, t) \\ + \int \left( V(Se^y, t) - V(S, t) - S(e^y - 1) \frac{\partial V(S, t)}{\partial S} \right) \nu(dy) = 0 \end{aligned}$$

Note that this is very similar to the Black-Scholes PDE except for the integral term, which of course represents the jump process.

#### 4.4 Option Example Problem

To make the above concepts more concrete, the following sample problem is formulated and solved. It outlines the basic method of attack for numerical solution of PDEs and further elucidate the underlying similarities between transport and finance equations. For comparison, recall the transport example in Section 2.3.

Consider a European call option on a stock with a strike price,  $K$ , of 10 \$. The volatility of the underlying asset is  $\sigma = 0.25 \text{ yr}^{-1/2}$  and the risk-free interest rate is  $r = 0.08 \text{ yr}^{-1}$ . The time to expiry is  $T = 0.5$  year. For more details on the variables and parameters, see Section 4.2. The final condition is the standard

$$V(S, T) = \max(S - K, 0)$$

meaning that the option is worth the amount the strike price exceeds the underlying stock price and is worthless if the stock price is lower than the strike price. The following boundary conditions are used

$$\begin{aligned} V(0, t) &= 0 \\ \left. \frac{\partial V(S, t)}{\partial S} \right|_{S \rightarrow \infty} &= 1 \end{aligned}$$

meaning that the option becomes worthless if the stock price drops to zero and the option value will change exactly as the stock price changes as the price becomes very high. For convenience of solution, the non-dimensionalization discussed in Section 4.2 is employed:

$$\text{dimensionless asset price: } \bar{x} = \ln\left(\frac{S}{K}\right)$$

$$\text{dimensionless time: } \bar{t} = \frac{1}{2} \sigma^2 \cdot (T - t)$$

$$\text{dimensionless option price: } v(\bar{x}, \bar{t}) = \frac{V}{K}$$

$$\text{parameter: } k = \frac{r}{\frac{1}{2} \sigma^2}$$

The equation is now

$$\frac{\partial v}{\partial \tau} = \frac{\partial^2 v}{\partial x^2} + (k - 1) \frac{\partial v}{\partial x} - kv \tag{4.14}$$

with the initial and boundary conditions

$$\begin{aligned} v(0, \bar{t}) &= \max(e^{\bar{x}} - 1, 0) \\ v(0, \bar{t}) &= 0 \\ \frac{\partial v(\bar{x}, \bar{t})}{\partial \bar{x}} \Big|_{\bar{x} \rightarrow \infty} &\sim \infty \end{aligned}$$

Note that for practical purposes, the infinities in the last boundary condition are approximated by sufficiently high asset prices (usually around 10 times the strike price). Observing the similarity to equation (2.20), the discretization of (4.14) is as follows with the discrete approximations for  $v$  being replaced by  $w_i$ :

$$A = \begin{pmatrix} -\frac{2}{h^2} - k & \frac{1}{h^2} + \frac{(k-1)}{2h} & & & & & \\ \frac{1}{h^2} - \frac{(k-1)}{2h} & -\frac{2}{h^2} - k & \frac{1}{h^2} + \frac{(k-1)}{2h} & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \frac{1}{h^2} - \frac{(k-1)}{2h} & -\frac{2}{h^2} - k & \frac{1}{h^2} + \frac{(k-1)}{2h} & \\ & & & & \frac{1}{h^2} - \frac{(k-1)}{2h} & -\frac{2}{h^2} - k & \end{pmatrix}$$

and the vector  $g$  is defined to handle the boundary conditions

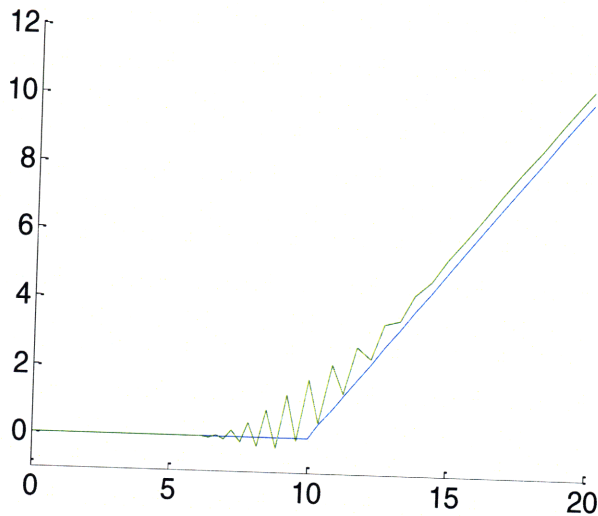
$$g = \left[ \left( \frac{1}{h^2} - \frac{(k-1)}{2h} \right) \cdot v(0, \bar{t}), 0, \dots, 0, \left( \frac{1}{h^2} + \frac{(k-1)}{2h} \right) \cdot \left( 2h \cdot \frac{\partial v}{\partial \bar{x}}(1, \bar{t}) + w_{M-1} \right) \right]^T$$

giving the equation

$$w'(t) = Aw(t) + g(t), \quad w(t_0) = w_0.$$

This equation was also integrated using a fourth order Runge-Kutta method and then transformed back into the original variables to give the following graph:





**Figure 4.2: Payoff function (dashed line) and approximate option value vs. asset price**

There is fairly close agreement with the analytical solution (see Figure 4.1). The present value of the option increases with option price and the amount that it exceeds the final price reflects the time value of money and uncertainty in the underlying asset price in the future.

As in the transport case, there are some problems with the solution. For underlying asset prices near the strike price, the solution yields option prices that are less than zero which is clearly impossible. Also, there are a few places where the solution predicts the present value of the option would be less than the future value, which can be refuted by an arbitrage argument.

While the need for accuracy is of general importance, these two examples immediately obviate the need for positivity preservation. The nonsensical negative values for concentration and price are easy to catch when comparing to known solutions, but if these methods were employed in a larger process, the negative values being input could cause problems from severe divergence in the overall solution to incorrect decisions downstream.

Similar to the transport example in Section 2.3, it is relatively easy to eliminate these errors by using a different spatial discretization and/or implicit time integration method. But as the systems become more complicated (e.g. multiple non-linear reactions, exotic option types with multiple underlying assets, etc.) the challenge of avoiding such inaccuracies becomes much greater. The underlying causes of (and some remedies for) these errors are discussed in the next section. Solution methods that can both eliminate these errors and remain efficient are one of the central goals of this thesis.

## 5.0 Novel Solution Techniques

There are many methods used for solving the types of challenges outlined in the previous sections. Presented here are several new tools that have been incorporated into the methods developed in this thesis.

### 5.1 Runge-Kutta Chebyshev

Implicit numerical methods depend on the evaluation of the function at the time step currently being calculated, so some type of numerical solution method must be employed. As these solvers can be costly, recently developed techniques allow explicit techniques to be used on systems that would normally rely upon BDF-type methods.

Developments of explicit methods that have larger stability domains [see, e.g. Zhang, 2004; Ascher *et al.*, 1997] allow for somewhat stiff systems to be handled. Based on Runge-Kutta methods, they still require a greater number of time steps than a corresponding implicit method, but the simpler computation at each time step more than makes up for this.

The basic concept of Runge-Kutta Chebyshev methods is to construct an explicit method with stability domain extended as far as possible along the negative real axis in the framework of Runge-Kutta. This can allow certain systems to be solved that would generally require an implicit method to be solved more efficiently.

Any explicit Runge-Kutta method has a stability region that encompasses some section of the negative real line. The largest magnitude value that is included is referred to as the real stability boundary,  $\beta_R$ . This value is determined by the stability function of the method, which takes the form

$$R(z) = \gamma_0 + \gamma_1 z + \gamma_2 z^2 + \dots, \quad z = \tau\lambda \quad (5.1)$$

where  $\tau$  is the time step size and  $\lambda$  is an eigenvalue of the matrix from the general ODE system  $w' = Aw + g(t)$ . Note that to remain first order consistent, the first two coefficients must be one. To see this, recall that the solution to the scalar test problem,

$$\frac{\partial}{\partial t} w(t) = \lambda w(t) \quad \Rightarrow \quad w(t) = e^{t\lambda}$$

and the approximation

$$e^z = 1 + z + \frac{1}{2}z^2 + \dots$$

The form of the stability polynomial is determined by the stages,  $s$ , of the Runge-Kutta method. As it turns out the polynomial,  $P_s(z)$ , that maximizes  $\beta_R$  is a shifted Chebyshev polynomial of the first kind,

$$P_s(z) = T_s\left(1 + \frac{z}{s^2}\right)$$

where  $T_s$  is the Chebyshev polynomial of degree  $s$ , defined as

$$\begin{aligned} T_s(x) &= \cos(s \arccos(x)), \quad x \in [-1, 1] \quad \text{or} \\ T_0(x) &= 1, \quad T_1(x) = x, \quad T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \quad 2 \leq j \leq s, \quad x \in \mathbb{C}. \end{aligned}$$

Recall that  $R(z)$  must be less than or equal to one for stability. With this constraint, it turns out that the largest value for  $\beta_R$  that can be achieved is  $2s^2$  as the following graph illustrates.

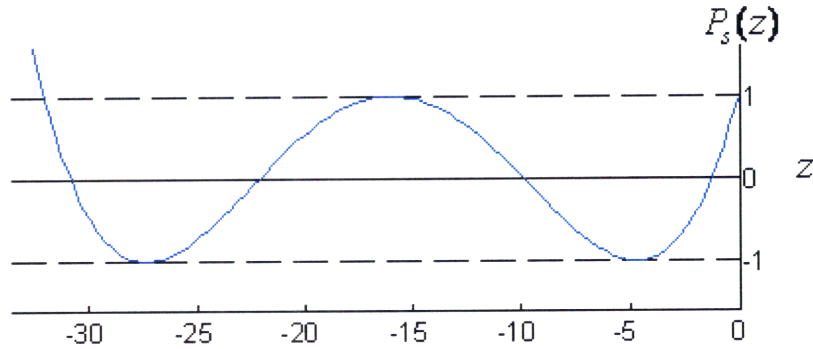


Figure 5.1: Shifted Chebyshev polynomial,  $s=4$

To prove this is true, consider a hypothetical polynomial of order  $s$  the form of (5.1) that meets the same requirements and has  $\beta_R \geq 2s^2$ . Such a polynomial would intersect  $P_s(z)$   $s-1$  times since  $P_s(z)$  has  $s-1$  points of tangency with the lines  $\pm 1$  for  $z < 0$ . There is then a difference polynomial of the form  $z^2(\tilde{\gamma}_2 z^2 + \dots + \tilde{\gamma}_s z^{s-2})$  since the first two coefficients are the same. But such a polynomial can have at most  $s-2$  roots which contradicts the previous statements, so there is no difference polynomial since the best achievable is  $P_s(z)$ .

Using shifted Chebyshev polynomials to define the stability function results in the following stability domain.

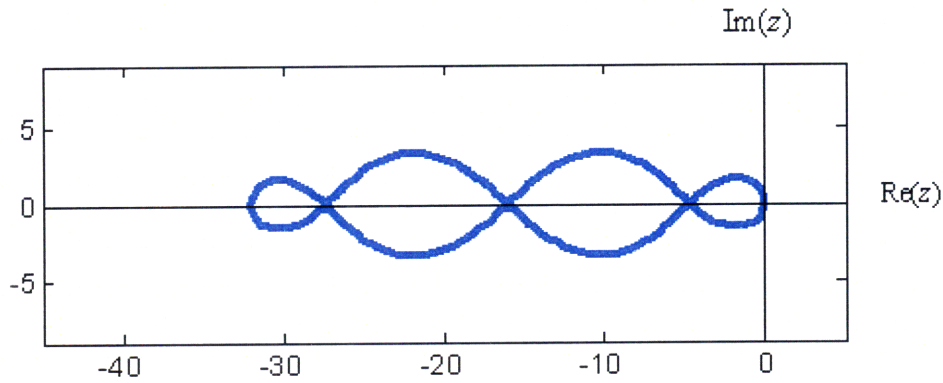


Figure 5.2: Stability domain for shifted Chebyshev polynomial,  $s=4$

Note that the upper and lower bounds in the imaginary direction of the stability domain exactly coincide with the real line at  $s-1$  locations. This can be problematic as slight perturbations in  $z$  could lead to instabilities. For this reason, damping is added by slightly modifying the polynomials

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

where  $\omega_0 > 1$  is a parameter often chosen to be  $\omega_0 = 1 + \varepsilon / s^2$  where the damping coefficient  $\varepsilon$  is some small positive number.  $\omega_1$  is necessary to obtain first order consistency. This new polynomial results in the following stability domains.

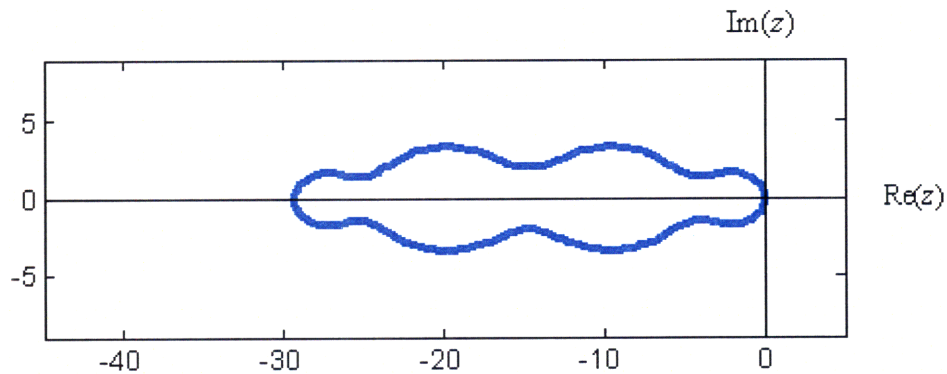


Figure 5.3: Stability domain for damped, shifted Chebyshev polynomial,  $s=4$ ,  $\varepsilon=2/13$

Note that now the stability domain is now much less restricted along the real axis. The damping coefficient can also act as a parameter to capture eigenvalues that are some distance from the real line. Of course, there is some price to pay for this: The real stability boundary decreases linearly with  $\varepsilon$ . The reduction can be approximated as

$$\beta_R = \frac{2\omega_0 T_s'(\omega_0)}{T_s(\omega_0)} \approx \left(2 - \frac{4}{3}\varepsilon\right)s^2.$$

So far the discussion has been limited to first order, but second order consistency and higher is readily achievable. As above, though, the price paid for higher order consistency is a reduction in the real stability domain. There are no known analytical expressions for the optimal coefficients for order greater than one, but numerical approximations exist.

The same notions about damping also apply to higher order methods. One form for the damped polynomial is

$$P_s(z) = 1 + \frac{T_s''(\omega_0 + \omega_1 z)}{T_s''(\omega_0)} T_s(\omega_0 + \omega_1 z) - T_s(\omega_0), \quad \omega_1 = \frac{T_s'(\omega_0)}{T_s''(\omega_0)}$$

For higher order versions and other variations, see [Hundsdorfer & Verwer 2003].

The implementation of these methods in the Runge-Kutta framework is the next major hurdle. As it turns out, stability must be considered on each intermediate stage as well as in the standard overall context. A full explanation of this can be found in [Hundsdorfer & Verwer 2003], but the important result is that the Runge-Kutta coefficients must be chosen with respect to this new set of constraints. The method of van der Houwen & Sommeijer is one of the best implementations to date and has been employed in several different contexts.

There are a few other methods that fall into the category of explicit extended stability domain methods. The most promising of these is the Orthogonal Runge-Kutta Chebyshev method of Abdulle. It allows for order consistency of four and includes a somewhat larger portion of the imaginary axis in its stability region than does RKC. Currently, it is not as well-developed as RKC, but may become an important option as its development continues.

## 5.2 Positivity Preservation

Maintaining positivity of the solution is an important property of any numerical method developed in this thesis. It presents challenges in both the spatial and temporal discretizations. The prevention of unphysical diffusion is another desired trait that is often at odds with positivity. These two phenomena are first briefly discussed in terms of spatial discretization on the simple constant coefficient convection equation

$$\frac{\partial b}{\partial t} + v \frac{\partial b}{\partial x} = 0, \quad b(x, t) = \begin{cases} \sin(\pi x)^{50}, & x \in [0, 1] \\ 0, & \text{otherwise} \end{cases}$$

The initial graph of this solution is as follows.

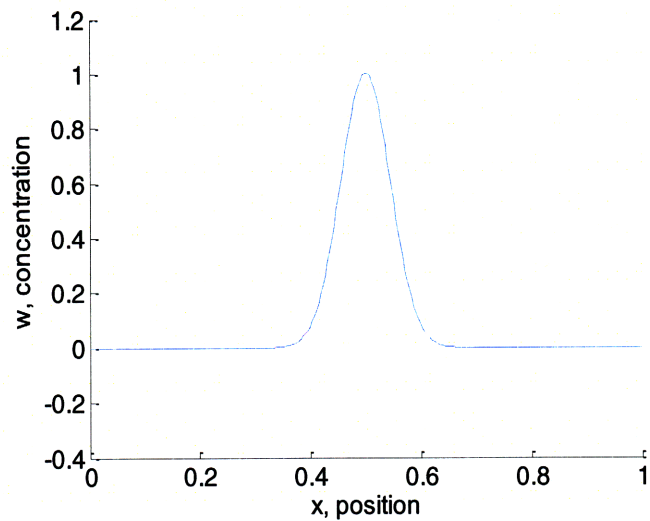


Figure 5.4: Convection Equation, Initial Condition

The solution of this equation is simply

$$b(x,t) = \sin(\pi(x-vt))^{50},$$

a shift of the solution along the  $x$ -axis without any change in shape (If  $v$  is an integer value, the graph will shift one period and be indistinguishable from the initial conditions).

Consider first the first-order upwind approximation (see Section 2.2.1)

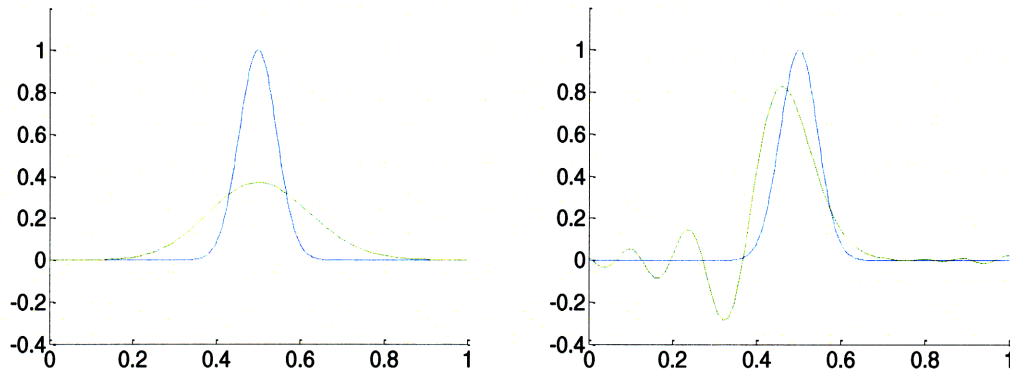
$$w'_j(t) = \frac{v}{h} w_{j-1}(t) - w_j(t), \quad j = 1, 2, \dots, m$$

and assume periodic boundary conditions on the interval  $[0,1]$ .

Also the second-order central approximation

$$w'_j(t) = \frac{v}{2h} w_{j-1}(t) - w_{j+1}(t), \quad j = 1, 2, \dots, m$$

under the same conditions. These systems were solved using an implicit trapezoid rule with a very small time step so that all of the errors are due to the spatial discretizations.



**Figure 5.5: Concentration vs. Position, Smooth Profile**

The approximation (solid line) with  $h=1/50$  using a 1<sup>st</sup> order upwind scheme (left) and a 2<sup>nd</sup> order central scheme (right).

The upwind method has an absolute error of 0.0660, a relative error of 0.0045 and a peak error of 0.4679, while those same respective errors for the central scheme are 0.0618, 0.0049, and 0.2990. The definition of these errors was put forth in Section 2.2.5.

It is immediately clear that these two methods introduce significant errors, and this brings up several important points. Firstly, note that the second order scheme does not seem to be especially more accurate than the first order scheme. This demonstrates that although higher order usually results in a better solution, this is not guaranteed to be the case.

The first scheme exhibits an inaccurate spread in the solution that seems to act as if there is some type of diffusion. The reason for this can be seen by further expanding the underlying difference formula (2.8) to give

$$\frac{\partial b}{\partial x}(x) = \frac{1}{2}h \frac{\partial^2 b}{\partial x^2}(x) + \frac{1}{h} b(x) - b(x-h) + O(h^2).$$

So by using the first order upwind approximation a diffusion term is introduced into the solution method leading to the spreading of the solution. This phenomenon is known as “numerical diffusion.”

The second scheme shows fairly large oscillations leading to negative values for concentration. However, the overall character of the solution is reasonably well preserved. A similar expansion of the second-order central approximation (2.9) yields a spatial third derivative. This dispersion term manifests itself as waves that travel at different speeds than the actual waves in the system, thereby causing oscillations.

While there are spatial discretization schemes that utilize more points and can provide more accurate results, they only mitigate the two problems discussed above: no basic method can remain completely positive and retain a very high level of accuracy. To

maintain positivity more advanced techniques are often needed, each with their own drawbacks.

One example here is implemented here for expository purposes. It was developed by [Hundsdoerfer *et al.*, 1995]. It is designed to solve an advection equation of the form

$$\frac{\partial b}{\partial t} + \frac{\partial vb}{\partial x} = 0.$$

Its implementation in MATLAB was undertaken to demonstrate the properties of such methods in general. While the above equation is fairly simple, it was shown to be applicable in two and three dimensions as well. More importantly, this method serves as a starting point for the more advanced positivity preserving methods developed within this thesis.

Formally, positivity requires that for any positive initial condition  $\{w_j(t_0)\}$ ,  $w_j(t_0) \geq 0$ ,  $\forall j$  the solution over time  $\{w_j(t)\}$ , must remain positive for all  $t \geq t_0$ . In practice, a lack of positivity manifests itself as over- and under-shoots of the real solution due to spurious oscillations. Note that here, as in the rest of this thesis,  $w$  values represent numerical approximations while  $b$  values are the exact quantity.

When using the method of lines (MOL) approach, the spatial and temporal steps are considered independently. They are still, however, closely related, especially in the context of positivity preservation. As mentioned above higher order methods are generally more accurate, but only the first order upwind approximation is inherently positivity preserving. This fundamental problem leads to the concept behind this (and many other) positivity preserving schemes: Use a high order discretization as large a region as possible and employ decreasingly accurate (but increasingly positivity preserving) discretizations as the variations in the solution become increasingly strong.

For simplicity of implementation, it was assumed that the velocity,  $v$ , was constant. However, it is fairly straightforward to evaluate the velocity along with the concentration at each point.

The stencil width of this scheme is five, meaning that values from the  $(j-2)^{\text{th}}$  through  $(j+2)^{\text{th}}$  spatial points are used to estimate the value at  $j$ . Wider stencils can add more accuracy, but cause more difficulty when applying boundary conditions.

The final semi-discrete form is

$$w'(t) = \frac{v}{h} \left[ 1 + \frac{1}{2} \phi_{j+1/2} - \frac{\frac{1}{2} \phi_{j-1/2}}{r_{j-1/2}} \right] w_{j-1} - w_j \quad (5.2)$$

where  $r_{j-1/2}$  is the slope ratio,



$$r_{j-1/2} = \frac{w_j - w_{j-1}}{w_{j-1} - w_{j-2}}$$

and  $\phi_{j-1/2}$  is the positivity preserving filter. The slope ratio is the method for determining regions of strong variation in the positivity filter. To maintain positivity, the limiter must have the following properties

$$\begin{aligned} \phi_{j-1/2} &= 0, \quad \text{if } r_{j-1/2} \leq 0 \\ 0 &\leq \phi_{j-1/2}, \phi_{j-1/2} \leq \delta, \quad \text{for some } \delta > 0. \\ \phi_{j-1/2} &\leq 2r_{j-1/2} \end{aligned} \tag{5.3}$$

In its final implementation it takes the form

$$\phi(r) = \max(0, \min(2r, \min(\delta, K(r)))) \tag{5.4}$$

where

$$K(r) = \frac{1-k}{2} + \frac{1+k}{2} r.$$

The  $k$ 's take values of values of 1, -1, and  $\frac{1}{3}$ , to obtain second-order central, second-order upwind, and third-order upwind biased discretizations, respectively.

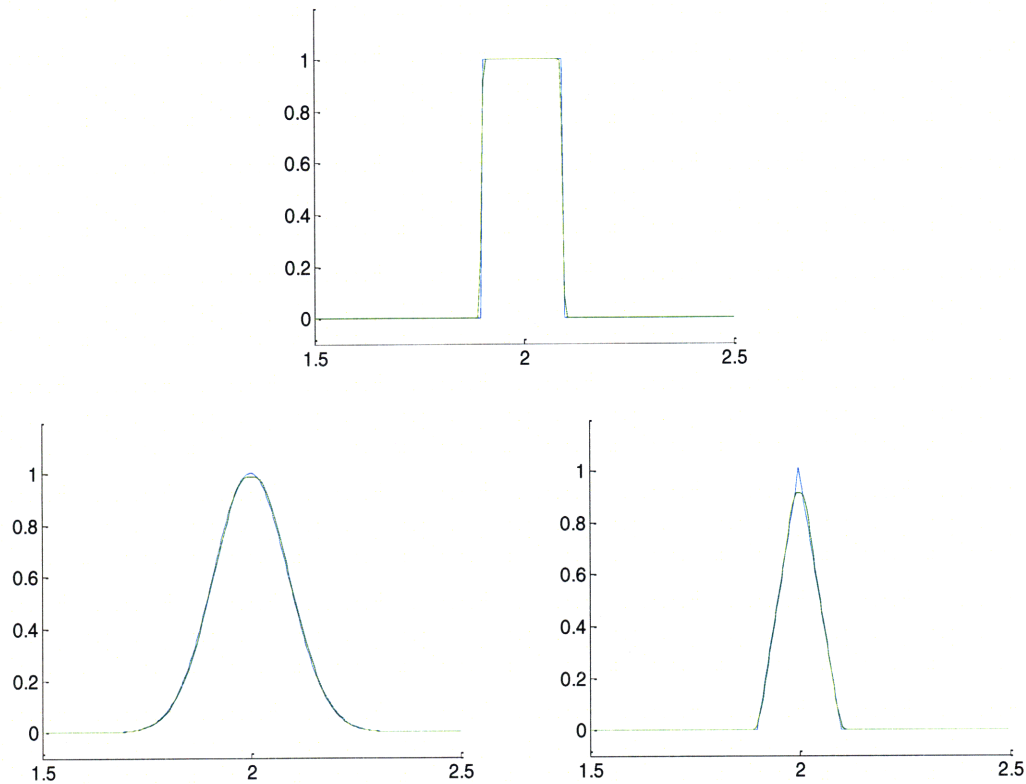
After the spatial discretization, the temporal integration must also be constructed so as to maintain positivity as well. The key factor when considering positivity in time is the Courant number at the most basic level. Later we will see a more thorough analysis of the limits on time step size in terms of stability domains but this is the method employed in this paper. In this case the requirements on the time step are

$$\tau \leq \frac{v_0 h}{|v|},$$

where  $\tau$  is the time step size and  $v_0$  is the critical Courant number,

$$v_0 = \frac{1}{1 + \delta/2}$$

For testing purposes, Runge-Kutta methods meeting the above requirements were employed to integrate the discretization (5.2) & (5.4) through time with time step based on 90% of the maximum Courant number. The results are shown below. Additional results can be seen in Section 6.3.



**Figure 5.6: Concentration vs. Position, 3 profiles**

This plot shows the approximations (solid lines) for square ( $h=1/50$ ), smooth ( $h=1/150$ ), and triangular ( $h=1/250$ ) profiles using the scheme of Hundsdorfer.

Explicit Runge-Kutta methods from order 1 (which is just the forward Euler method) to order 4 (the classical Runge-Kutta method) all showed similar results, with the fourth order method being the most accurate, as expected. However, the relative advantage was fairly small. This indicates that the time step size restrictions from the Courant number are greater than what is normally needed to obtain sufficient accuracy in the temporal integration.

As expected, the square peak is easily reproduced, while the sinusoidal and triangular peaks require an increasing number of space points to obtain an accurate solution. However, the results are far superior to the simple upwind discretization while still maintaining positivity. The square profile is noteworthy in that there appears to be no Gibbs phenomenon (overshoot on the approximation of step-like profiles). This demonstrates that the underlying feature of the positivity preserving method is a prevention of over- and under-shoots.

### **5.3 Operator Splitting**

Ideally, the above large-domain explicit methods are able to solve the systems of interest. However, it is possible that there are some terms in equations of interest that have

characteristics that make implicit methods necessary. For example, in the general convection-diffusion-reaction equation,

$$\frac{\partial b_i}{\partial t} + \nabla \cdot \mathbf{v} b_i = \nabla \cdot D_i \nabla b_i + f(b_i, t),$$

the stiffness ratio of the convection and diffusion terms scale with the square of the number of grid points in the spatial discretization (see Section 2.2.7). This may remain a manageable amount within the context of the explicit methods mentioned above. However, the reaction terms can, depending on the reaction constants, have stiffness ratios that are several orders of magnitude larger than for the other terms.

Operator splitting techniques are designed to use different solution methods on each term. This can allow for the efficient solution of most of the equation with explicit methods while only employing implicit techniques where necessary, thereby resulting in a much more efficient solution. The key point is that solving the system of equation twice (or more) at each time point may be much more efficient than attempting to solve it only once but with a computationally intensive method.

The actual splitting can occur term-wise or even for each spatial dimension within a term. This is done to allow the one dimensional techniques to be easily applied to multidimensional systems. An interesting challenge presents itself when, for example, the matrix of diffusion coefficients contains off-diagonal terms, resulting in cross-derivative terms. This also occurs in systems such as equation (4.12). See [Mitchell & Griffiths, 1980] for more on this topic.

Of course, as with any other technique, there are several aspects that must be considered in the implementation. One of the primary challenges lies in the splitting error. If for a system of the form

$$w'(t) = Aw, \quad w(t_0) = w_0$$

the matrix  $A$  can be split into two terms,  $A = A_1 + A_2$ , an approximate solution would have the form

$$w_{n+1} = e^{\tau A_2} e^{\tau A_1} w_n.$$

This is inexact since for the exact solution we have

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

and the difference between the two can be seen by expanding the exponentials

$$e^{\tau A} = I + \tau (A_1 + A_2) + \frac{1}{2} \tau^2 (A_1 + A_2)^2 + \dots$$

$$e^{\tau A_2} e^{\tau A_1} = I + \tau (A_1 + A_2) + \frac{1}{2} \tau^2 (A_1^2 + 2A_2A_1 + A_2^2) + \dots$$

The second term in each expression is different unless the commutator,

$$A_1, A_2 = A_1A_2 - A_2A_1$$

is zero. Unfortunately, this discrepancy leads to an order consistency of only one, as the above expansion showed.

There are many different ways of recombining the matrices, notably the method of Strang,

$$w_{n+1} = e^{\frac{1}{2}\tau A_1} e^{\tau A_2} e^{\frac{1}{2}\tau A_1} w_n$$

which, due to a symmetric cancellation of some of the error terms, has a local truncation error is of order two.

The same logic applies to non-linear problems as well. For Strang splitting method, the form becomes

$$\frac{d}{dt} w^*(t) = F_1(t, w^*(t)), \quad t_n < t \leq t_{n+\frac{1}{2}}, \quad w^*(t_n) = w_n$$

$$\frac{d}{dt} w^{**}(t) = F_2(t, w^{**}(t)), \quad t_n < t \leq t_{n+1}, \quad w^{**}(t_n) = w^*(t_{n+\frac{1}{2}})$$

$$\frac{d}{dt} w^{***}(t) = F_1(t, w^{***}(t)), \quad t_{n+\frac{1}{2}} < t \leq t_{n+1}, \quad w^{***}(t_{n+\frac{1}{2}}) = w^{**}(t_{n+\frac{1}{2}})$$

which leads to  $w_{n+1} = w^{***}(t_{n+1})$ . This also has a consistency of order two.

When considering what type of splitting is to be used, the commutator should always be considered, since when it is identically zero, a simpler splitting method can be employed. For example consider a general PDE that can be split into two parts

$$w'(t) = f_1(w) + f_2(w).$$

The commutator is then defined as

$$f_1, f_2(w) = \frac{\partial f_1(w)}{\partial w} f_2(w) - \frac{\partial f_2(w)}{\partial w} f_1(w).$$

In most real-world situations, this is not zero, but when this is the case, it can be quite beneficial.

Boundary conditions usually cause the greatest amount of error when attempting to implement a splitting scheme, often resulting in a reduction of order in the method. Stiff terms can lead to similar issues. Consider the linear equation

$$w'(t) = Aw(t) + g(t) = A_1w(t) + g_1(t) + A_2w(t) + g_2(t)$$

where the boundary conditions for each of the two main terms are contained within the  $g_i(t)$  terms. But when evaluating the commutator, it can be seen that there are other terms that manifest themselves:

$$F_1, F_2 (w) = A_1, A_2 w + A_1g_2 - A_2g_1.$$

Even if the  $A$  matrices commute, the total commutator may be large. For this total commutator to be zero depends on certain compatibility constraints about the specific boundary conditions being satisfied that are independent of the general problem.

Unfortunately, there remain still no general procedures to handle the error induced by boundary conditions. Knowledge of the specific situation being analyzed can allow the integration of some specific techniques into the actual code, but this of course limits the overall robustness of the method.

There are four major classifications of operator splitting methods that are of interest in this thesis: LOD, ADI, IMEX, and AMF. Each has advantages and drawbacks and the system being solved dictates which one will be the most useful.

Locally one dimensional (LOD) methods select a fixed low order one step method with a step size  $\tau$  and apply it to each of the split terms. The simplest scheme is the LOD Backward Euler method. For the nonlinear system

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

consider the splitting

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

The method is then

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

where the  $v_i$  terms are internal vectors between each full time step. These internal vectors do not give consistent approximations to the exact solution. This issue becomes

important when considering steady-states of a system, as this method cannot return the exact steady-state solutions. Despite the implicit basis for the internal steps in this function, the stability of this method requires product of the eigenvalues and the time step,  $\lambda_i\tau$  to have a modulus less than one.

There are several more advanced LOD methods based on the Crank-Nicolson method and the implicit trapezoid rule that follow the same basic pattern. These methods are of order two and some techniques allow boundary conditions to be handled without order reduction.

The original alternating direction implicit (ADI) method uses a splitting of only two terms,

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

It takes the form

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

It is from this method that the general class of ADI methods derives its name since the implicit use of  $F_1$  and  $F_2$  alternates between the two stages. In contrast to the LOD methods, the internal framework does give consistent approximations so steady-state systems can be solved exactly.

More advanced methods can handle a splitting of more than two terms while retaining the features listed above. The Douglas method actually has unconditional stability for two-term splitting, though it becomes more restrictive as the number of split terms increases.

There are several methods that can be characterized within the implicit-explicit (IMEX) framework. The IMEX- $\theta$  method illustrates the basic concept. In this method division is by portions of the entire function with implicit or explicit characteristics. Define  $F_0$  as the non-stiff portion and  $F_1$  as the stiff portion:

$$w'(t) = F(t, w(t)) = F_0(t, w(t)) + F_1(t, w(t)).$$

The time stepping rule is then

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

This is effectively a forward Euler step on the non-stiff portion and a theta method step on the stiff portion. Note that this only requires one time step, so IMEX methods can generally be applied to multi-step methods, which is not true of the above two method types. As expected from the underlying methods, this scheme is only of order one. Also,

there is some restriction on the stability domain that must be balanced between the two different portions.

IMEX methods can be extended in many different ways in the context of multi-step or Runge-Kutta methods. However, the stability analysis is ultimately very specific to the actual implementation. For examples, see [Kennedy & Carpenter, 2003] and [Zhang, 2004].

Approximate matrix factorization (AMF) methods employ a similar splitting scheme to the IMEX, but with  $s$  implicit terms

$$w'(t) = F(w(t)) = F_0(w(t)) + F_1(w(t)) + \dots + F_s(w(t)).$$

Note that this is based on an autonomous system, but a non-autonomous system can be easily modified so it can be written in the above form. The most basic example is the following:

$$w_{n+1} = w_n + I - \gamma\tau A_s^{-1} \dots I - \gamma\tau A_1^{-1} \tau F(w_n)$$

where

$$A_i = F'_i(w_n) + O(\tau), \quad i = 1, \dots, s$$

is an approximate Jacobian and  $\gamma$  is a free parameter. These methods retain the property of accurately representing the steady state.

Second order methods exist as well and are more useful in many situations. There are actually many such methods that maintain A-stability for the implicit portion, while exhibiting reasonable stability domains for the explicit portion.

## 5.4 High Dimensional Systems

In basic transport systems there is obviously a maximum of three dimensions. However, in many systems, the “spatial” coordinates can refer to any number of things that are not constrained to a set amount. For example, in population balances, there can be any number of attributes that are treated the analogously to the coordinates of space. In option pricing, the problem is very well known, as the underlying assets take the place of spatial variables. This “curse of dimensionality” can be an issue in terms of the achievability of storage and computation, as is outlined below.

Consider a general linear discretized expression for  $w$ ,

$$w' = Aw.$$

If  $m$  is the number of spatial discretization points and there is one species, the dimensions of  $A$  are  $m \times m$ . If there are  $r$  chemical species, there must be  $r$  times as many spatial discretization points so the matrix now has dimension  $rm \times rm$ . In terms of memory, the matrix now has  $r^2 m^2$  data points to store. If the evolution of the data is important then this storage requirement must be multiplied by the number of time steps.

The issue of additional attributes can present greater challenges. Basically, the  $A$  matrices considered so far correspond to the one dimensional case. When another dimension is added the number of terms in the spatial vector must be squared. This means that the number of terms in the matrix increases by the fourth power. For a system with  $k$  dimensions this results in an  $A$  matrix with an increase of  $2k^{\text{th}}$  power number of terms. So for a general system with  $m$  discretization points,  $r$  components and  $k$  dimensions, the resulting a matrix has up to  $(rm)^{2k}$  number of data points to be stored. And of course the lack of sparseness in integral equations (if all the dimensions have some integral evaluated over them) may require the storage of the full number of data points.

These challenges have resulted in different approaches to solve these types of problems. The most common choice is using some type of Monte Carlo simulation since the conversion rate is  $(rm)^{-1/2}$  regardless of the number of dimensions. More advanced methods that still use the finite difference/finite element framework have been developed as well, see e.g. [Berridge & Schumacher, 2004]. Finally, more advanced coding techniques exploiting parallel processors can help with this difficulty as well.



## 6.0 Spatial Discretization Methods

As the method of lines (MOL) approach is taken in this thesis, the spatial and temporal solution parts are discussed in different chapters. This one focuses on the former and the next chapter on the latter. Of course the interaction between the parts is non-trivial and is discussed in portions of this and later sections.

There are many practical details in the implementation of the spatial discretization. The overall goal of the method is to be able to easily adapt to as many different situations as possible where the “space” variable can in fact be many different things. This means it should be able to handle the inclusion of multiple species and multiple dimensions and other items of this nature. Also it needs to be able to handle entirely different systems such as population balances and financial models. Obviously there will be some large changes in the specific situations but the framework should be as compatible as possible. Most of the discussion here will focus on the transport problem with the understanding that many of the concepts have broader applications.

### 6.1 Implementation of the Positivity Preserving Method

Positivity preservation is an important consideration when developing numerical solutions. There are a few different methods that can achieve the desired properties of mass conservation, efficiency, etc. The choice employed in this work is based upon the method of [Hundsdofer, 1998]. This method was chosen since it fits in well with the finite difference framework and it is possible to adapt it to more advanced situations such as variable velocity and multiple dimensions.

This method was discussed in Section 5.2 in some detail. Here a similar method is developed from a different starting point to result in what is (hopefully) a more lucid final form.

Recall that the positivity preservation is needed on the advection equation,

$$\frac{\partial b}{\partial t} + \frac{\partial vb}{\partial x} = 0$$

where  $b$  in discretized form corresponds to  $w$ . To more easily incorporate variable velocity, flux is chosen as the variable of interest rather than concentration. As was noted in Section 2.2.2, mass conservation requires that the flux form of the standard discretization be used so this offers another incentive to work with flux. The flux form is

$$w'_j(t) = \frac{1}{h} f_{j-1/2}(t, w(t)) - f_{j+1/2}(t, w(t)) \quad (6.1)$$

where the values for  $f_{j\pm 1/2}$  refer to the flux between two cells, or halfway between adjacent grid points. In general, flux is simply the product of velocity and concentration at a given

point. This form can actually represent all of the standard spatial discretizations. For example the first order upwind discretization has the flux values

$$\begin{aligned} f_{j-1/2}(t, w) &= \max(v_{j-1/2}, 0)w_{j-1} + \min(v_{j-1/2}, 0)w_j \\ f_{j+1/2}(t, w) &= \max(v_{j+1/2}, 0)w_j + \min(v_{j+1/2}, 0)w_{j+1} \end{aligned}$$

where  $v_{j\pm 1/2} = v(x_{j\pm 1/2}, t)$ . Recall the maximum and minimum functions are necessary to account for negative velocities.

The upwind scheme remains positive for any scheme but is plagued by artificial diffusion. Any other higher order schemes can result in spurious oscillations as discussed in Section 5.2. These effects show up in regions of strong variation which can be characterized by the slope ratios

$$r_j = \frac{w_j - w_{j-1}}{w_{j+1} - w_j}, \quad r_{j-1} = \frac{w_{j-1} - w_{j-2}}{w_j - w_{j-1}}, \quad r_{j+1} = \frac{w_{j+1} - w_j}{w_{j+2} - w_{j+1}}$$

Other standard discretizations can be handled in the same form via the  $\kappa$ -method of [van Leer, 1985]. The values of 1, -1, and  $\frac{1}{3}$  for  $\kappa$  correspond, respectively, to the second order central, second order upwind, and third order upwind biased spatial discretizations. This method is expressed in terms the slope ratio which can allow us to see where the higher order methods cause negative values. The  $\kappa$  and  $r$  dependencies can be expressed through the function  $\phi$  as

$$\phi(r) = \frac{1}{4}(1 + \kappa) + \frac{1}{4}(1 - \kappa)r$$

and  $\phi$  is placed in the flux equations as

$$\begin{aligned} f_{j-1/2}(t, w) &= v_{j-1/2} \left[ w_{j-1} + \phi(r_{j-1})(w_j - w_{j-1}) \right] \\ f_{j+1/2}(t, w) &= v_{j+1/2} \left[ w_j + \phi(r_j)(w_{j+1} - w_j) \right] \end{aligned}$$

for positive velocities and

$$\begin{aligned} f_{j-1/2}(t, w) &= v_{j-1/2} \left[ w_j + \phi(r_j^{-1})(w_{j-1} - w_j) \right] \\ f_{j+1/2}(t, w) &= v_{j+1/2} \left[ w_{j+1} + \phi(r_{j+1}^{-1})(w_j - w_{j+1}) \right] \end{aligned}$$

for negative velocities. Note that for the negative velocities the arrangement of the bracketed terms is “reflected” around the  $j_{\pm 1/2}$  point relative to the positive velocity version.

Removing the second term in the brackets of the above expressions reduces the equations back to the upwind differencing method. Therefore this second term can be thought of as a correction to the artificial diffusion which plagues the basic method. In this context, if this correction is too large it can result in negative values. Tempering the effect is known as flux limiting and it amounts to bounding  $\varphi(r)$  by prescribed values.

To see how negativity can show up, we write out (4.14) with the full fluxes. First assume that velocity is constant for simplicity.

$$\begin{aligned} w'_j(t) &= \frac{1}{h} v \left[ w_{j-1} + \varphi(r_{j-1})(w_j - w_{j-1}) \right] - v \left[ w_j + \varphi(r_j)(w_{j+1} - w_j) \right] \\ &= \frac{v}{h} \left[ 1 - \varphi(r_{j-1}) - \frac{1}{r_j} \varphi(r_j) \right] (w_{j-1} - w_j) \end{aligned}$$

This makes it apparent that the bracketed term must be greater than zero for the solution to remain positive. Indeed this actually guarantees positivity as shown in [Hundsdorfer & Verwer, 2003 (p.116)]. This can be ensured with the following bounds,

$$0 \leq \varphi(r) \leq 1, \quad 0 \leq \frac{1}{r} \varphi(r) \leq \delta$$

where  $\delta$  is a parameter that can take any positive value. Making it too large can increase the eigenvalues of the problem domain and effectively limit the time integration. To show this, first express the bracketed term as  $\gamma_j(w)$ . Rearranging the positivity bounds shows that the restriction for  $\gamma$  is

$$0 \leq \gamma(w) \leq 1 + \delta.$$

Now recall the explicit Euler method and apply it to this problem,

$$w_j^{n+1} = w_j^n + \tau \frac{v}{h} \gamma_j(w^n) (w_{j-1}^n - w_j^n)$$

To ensure stability we note that this is effectively an upwind approximation multiplied by an extra term that has a maximum value of  $1 + \delta$ . Therefore the stability restriction here is

$$v \equiv \tau \frac{v}{h} \leq \frac{1}{1 + \delta}$$

where  $v$  is the Courant number. In the case where variable velocities are involved equation (4.14) must be expanded for all the possible combinations of signs of  $v_{j-1/2}$  and  $v_{j-1/2}$  and compared to ensure positivity. It turns out that the restriction then becomes

$$\tau \frac{|v_{j\pm 1/2}|}{h} \leq \frac{1}{1+\delta} \quad \text{and} \quad \tau \frac{|v_{j+1/2} - v_{j-1/2}|}{h} \leq \frac{1}{1+\delta}$$

where the first restriction applies to velocities with the same sign and the second applies to the case of opposite signs.

If it is assumed that the value of  $\gamma_j(w)$  remains constant between every given interval  $t_n \leq t \leq t_{n+1}$ , it can be shown that the same Courant number restriction applies to all Runge-Kutta methods with number of stages and order equal (see Section 2.2.8). In practice the actual eigenvalues of the discretized and filtered system may need to be considered to ensure stability in more advanced time integration methods. This will be discussed further in the next chapter. Overall, it turns out that 1 is generally an acceptable choice for  $\delta$  to allow for stability.

With these restrictions on the function  $\phi$ , it can be restated as follows

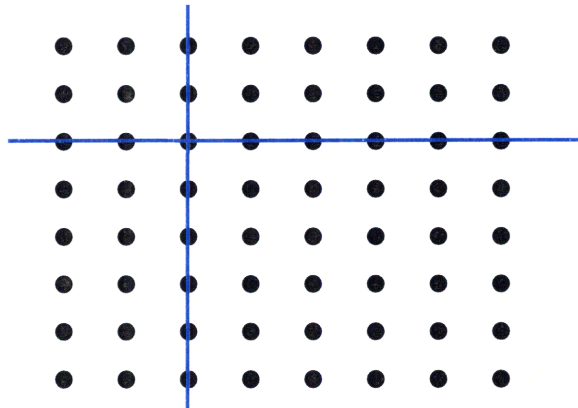
$$\phi(r) = \max \left( 0, \min \left( \delta, \frac{1}{4}(1+\kappa) + \frac{1}{4}(1-\kappa)r, r \right) \right)$$

For the actual implementation of this filter there are a few practical considerations. First off, the ratio  $r$  has many situations that could result in dividing by zero. One solution is to add a small perturbation to the denominator. This has the disadvantage of introducing more error into the system and can still result in small negative values. The method adopted in this work is to consider each case that could result in divide by zero errors and catch them in the program and assign them a value based on certain conventions. For example, zero divided by zero becomes one and anything else divided by zero is infinity times the sign of the numerator, where infinity is a value defined either by a built-in part of the programming language or as some constant in the program.

Additionally, there is a directionality associated with the method in terms of time. If the integration occurs in the direction of decreasing time the method needs to be altered. This can be accomplished most easily by changing the sign of the velocities right before the method and then reversing the sign for all the results.

Some care also needs to be taken when considering the boundaries since the formulae depend upon values as far as 2 grid points above and below the current point. This is also important when handling the boundary conditions. The way that these issues have been handled in this work is to revert to the upwind method at the extreme grid points. This allows for the boundary conditions to be applied in the same manner as in the non-limited case.

Application of positivity preservation to the multidimensional case is fairly straightforward to explain. Along each row of each dimension the same process as the one-dimensional case is run. For example, in the 2-D case below the positivity algorithm would be run along each column with respect to the points above and below a given point, and then along each row with respect to the points left and right of a given point.



While this concept is quite simple the implementation is quite challenging, especially when written to allow for an arbitrary number of dimensions. The details are discussed in the next section.

## 6.2 More Details of the Implementation

Almost all of the standard methods are based around one dimensional one component models. Theoretically it is easy to scale up these implementations to more complicated systems. However there are many practical details that must be considered.

The arrangement of the  $w$  vector is of great significance to the efficiency of the solution method. This vector must contain the quantity of interest for every species at every grid point in the solution space. To start out consider the case when there are multiple species in the system. The best way to arrange them in the vector is to have all the species at each grid point be listed successively, followed by all the species at the next grid point. So if the species are A, B, and C, and the grid points are denoted with  $j$  then the vector would have the form

$$w = [\dots \ w_{j-1}^A \ w_{j-1}^B \ w_{j-1}^C \ w_j^A \ w_j^B \ w_j^C \ w_{j+1}^A \ w_{j+1}^B \ w_{j+1}^C \ \dots]^T .$$

This arrangement is advantageous when there are reactions with disparate coefficients that result in stiff systems. The reaction system at each grid point can be handled individually with an implicit method thus avoiding the need to invert the entire matrix. This is discussed further in Section 7.1.4.

The multi-dimensional case presents more of a challenge. These systems can be very large as the number of grid points increases as the power of the number of dimensions and all of the points must be arranged in the  $w$  vector. First off consider a two-dimensional system  $(x,y)$ . Say there are  $m_x$  grid points in the  $x$ -direction indexed by  $i_x$  and  $m_y$  grid points in the  $y$ -direction indexed by  $i_y$ . There many ways to index the grid points to the  $w$  vector, for example:

$$w = (x_1, y_1) \ (x_1, y_2) \ (x_2, y_1) \ (x_1, y_3) \ (x_3, y_1) \ \dots^T$$

To choose the most efficient one we consider the arrangement that the coefficients would need to take in the  $A$  matrix. We want the points labeled in some regular pattern so that points that are adjacent in the grid are as close as possible in the  $w$  vector. This is useful because it makes the coefficients in the  $A$  matrix as close to the diagonal as possible. The closer an matrix is to being diagonal the more efficient it is to invert if that is necessary in the solution process. It turns out that the best arrangement to meet these conditions is

$$w = \left[ (x_1, y_1) \ (x_1, y_2) \ \dots \ (x_1, y_{m_y}) \ (x_2, y_1) \ \dots \ (x_2, y_{m_y}) \ \dots \ (x_{m_x}, y_{m_y}) \right]^T$$

which can be obtained with the indexing rule, where  $j$  is the index of  $w$

$$j = (i_x - 1) \times m_y + i_y.$$

The challenge now is to figure out how to apply this to an arbitrary number of dimensions. Let us consider a case with four dimensions,  $(x, y, z, a)$  where the number of grid points and index names for the new dimensions are  $m_z$  and  $i_z$  and  $m_a$  and  $i_a$ , respectively. The same basic pattern expanded to four dimensions can be achieved with the following indexing rule

$$j = (i_x - 1) \times m_y m_z m_a + (i_y - 1) \times m_z m_a + (i_z - 1) \times m_a + i_a$$

The overall pattern is evident at this point and can be expanded to  $d \geq 2$  dimensions as

$$j = \sum_{i=1 \dots i_{d-1}} \left( (i-1) \prod_{m=m_d \dots m_{i+1}} m \right) + i_d. \quad (6.2)$$

where  $j$  is the index of the  $w$  vector corresponding the to the grid point  $(i_1, \dots, i_d)$  in the problem space.

To ensure a clear implementation in an actual program it is worthwhile to explore a few ways to think about how to visualize the set up. The key challenge of using these indices is setting up all of the discretization coefficients via the  $A$  matrix or positivity formula.

To better understand the patterns that show up let us first consider a three dimensional system  $(x, y, z)$ . Each dimension is discretized with a stencil width of three; for compactness of notation denote the discretization coefficients

$$x_-, x_0, x_+$$

and the other dimensions similarly. Let the system be defined with only 3 grid points in each dimension so that there are a total of 27 grid points in the problem space. A portion of the  $A$  matrix then looks as follows:

$$\begin{array}{cccccccccc}
 D & z_+ & - & y_+ & - & - & - & - & - & x_+ \\
 z_- & D & z_+ & - & y_+ & - & - & - & - & - \\
 - & z_- & D & - & - & y_+ & - & - & - & - \\
 y_- & - & - & D & z_+ & - & y_+ & - & - & - \\
 - & y_- & - & z_- & D & z_+ & - & y_+ & - & - \\
 - & - & y_- & - & z_- & D & - & - & y_+ & - \\
 - & - & - & y_- & - & - & D & z_+ & - & - \\
 - & - & - & - & y_- & - & z_- & D & z_+ & - \\
 - & - & - & - & - & y_- & - & z_- & D & - \\
 x_- & - & - & - & - & - & - & - & - & D
 \end{array}$$

where  $D = x_0 + y_0 + z_0$ . Using the formula (6.2) to assign values to the  $w$  vector it is seen that multiplying the vector by this  $A$  matrix will result in the coefficients being multiplied by their appropriate corresponding values.

In practice in an actual program the conversion formula can be used at the beginning and end so the problem can be set up as an array of arbitrary dimension and then converted for the solution procedure. This does, however, make some of the logic inside the program more difficult.

For example, in the iterative portion of the program it is most efficient to cycle through each dimension when assigning values. Because of this it is helpful to assign index values that map the adjacent points in solution space to the corresponding values from the  $w$  vector (using a formula based on (6.2)) at the beginning of the loop. This allows for the rest of the code in the loop to work with the same indexed variables. For example, looking at the above  $A$  matrix portion it is seen that one point upwind of the  $z$ -dimension is 1 point away, for  $y$  its upwind point is 3 points away and for  $x$  it is 9 points.

These same ideas work well for the more complicated flux-based positivity preserving methods. Overall, this same logic is used throughout the program to allow the any number of dimensions to be handled using the same code.

### 6.3 Results

The implementations of these ideas in various contexts are discussed in a bit more detail in the following chapters. As mentioned above there are relatively few changes in the basic structure. These methods are tested along with the Runge-Kutta-Chebyshev methods developed in the next chapter. The results of problems solved using these methods are discussed together.

## 7.0 Implementation of the Runge-Kutta Chebyshev Method

A time integration scheme based on the one developed in [Sommeijer & Verwer, 2006] (which is itself based on several earlier works) has been implemented in MATLAB. Given below is a detailed explanation of the method as well as a discussion of its practical uses and performance.

### 7.1 Detailed Development of the RKC Method

#### 7.1.1 Standard RKC

As discussed in Section 5.1, the Runge-Kutta Chebyshev method is based on using internal stages to increase the stability domain along the real axis. For most applications accuracy order of at least two is needed so the second order method will be considered from here on. The stability domain of the second order perturbed method is the following:

$$P_s(z) = 1 + \frac{T_s''(\omega_0 + \omega_1 z)}{T_s''(\omega_0)} T_s(\omega_0 + \omega_1 z) - T_s(\omega_0), \quad \omega_1 = \frac{T_s'(\omega_0)}{T_s''(\omega_0)}$$

where  $T_s$  represents a Chebyshev polynomial of degree  $s$  and  $\omega_0 > 1$  is a parameter often chosen to be  $\omega_0 = 1 + \varepsilon / s^2$  where the damping coefficient  $\varepsilon$  is some small positive number.  $\omega_1$  is chosen to maintain second order consistency. To consider the actual formula for the integration scheme, it is first necessary to discuss a bit more about the internal stages.

Recall the basic Runge-Kutta form discussed in Section 2.2.8:

$$w_{n+1} = w_n + \tau \sum_{i=1}^s b_i k_i$$

$$k_i = F \left( t_n + c_i \tau, w_n + \tau \sum_{j=1}^s a_{ij} k_j \right)$$

where the  $a$ ,  $b$ , and  $c$  represent the coefficients from the used in the method. To consider the stability of the internal stages, the above method can be rewritten in the following form for explicit methods:

$$w_{n0} = w_n$$

$$w_{nj} = w_n + \tau \sum_{k=0}^{j-1} a_{jk} F(t_n + c_k \tau, w_{nk}), \quad j = 1, \dots, s$$

$$w_{n+1} = w_{ns}$$



where  $w_{nj}$  indicates the value of  $w$  at time step  $n$ , stage  $j$ . Note that  $a$  and  $c$  (and later  $b$ ) have been redefined to include a value at index zero which does not correspond to anything on the standard Butcher array. For the  $c_s$ , this is actually just a shift down one of every index. The new  $a$  values amount to shifting the  $A$  portion of the butcher array down to include the  $b$  row at the bottom (basically shift both indices of  $a$  down one). This modified butcher array now looks like this:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & & a_{2s} \\ \vdots & \vdots & & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \Rightarrow \begin{array}{c|ccc} c_0 & & & \\ c_1 & a_{10} & & \\ \vdots & \vdots & & \\ c_{s-1} & a_{s-1,0} & a_{s-1,1} & \cdots \\ \hline & a_{s,0} & a_{s,2} & \cdots & a_{s,s-1} \end{array}$$

To determine these coefficients for the Runge-Kutta Chebyshev method, the internal stability must be analyzed.

Consider the general linear system

$$w'(t) = Aw(t) + g(t), \quad w(t_0) = w_0.$$

Also consider a perturbed version of the above explicit solution method,

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

where the  $r_j$  represents local perturbations, e.g. round-off errors. Step-wise and stage-wise errors can be denoted, respectively,

$$\begin{aligned} e_n &= \tilde{w}_n - w_n \\ e_{nj} &= \tilde{w}_{nj} - w_{nj} \end{aligned}$$

The error for each stage can be written in terms of matrix polynomials  $R_j$  and  $Q_{k-j}$  (the subscripts indicate the degree of the polynomial) as

$$e_{nj} = R_j \tau A e_n + \sum_{k=1}^j Q_{j-k} \tau A r_k, \quad j = 1, \dots, s$$

so the total error for a step,  $e_{n+1} = e_{ns}$ , is

$$e_{n+1} = R_s(\tau A) e_n + \sum_{j=1}^s Q_{s-j}(\tau A) r_j.$$

This equation captures the error for a single step.  $R$  is the standard stability polynomial defined in Section 2.2.6 that relates the error propagation from step to step and  $Q$  is the internal stability polynomial. The internal stability refers to the accumulation of stage perturbations,  $r_j$ , within a single step. Assuming  $A$  is a normal matrix, the error norm for each step can be bounded by

$$\|e_{n+1}\| \leq \max_{z=\tau\lambda} |R_s(z)| \|e_n\| + \sum_{j=1}^s \max_{z=\tau\lambda} |Q_{s-j}(z)| \|r_j\|$$

where  $\lambda$  is an eigenvalue of  $A$  and  $\|\cdot\|$  is some vector norm. Even if the error is acceptably small with respect to  $R$ , error can quickly accumulate on the internal stages through  $Q$ . This becomes especially problematic in cases with a large number of stages such as the Runge-Kutta Chebyshev methods.

Both  $R$  and  $Q$  are of course determined by the coefficients of the method. When choosing coefficients in a manner different than the method of Butcher (see Section 2.2.8) it becomes important to address the internal stability issue. A natural choice for  $Q$  is to make its stability domain the same as (or similar to) that of  $R$  so that same stability required of the overall method is sufficient at each stage as well. [Houwen & Sommeijer, 1982] demonstrated that if the stability domain  $R$  is defined by coefficients as

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

(note that this  $a$  and  $b$  are different than those defined previously) then the  $Q$  polynomials will be defined by the recursion

$$Q_{s-j}(z) = \frac{b_j}{b_s} U_{s-j}(\omega_0 + \omega_1 z), \quad j = 1, \dots, s$$

where  $U_i$  is the  $i^{\text{th}}$  degree Chebyshev polynomial of the second kind. The bounds of this type of polynomials are similar to the polynomials of the first kind and it can be shown that the error is bounded by

$$\|e_{n+1}\| \leq \|e_n\| + \frac{1}{2} s(s+1) K \max_j \|r_j\|$$

where  $K$  is some constant of moderate size independent of  $A$ ,  $\tau$ , and  $s$ , as long as  $z = \tau\lambda$  stays within the stability domain defined by  $R$ . Basically this means that the error  $r$  grows at most quadratically with the number of stages. See [Hundsdorfer & Verwer, 2004] for more details.

With  $R_j$  defined by the Chebyshev polynomials as above, it is now possible to determine the coefficients  $b_j$ . Since the Chebyshev polynomials can be defined by the recursion,

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \quad 2 \leq j \leq s, \quad x \in \mathbb{C}$$

it follows that  $R$  can also be defined recursively. We know that  $R_j(0) = 1$  for all  $j$ . Directly calculating the first two  $R$  polynomials gives

$$\begin{aligned} R_0(z) &= 1 - b_0 T_0(\omega_0) + b_0 T_0(\omega_0 + \omega_1 z) = 1 - b_0 + b_0 = 1, \\ R_1(z) &= 1 - b_1 T_1(\omega_0) + b_1 T_1(\omega_0 + \omega_1 z) = 1 + b_1 \omega_1 z \end{aligned}$$

and we can now apply the recursion:

$$\begin{aligned} R_j(z) &= 1 - b_j [2\omega_0 T_{j-1}(\omega_0) - T_{j-2}(\omega_0)] + b_j [2(\omega_0 + \omega_1 z) T_{j-1}(\omega_0 + \omega_1 z) - T_{j-2}(\omega_0 + \omega_1 z)] \\ &= 1 - \frac{2b_j \omega_0}{b_{j-1}} + \frac{b_j}{b_{j-2}} + \left( \frac{2b_j \omega_0}{b_{j-1}} + \frac{2b_j \omega_1}{b_{j-1}} z \right) [b_{j-1} T_{j-1}(\omega_0) - b_{j-1} T_{j-1}(\omega_0 + \omega_1 z)] \\ &\quad - \frac{b_j}{b_{j-2}} [b_{j-2} T_{j-2}(\omega_0) - b_{j-2} T_{j-2}(\omega_0 + \omega_1 z)] - \frac{2b_j \omega_1}{b_{j-1}} [1 - b_{j-1} T_{j-1}(\omega_0)] \\ &= 1 - \mu_j - \nu_j + \mu_j R_{j-1}(z) + \nu_j R_{j-2}(z) + \tilde{\mu}_j R_{j-1}(z) + \tilde{\gamma}_j z \end{aligned}$$

where the new coefficients are

$$\tilde{\mu}_j = b_j \omega_1, \quad \mu_j = \frac{2b_j \omega_0}{b_{j-1}}, \quad \nu_j = \frac{-b_j}{b_{j-2}}, \quad \tilde{\mu}_j = \frac{2b_j \omega_1}{b_{j-1}}, \quad \tilde{\gamma}_j = -a_{j-1} \tilde{\mu}_j.$$

If  $R_s$  is the earlier-defined polynomial  $P_s$ , then  $\omega_0$  and  $\omega_1$  are as defined above. To define the  $b_j$  coefficients for a second order formula, one method is to ensure that all of the internal stages remain second order consistent. To do this, first do a Taylor series expansion of  $R_j$  about the origin. This gives

$$R_j(z) = 1 + b_j \omega_1 T_j'(\omega_0) z + \frac{1}{2} b_j \omega_1 T_j''(\omega_0) z^2 + O(z^3)$$

which must match the expansion for  $e^{cz}$ ,

$$e^{cz} = 1 + cz + \frac{1}{2} cz^2 + O(z^3)$$

for each component to the second term so we have

$$b_j = \frac{T_j''(\omega_0)}{T_j'(\omega_0)^2}.$$

This gives, with the damping parameters discussed above, the following stability function for each internal stage:

$$R_j(z) = 1 + \frac{T_j''(\omega_0 + \omega_1 z)}{T_j'(\omega_0)^2} T_j(\omega_0 + \omega_1 z) - T_j(\omega_0), \quad j = 2, \dots, s \quad (7.1)$$

The first stage does not have high enough powers of  $z$  to support any higher than first order, so there is more freedom in defining the values of  $b_0$  and  $b_1$ . The following convention is used,

$$b_0 = b_2, \quad b_1 = 1/\omega_0,$$

which will be discussed further below.

Now that all of the coefficients are specified the actual formula for each stage can be written. By noting that by the definition of the stability function,

$$w_{nj} = R_j(z)w_n \quad \text{and} \quad \tau F_j = zw_{nj}$$

where

$$F_j = F(t_n + c_j \tau, w_{nj}),$$

the formula for each internal stage is as follows:

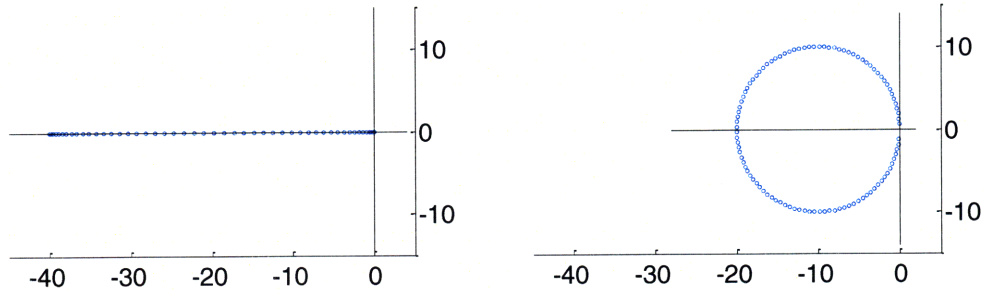
$$\begin{aligned} w_{n0} &= w_n \\ w_{n1} &= w_{n0} + \tilde{\mu}_1 \tau F_0 \\ w_{nj} &= (1 - \mu_j - \nu_j)w_{n0} + \mu_j w_{n,j-1} + \nu_j w_{n,j-2} + \tilde{\mu}_1 \tau F_{j-1} + \tilde{\nu}_j \tau F_0 \\ w_{n+1} &= w_{ns} \end{aligned} \quad (7.2)$$

This is the basic form of the Runge-Kutta Chebyshev method. However, its usefulness is limited to simple diffusion problems. The next consideration is for the case when advection is present as well.

### 7.1.2 RKC for Advection-Diffusion

The most important difference between advection and diffusion problems from a numerical solution perspective is the location of the eigenvalues that result from spatial discretization. While the eigenvalues resulting from the discretization matrices for second-order or fourth-order approximations of the second derivative extend along the real axis, the commonly used discretizations for the advection term lead to eigenvalues

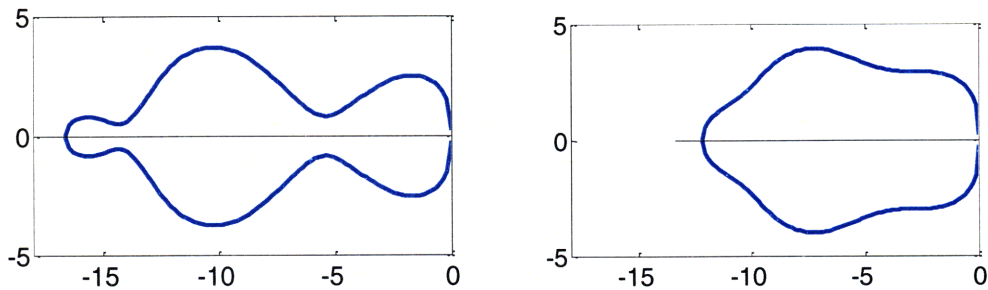
with imaginary values. Consider a case with  $d = 0.001$  approximated with a second-order approximation and a case with  $\nu = 0.1$ , both with 100 grid points:

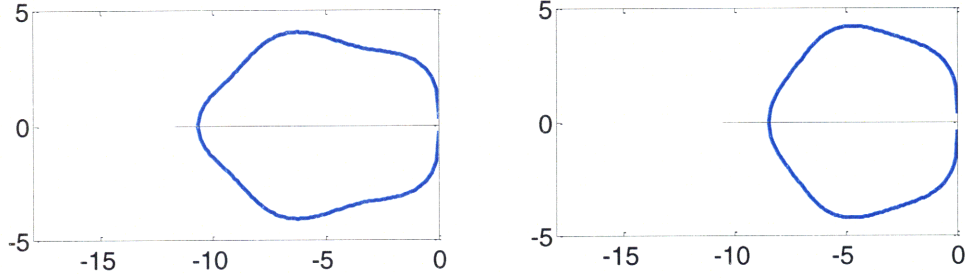


**Figure 7.1: Eigenvalues for typical diffusion (left) and advection problems**

Recalling the stability domain provided by the standard RKC method (*cf.* Figure 5.3), it is clear that the standard Runge-Kutta Chebyshev method would need very small time steps to capture the imaginary eigenvalues thus negating the primary benefit of the method.

The only free parameter available for changing the shape of the stability domain is the damping coefficient,  $\varepsilon$ . Increasing it extends the stability domain in the imaginary direction at the expense of a reduction in stability along the real axis. After a certain point, however, increasing  $\varepsilon$  further has little effect on the shape of the stability domain as can be seen below.





**Figure 7.2: Effect of increasing  $\varepsilon$  on stability domain; 5 stages,  $\varepsilon = 2/13, 5, 10, 100$**

Modifying the stability domain to solve advection-diffusion problems with various spatial discretizations has been explored extensively in [Verwer *et al.*, 2004]. Their approach was to use Fourier analysis of the to determine the eigenvalues for the system being solved and then use a method conceived of by [Wesseling, 2001] to determine critical values for the time step to ensure that the eigenvalues remain within various geometric shapes. These shapes were then inscribed in the stability domain so that the critical time step for stability could be determined with simple algebraic equations.

This method was advantageous for many advection-diffusion problems but it was still inadequate for systems with eigenvalues near the imaginary axis and completely useless for systems with purely imaginary eigenvalues, such as centrally-based discrete approximations of the first derivative. To resolve this difficulty an enhanced two-step method was developed by [Sommeijer & Verwer, 2006].

The two step method can be written as follows

$$w_{n+1} = \alpha_{-1}w_{n-1} + \alpha_0w_n + \alpha_\eta\tilde{w}_{n+\eta} \quad (7.3)$$

where the  $\alpha$ s are determined to achieve maximal order and  $\eta$  is a factor adjusting the time step size.  $\tilde{w}_{n+\eta}$  represents a normal one-step Runge-Kutta Chebyshev step taken with time step size  $\eta\tau$ .

To satisfy second order consistency the  $\alpha$ 's must be adjusted based on a Taylor series expansion of the all the terms involved. Note that the expansion of the RKC step is as follows

$$\tilde{w}_{n+\eta} = w + \eta\tau_n w' + \frac{1}{2}\eta^2\tau_n^2 w'' + \frac{1}{6}c\eta^3\tau_n^3 + O(\tau_n^4), \quad w = w(t_n).$$

Since it is only second-order accurate there is an additional term,  $c$ , in front of the third-order term. This can be found by expanding the equation for the polynomial of the stability domain,

$$P_s(z) = 1 - b_s T_s(\omega_0) + b_s T_s(\omega_0 + \omega_1 z)$$

and expanding it to give

$$\begin{aligned} P_s(z) &= P_s(0) + P_s'(0)z + \frac{1}{2}P_s''(0)z^2 + \frac{1}{6}P_s'''(0)z^3 + O(z^4) \\ &= 1 + z + \frac{1}{2}z^2 + \frac{1}{6} \frac{T_s''(\omega_0)T_s'''(\omega_0)}{T_s'(\omega_0)^2} z^3 + O(z^4) \end{aligned}$$

to determine the value for  $c$ .

Expanding the rest of the terms in equation (7.3) gives

$$\begin{aligned} w_{n+1} &= w + \tau_n w' + \frac{1}{2} \tau_n^2 w'' + \frac{1}{6} \tau_n^3 w''' + O(\tau_n^4) \\ w_n &= w + O(\tau_n^4) \\ w_{n-1} &= w + \tau_{n-1} w' + \frac{1}{2} \tau_{n-1}^2 w'' + \frac{1}{6} \tau_{n-1}^3 w''' + O(\tau_{n-1}^4) \end{aligned}$$

where  $w$  here means  $w(t_n)$ . Inserting these expansions into (7.3) shows that clever use of the  $\eta$  tern allows for order of up to three to be obtained. If the ratio  $r_n = \tau_n/\tau_{n-1}$  is defined then to obtain agreement for each power of  $\tau$ , the following algebraic conditions must be met:

$$\begin{aligned} 1 &= \alpha_0 + \alpha_{-1} + \alpha_\eta \\ r_n &= -\alpha_{-1} + \eta r_n \alpha_\eta \\ r_n^2 &= -\alpha_{-1} + \eta^2 r_n^2 \alpha_\eta \\ r_n^3 &= -\alpha_{-1} + c \eta^3 r_n^2 \alpha_\eta \end{aligned}$$

Solving this system of equations returns

$$\begin{aligned} \alpha_0 &= \frac{-1 - r_n + \eta r_n + \eta}{\eta} \\ \alpha_\eta &= \frac{1 + r_n}{\eta(1 + \eta r_n)} \\ \alpha_{-1} &= \frac{r_n^2(1 - \eta)}{1 + \eta r_n} \\ \eta &= \frac{r_n - 1 \pm \sqrt{r_n^2 - 2r_n + 1 + 4r_n c}}{2r_n c} \end{aligned}$$

where  $c$  is as defined above. Note that  $\eta$  must always be positive so the positive root is the correct choice in the final equation. It is important to note here that the first three

equations result in second order consistency for any system, non-linear or linear. The fourth condition guarantees stability only for linear differential problems. The conditions for second order consistency can be shown to be globally valid over the entire time integration for general non-linear problems. Beyond the second order, general non-linear problems can only be shown to be consistent locally, but not globally. That being said, there are many non-linear equations that would end up bring third-order consistent using this method, but there is no overall guarantee. Overall this third order consistency is more of a bonus rather than the primary purpose of using this method.

Indeed, the primary purpose of this two-step method is to achieve a more favorable stability domain. Firstly, recall the root condition requirement of the characteristic polynomial discussed in Section 2.2.9. The characteristic polynomial of the two-step method under consideration is

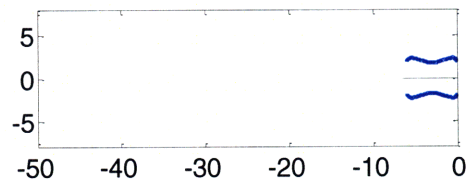
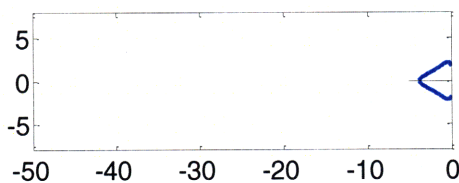
$$\pi(\zeta) = \zeta^2 - \alpha_0 + \alpha_\eta P_s(\eta z) \zeta - \alpha_{-1}$$

where  $P_s$  is the polynomial function of the  $s$ -stage Runge-Kutta Chebyshev method. This quadratic function has two distinct roots,

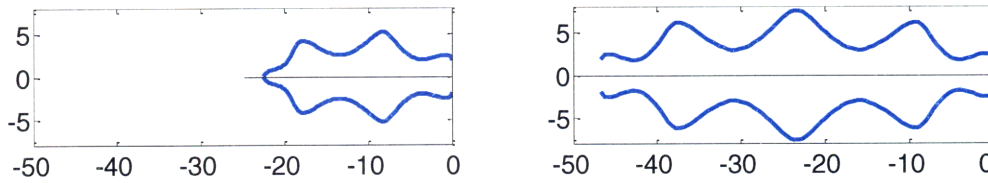
$$\zeta = \frac{\alpha_0 + \alpha_\eta P_s(\eta z) \pm \sqrt{\alpha_0^2 - 2\alpha_0 \alpha_\eta P_s(\eta z) + \alpha_\eta^2 P_s^2(\eta z) + 4\alpha_{-1}}}{2},$$

so the root condition requires  $|\zeta| \leq 1$  for each  $\zeta$ . If  $\eta$  is defined in terms of  $c$  as above then this equation has all the parameters known for a given time step ratio. Thus the stability domain can be solved for by plotting out in the complex plane the values of  $z$  satisfying this equation.

As mentioned above, the free parameter is the damping coefficient  $\varepsilon$ . This still remains the case for the above equation (assuming  $\eta$  is defined by  $c$ ). Using this parameter allows for the adjustment of the stability domain. Consider the examples shown in below:







**Figure 7.3: Stability domains for 2 step RKC; 3, 4, 7, 10 stages**

By adjusting  $\varepsilon$  a consistent upper bound on the imaginary height of the stability domain can be obtained for any number of stages.

As it turns out, there is a limit to how far the stability domain can be extended in the imaginary direction even as  $\varepsilon \rightarrow \infty$ . And the more  $\varepsilon$  is increased, the greater the decrease in stability along the real axis. For convenience the notation of [Sommeijer & Verwer, 2006] is employed which denotes the real and imaginary stability domains as  $\beta_{\text{Re}}(s)$  and  $\beta_{\text{Im}}(s)$ , respectively. Technically it is possible to extend  $\beta_{\text{Im}}(s)$  more by leaving  $\eta$  as a free parameter. However, it was found that this only increased  $\beta_{\text{Im}}(s)$  by around 10% at the expense of greatly reduced  $\beta_{\text{Re}}(s)$  [Sommeijer & Verwer, 2006] and then the third order consistency is lost.

The bounds on the stability domain bring up the concept of embedded shapes within the stability domain. Note that any of the stability domains discussed thus far, such as in Figure 5.3 have rather irregular boundaries. It is rather difficult to guarantee eigenvalues are contained in such regions, especially when most eigenvalues for advection-diffusion problems (and many other systems) are in elliptical or linear arrangements. The idea of embedding shapes was conceived of by [Wesseling, 2001] and applied to the Runge-Kutta Chebyshev method by [Verwer *et al.*, 2004].

The basic concept came from performing a Fourier-von Neumann analysis (see below) on the spatial discretization scheme and then determining what values for the physical coefficients and grid size will ensure the eigenvalues all remain within various simple geometric shapes. For the two-step method being discussed the most natural shape to embed is a rectangle. The dimensions of this rectangle are simply  $\beta_{\text{Re}}(s)$  by  $2\beta_{\text{Im}}(s)$ . Now it is very easy to ensure stability. All that is needed is for the magnitude of the real part of every eigenvector (multiplied by  $\tau$ ) to be less than or equal to  $\beta_{\text{Re}}(s)$  and the imaginary part to be less than or equal to  $\beta_{\text{Im}}(s)$ . If the eigenvalues of the system are known then the maximum value for the time step can be determined.

Since there is a limit to how large  $\beta_{\text{Im}}(s)$  can be made to be, it is logical to choose some reasonable target value and then use  $\varepsilon$  to make  $\beta_{\text{Re}}(s)$  as large as possible for a given

number of stages. The desired value for stability in the imaginary direction chosen by [Sommeijer & Verwer, 2006] was  $\beta_{\text{Im}}(s) = \sqrt{3}$ .

Of course for pure diffusion problems it makes sense to go back to using the one-step method with  $\varepsilon = 2/13$  and not worrying about the size of  $\beta_{\text{Im}}(s)$ . The underlying similarity between the one-and two-step methods allows this to be easily implemented algorithmically.

### 7.1.3 Mapping Out Eigenvalues

With the stability domain well defined it now remains to determine the locations of the eigenvalues of the differential equation system in the complex plane. There are a couple of ways to approach this problem. Recall the general system being solved:

$$w'(t) = F(t, w(t)) .$$

For the general non-linear case the Jacobian must be considered but ultimately the system that matters for the solver time steps is the familiar relationship

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

where the  $A$  matrix could be a function of time. The eigenvalues can be characterized in several ways. Obviously they can be found by many different numerical techniques, but with the stability region defined by the rectangular region described above the only concern is the largest real and imaginary parts of the eigenvalues. With this being the case there are two strong contenders for use in the next step.

One option is to use Gershgorin's Theorem. Simply put, Gershgorin's Theorem allows the definition of a region that will encompass all of the eigenvalues of a square matrix. A radius is defined by summing the off-diagonal elements of each row,

$$r_i = \sum_{\substack{j=1 \\ j \neq i}}^M |a_{ij}| .$$

Then the eigenvalues all lie within one of the regions defined by  $z : |z - a_{ii}| \leq r_i$  .

Practically this means that the upper bound for the real and imaginary eigenvalues can be determined by finding

$$\begin{aligned} & \min_i \text{Re}(a_i - r_i) \\ & \max_i \text{Im}|a_i \pm r_i| \end{aligned}$$

and defining  $\tau$  such that  $z$  remains contained in the stability boundary defined above. This approach would work for any system being solved as long as some square matrix  $A$  as defined above can be constructed.

However, it is possible to use knowledge of the spatial discretization used to define tighter bounds on the eigenvalues. Fourier analysis determines the eigenvalues of a system based on the propagation of Fourier modes induced by a given difference scheme. It technically only applies to linear systems with spatially periodic boundary conditions, but the results can be used to estimate the upper bounds for the eigenvalues on more complicated systems.

Recall the Fourier series, modes, and coefficients as they were expressed in Section 2.2.10. Now consider the semi-discrete system  $w'(t) = Aw(t)$  with an initial condition given by a single Fourier mode,  $w(0) = \varphi_k$ . The solution of this system is

$$\begin{aligned} w(t) &= e^{\lambda_k t} \varphi_k \\ w_j(t) &= e^{\lambda_k t} e^{2\pi i k x_j} \end{aligned}$$

where the  $\lambda_k$  is the eigenvalue of  $A$  corresponding to the Fourier mode and the  $j$  subscript indicates the element of the vector/grid.

For the advection equation, the exact PDE has the solution

$$w(x_j, t) = e^{-2\pi i k v t} e^{2\pi i k x_j} .$$

The eigenvalues of the exact solution,  $-2\pi i k v$ , now need to be compared to the eigenvalues of the approximation. To determine the eigenvalues of the approximate solution the Fourier mode is put into the spatial discretization scheme. The eigenvalues are then as follows:

**Table 7.1: Eigenvalues for common advection schemes**

1 <sup>st</sup> order upwind	
$w'_j(t) = \frac{v}{h} w_{j-1}(t) - w_j(t)$	$\lambda_k = \frac{ v }{h} \cos(2\pi kh) - 1 - \frac{iv}{h} \sin(2\pi kh)$
2 <sup>nd</sup> order central	
$w'_j = \frac{v}{2h} w_{j-1} - w_{j+1}$	$\lambda_k = -\frac{iv}{h} \sin(2\pi kh)$
2 <sup>nd</sup> order upwind	
$w'_j = \frac{v}{h} \left( -\frac{1}{2} w_{j-2} + 2w_{j-1} - \frac{3}{2} w_j \right)$	$\lambda_k = -4 \frac{ v }{h} \sin^4(\pi kh) - i \frac{v}{h} \sin(2\pi kh) 2 - \cos(2\pi kh)$
3 <sup>rd</sup> order upwind-biased	
$w'_j = \frac{v}{h} \left( -\frac{1}{6} w_{j-2} + w_{j-1} - \frac{1}{2} w_j - \frac{1}{3} w_{j+1} \right)$	$\lambda_k = \frac{-4 v }{3h} \sin^4(\pi kh) - \frac{i v}{3h} \sin(2\pi kh) 4 - \cos(2\pi kh)$
4 <sup>th</sup> order central	
$w'_j = \frac{v}{h} \left( -\frac{1}{12} w_{j-2} + \frac{2}{3} w_{j-1} - \frac{2}{3} w_{j+1} + \frac{1}{12} w_{j+2} \right)$	$\lambda_k = -\frac{i v}{3h} \sin(2\pi kh) 4 - \cos(2\pi kh)$
In all cases $j = 1, \dots, m$	In all cases $k = 1, \dots, m$

The diffusion equation can be handled similarly. Its exact solution is

$$w(x_j, t) = e^{-4\pi^2 k^2 dt} e^{2\pi i k x_j},$$

and the eigenvalues for the discretization schemes are as follows:

**Table 7.2: Eigenvalues for common diffusion schemes**

2 <sup>nd</sup> order central	
$w'_j = \frac{d}{h^2} w_{j-1} - 2w_j + w_{j+1}$	$\lambda_k = -\frac{4d}{h^2} \sin^2(\pi kh)$
4 <sup>th</sup> order central	
$w'_j = \frac{d}{h^2} \left( -\frac{1}{12} w_{j-2} + \frac{4}{3} w_{j-1} - \frac{5}{2} w_j + \frac{4}{3} w_{j+1} - \frac{1}{12} w_{j+2} \right)$	$\lambda_k = -\frac{1}{6} \frac{d}{h^2} \cos(4\pi kh) + \frac{8}{3} \frac{d}{h^2} \cos(2\pi kh) - \frac{5}{2} \frac{d}{h^2}$
In all cases $j = 1, \dots, m$	In all cases $k = 1, \dots, m$

With the eigenvalues so defined the maximum real and imaginary parts are easily found so it is straightforward to choose a time step  $\tau$  that will ensure the  $z$  values remain in the stability domains described above.

As noted above, these eigenvalues are only explicitly correct for systems with periodic boundary conditions and discretizations as described above. Using different boundary conditions will change the matrix as will using positivity preserving filters. The different boundary conditions are not too much of a problem since from a practical standpoint this just means going from a sparse circulant matrix to a matrix with the same elements near the diagonal, but without the zeros in the corners, as depicted in this example:

$$\begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix} \rightarrow \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}$$

Intuitively we might expect the circulant matrix to have larger eigenvalues than the non-circulant matrix. While this has been shown to be true for several test cases, it cannot be rigorously proven, despite much effort. For all of the cases tested, the largest real part and the largest imaginary part of the eigenvalues for the non-circulant matrix has been shown to be less than or equal to the corresponding values for the circulant version. Overall, using the same estimates for the eigenvalue range seems to be acceptable for the standard boundary conditions.

The greater challenge lies with the use of a non-linear filter to preserve positivity. The positivity preserving filter is discussed in Section 5.2. The important result for the current discussion is that after the application of the filter a new matrix is formed relating  $w$  and its derivative. So the new system is

$$w'(t) = A_+(w)w(t)$$

where  $A_+(w)$  indicates the new matrix, which is a function of  $w$ . Clearly the Jacobian is now different at every time step unlike in the standard linear case. The challenge with this new system is that there is no longer a convenient formula to calculate the eigenvalues of  $A_+$ . As discussed in earlier the filter operates only on the advective portion since it is these discretizations that can lead to non-positive results. As such, the positivity filter is still based on one of the given advection schemes so looking at the eigenvalues for the corresponding matrix from the non-filtered method can provide a rough idea of their location. Unfortunately, the range of the real and imaginary parts can be somewhat different for the filtered results.

For example, consider a simple sine hill as well as a sharply peaked function over some interval in space.

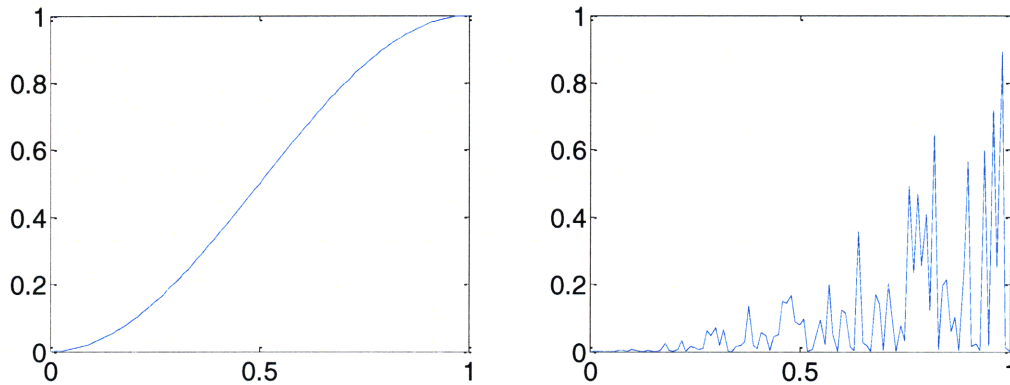


Figure 7.4: Example initial conditions, conc. vs. position

For both of these situations the positivity preserving filter will result in significant changes to the standard  $A$  matrix for a given advection scheme. To demonstrate the effect on the eigenvalues consider the following two cases. Case 1 sets  $d$  to zero,  $\nu = 0.1$  and  $h = 1/100$  (101 grid points). The initial condition is a sine hill. Case 2 keeps the same values for  $h$  and  $\nu$  but adds  $d = 0.001$  and gives the sharply peaked function mentioned above as its initial condition. All of the several advection schemes available show a change in the maximum real and imaginary eigenvalues. (Note that the 1<sup>st</sup> order upwind scheme is not seen since it is positivity preserving by construction.)

Table 7.3: Eigenvalues due to positivity filter

Adv. Scheme	Sine Hill		Random Peaks		
	Unfiltered	Filtered	Unfiltered	Filtered	
2 <sup>nd</sup> order central	0	10	39.99	70.14	Max Re Part
	10	9.99	10	10.89	Max Im Part
2 <sup>nd</sup> order upwind	39.86	38.85	79.86	63.32	Max Re Part
	21.96	21.35	21.95	10.17	Max Im Part
3 <sup>rd</sup> order upwind biased	13.29	13.8	53.28	68.24	Max Re Part
	13.71	13.68	13.71	10.45	Max Im Part

Clearly there are some eigenvalues resulting from the filtered version that are greater than their counterparts in the unfiltered version. If only the expected bounds from the unfiltered method were used to predict the bounds for the filtered version, instabilities would result. One interesting point is that the largest eigenvalue part of *all* the unfiltered schemes exceeds the largest eigenvalue part found in any of the filtered schemes. This makes some intuitive sense since the filtered matrix is essentially derived from combinations of the various advection schemes. Though this cannot be proven rigorously, several numerical tests have shown this same behavior. Overall it seems that using the global maxima of all the various advection schemes is a better estimator of the eigenvalues for a given filtered method than the method on which the filter is based.

To be absolutely safe, Gershgorin's Theorem is the only reliable way to completely bound the eigenvalues. But using the global maxima from all of the advection schemes seems to be another reliable option. Of course both of these methods will overestimate the true maximum eigenvalues by some amount thus requiring a smaller time step than is technically necessary for stability. The benefits of using the Fourier analysis estimates corresponding to the method underlying the filter (which often underestimates the maxima) and catching instabilities with the error correction method versus the more conservative methods just discussed in the next chapters.

Once the boundaries on the eigenvalues are satisfactorily set, the time step is chosen to ensure that  $z$  remains in the stability region. As mentioned previously the imaginary stability boundary of the two step Runge-Kutta Chebyshev method does not change greatly with  $\varepsilon$  so it is logical to choose a reasonable desired bound as the first step and determine  $\tau$  and the rest of the parameters. The bound chosen by [Sommeijer & Verwer, 2006] is  $\beta_{\text{Im}}(s) = \sqrt{3}$  for all  $s$ .

With  $\beta_{\text{Im}}(s)$  specified,  $\varepsilon$  can be adjusted to maximize  $\beta_{\text{Re}}(s)$  for a given  $s$ . Next,  $\tau$  can be chosen such that its product with the largest eigenvalues remains in the stability rectangle defined by  $\beta_{\text{Im}}(s)$  and  $\beta_{\text{Re}}(s)$ . If the restriction is found to depend on the imaginary boundary, then  $s$  can be reduced to encompass the largest real part.

#### 7.1.4 Reaction Problems and the IMEX Method

With advection-diffusion problems considered, it remains to discuss the inclusion of reaction terms. As has been mentioned in previous sections, the eigenvalues due to reaction terms can be arbitrarily stiff and this stiffness depends on the physics of the problem rather than the spatial discretization technique. As such there is generally a need to handle such terms implicitly.

In the two-step Runge-Kutta Chebyshev equation (7.3)  $\tilde{w}_{n+\eta}$  contains all of the implicit terms so it suffices to consider this part first. The equation is split as follows

$$w'(t) = F_E(t, w(t)) + F_I(t, w(t))$$

where the  $E$  and  $I$  subscripts refer to the explicit (advection-diffusion) and implicit (reaction) portions of the equation. Now consider the first step of the  $\tilde{w}_{n+\eta}$  portion which is simply the standard Runge-Kutta Chebyshev method with time step  $\eta\tau$ ,

$$w_{n1} = w_n + \tilde{\mu}_1 \eta \tau F_E(t_n, w_n) + \tilde{\mu}_1 \eta \tau F_I(t_n + \tilde{\mu}_1 \eta \tau, w_{n1})$$

where the constants are as defined in Section 7.1.1 and recall that the second subscript on  $w$  refers to the stage number. To determine stability, use the standard scalar test equation, this time defined as

$$w'(t) = \lambda_E w(t) + \lambda_I w(t). \quad (7.4)$$

Now the two scalars  $\lambda_E$  and  $\lambda_I$  correspond to the eigenvalues of the Jacobians the respective functions. Plugging in the  $\lambda$ 's to the equation for  $w_{n1}$  and rearranging gives

$$w_{n1} = R_1 \eta z_E, \eta z_I w_n$$

$$R_1 \eta z_E, \eta z_I = \frac{1 + b_1 \omega_1 \eta z_E}{1 - b_1 \omega_1 \eta z_I} = \frac{1 + \frac{\omega_1}{\omega_0} \eta z_E}{1 - \frac{\omega_1}{\omega_0} \eta z_I}$$

where  $z_E = \tau \lambda_E$  and  $z_I = \tau \lambda_I$ . Note that this is basically the stability region of the implicit Euler method. To achieve the stability function for the  $j^{\text{th}}$  term, the *ansatz* is made that the following is the correct form:

$$R_j \eta z_E, \eta z_I = 1 - b_j T_j(\omega_0) + b_j T_j \left( \frac{\omega_0 + \omega_1 \eta z_E}{1 - \frac{\omega_1}{\omega_0} \eta z_I} \right). \quad (7.5)$$

To see that this is in fact correct, the previous equation is written as

$$T_j(x) = \frac{-a_j}{b_j} + \frac{R_j}{b_j}, \quad a_j = 1 - T_j(\omega_0), \quad x = \frac{\omega_0 + \omega_1 \eta z_E}{1 - \frac{\omega_1}{\omega_0} \eta z_I}$$

and the Chebyshev recursion is inserted for  $T_j(x)$  to give

$$\frac{-a_j}{b_j} + \frac{R_j}{b_j} = 2 \frac{\omega_0 + \omega_1 \eta z_E}{1 - \frac{\omega_1}{\omega_0} \eta z_I} \left( \frac{-a_{j-1}}{b_{j-1}} + \frac{R_{j-1}}{b_{j-1}} \right) - \left( \frac{-a_{j-2}}{b_{j-2}} + \frac{R_{j-2}}{b_{j-2}} \right)$$

which can be rearranged to give

$$R_j \left( 1 - \frac{\omega_1}{\omega_0} \eta z_I \right) = a_j \left( 1 - \frac{\omega_1}{\omega_0} \eta z_I \right) + 2 \frac{b_j}{b_{j-1}} R_{j-1} \frac{\omega_0 + \omega_1 \eta z_E}{1 - \frac{\omega_1}{\omega_0} \eta z_I} - 2 \frac{b_j}{b_{j-1}} a_{j-1} \frac{\omega_0 + \omega_1 \eta z_E}{1 - \frac{\omega_1}{\omega_0} \eta z_I}$$

$$+ \frac{b_j}{b_{j-2}} a_{j-2} \left( 1 - \frac{\omega_1}{\omega_0} \eta z_I \right) - \frac{b_j}{b_{j-2}} R_{j-2} \left( 1 - \frac{\omega_1}{\omega_0} \eta z_I \right)$$

Recalling the definition of the stability function,



$$w_{nj} = R_j(z)w_n, \quad \tau F_{E,nj} = z_E w_{nj}, \quad \text{and} \quad \tau F_{I,nj} = z_I w_{nj}$$

where

$$F_{E,nj} = F_E(t_n + c_j \tau, w_{nj}) \quad \text{and} \quad F_{I,nj} = F_I(t_n + c_j \tau, w_{nj})$$

the previous function can be multiplied by  $w_n$  to give

$$\begin{aligned} w_{nj} - \tilde{\mu}_1 \eta \tau F_{E,j} &= a_j - \mu_j a_{j-1} - \nu_j a_{j-2} w_{n0} + \mu_j w_{j-1} + \nu_j w_{j-2} + \tilde{\mu}_j \eta \tau F_{E,j-1} + \tilde{\gamma}_j \eta \tau F_{E,0} \\ &\quad - \nu_j \tilde{\mu}_1 \eta \tau F_{I,j-2} - \tilde{\mu}_1 a_j - \nu_j a_{j-2} \eta \tau F_{I,0} \end{aligned}$$

Finally, using the fact that  $a_j - \mu_j a_{j-1} - \nu_j a_{j-2} = 1 - \nu_j - \mu_j$  the final form of the algorithm can be written as

$$\begin{aligned} w_{n0} &= w_n \\ w_{n1} &= w_{n0} + \tilde{\mu}_1 \eta \tau F_{E,0} + \tilde{\mu}_1 \eta \tau F_{I,1} \\ w_{nj} &= (1 - \mu_j - \nu_j) w_{n0} + \mu_j w_{n,j-1} + \nu_j w_{n,j-2} + \tilde{\mu}_1 \eta \tau F_{E,j-1} + \tilde{\gamma}_j \eta \tau F_{E,0} \\ &\quad + \tilde{\gamma}_j - (1 - \mu_j - \nu_j) \tilde{\mu}_1 \eta \tau F_{I,0} - \nu_j \tilde{\mu}_1 \eta \tau F_{I,j-2} + \tilde{\mu}_1 \eta \tau F_{I,j} \\ \tilde{w}_{n+\eta} &= w_{ns} \end{aligned} \tag{7.6}$$

Given the stability function for each stage (7.5) the overall stability function for  $\tilde{w}_{n+\eta}$  is

$$R_s \eta z_E, \eta z_I = 1 - b_s T_s(\omega_0) + b_s T_s \left( \frac{\omega_0 + \omega_1 \eta z_E}{1 - \frac{\omega_1}{\omega_0} \eta z_I} \right).$$

The stability requirement is that  $|R_s \eta z_E, \eta z_I| \leq 1$  for all allowable  $z$ 's. Since the (only) requirement on  $\eta z_E$  is that it is non-positive, it is apparent that

$$\left| \frac{\omega_0 + \omega_1 \eta z_E}{1 - \frac{\omega_1}{\omega_0} \eta z_I} \right| \leq |\omega_0 + \omega_1 \eta z_E|.$$

So as long as  $\eta z_E$  is bound by the two-step Runge-Kutta Chebyshev stability domain defined for  $z$  previously, the stability requirement will be met.

There is still a new issue of consistency to address. The implicit term is based on implicit Euler which is only first-order consistent. To see this effect, consider the approximation of  $e^{\eta z}$ , with  $z = z_E + z_I$ . Recall that  $\tilde{\mu}_1 = b_1 \omega_1 = \omega_1 / \omega_0$ . Rewriting the term of interest as

$$\left( \frac{\omega_0 + \omega_1 \eta z_E}{1 - \frac{\omega_1}{\omega_0} \eta z_I} \right) = \omega_0 + \omega_1 \eta \tilde{z}, \quad \tilde{z} = \frac{z}{1 - \tilde{\mu}_1 \eta z_I}$$

allows for the expansion for small  $z$ ,

$$\begin{aligned} R_s \eta z_E, \eta z_I &= R_s(0) + R_s'(0) \eta \tilde{z} + \frac{1}{2} R_s''(0) (\eta \tilde{z})^2 + \frac{1}{6} R_s'''(0) (\eta \tilde{z})^3 + O((\eta \tilde{z})^4) \\ &= 1 + \eta \tilde{z} + \frac{1}{2} (\eta \tilde{z})^2 + \frac{1}{6} \frac{T_s''(\omega_0) T_s'''(\omega_0)}{T_s'(\omega_0)^2} (\eta \tilde{z})^3 + O((\eta \tilde{z})^4) \\ &= 1 + \eta \tilde{z} + \frac{1}{2} (\eta \tilde{z})^2 + \frac{1}{6} c (\eta \tilde{z})^3 + O((\eta \tilde{z})^4) \quad \text{for large } s \end{aligned}$$

where  $c$  is as defined in Section 7.1.2. This now can be written as

$$\begin{aligned} R_s \eta z_E, \eta z_I &= 1 + \frac{\eta z}{1 - \tilde{\mu}_1 \eta z_I} + \frac{1}{2} \left( \frac{\eta z}{1 - \tilde{\mu}_1 \eta z_I} \right)^2 + \frac{1}{6} c \left( \frac{\eta z}{1 - \tilde{\mu}_1 \eta z_I} \right)^3 + O((\eta z)^4) \\ &= 1 + \eta z + \left( \frac{\tilde{\mu}_1 \eta^2 z_I z}{1 - \tilde{\mu}_1 \eta z_I} \right) + \frac{1}{2} (\eta z)^2 + \frac{1}{2} \eta z \left( \frac{\tilde{\mu}_1 \eta^2 z_I z}{1 - \tilde{\mu}_1 \eta z_I} \right) + \frac{1}{2} \left( \frac{\tilde{\mu}_1 \eta^2 z_I z}{1 - \tilde{\mu}_1 \eta z_I} \right)^2 \\ &\quad + \frac{1}{6} c (\eta z)^3 + \frac{1}{6} c (\eta z)^2 \left( \frac{\tilde{\mu}_1 \eta^2 z_I z}{1 - \tilde{\mu}_1 \eta z_I} \right) + \frac{1}{6} c \eta z \left( \frac{\tilde{\mu}_1 \eta^2 z_I z}{1 - \tilde{\mu}_1 \eta z_I} \right)^2 \\ &\quad + \frac{1}{6} c \left( \frac{\tilde{\mu}_1 \eta^2 z_I z}{1 - \tilde{\mu}_1 \eta z_I} \right)^3 + O(\eta^4 z^4, \eta^4 z^3 z_I, \eta^4 z^2 z_I^2, \eta^4 z z_I^3, \eta^4 z_I^4) \end{aligned} \quad (7.7)$$

Note that all of the denominators will go to one for small  $z$ . Even with that simplification the error compared to the expansion for  $e^{\eta z}$  is still order two. However, all of the parenthetical terms are multiplied by a leading  $\tilde{\mu}_1$  which is proportional to  $1/s^2$ . This means that all of those terms will disappear for reasonably sized  $s$  leaving the error as

$$e^{\eta z} - R_s \eta z_E, \eta z_I = \frac{1}{6} (\eta z)^3 - \frac{1}{6} c (\eta z)^3 + O(\eta^4 z^4, \eta^4 z^3 z_I, \eta^4 z^2 z_I^2, \eta^4 z z_I^3, \eta^4 z_I^4)$$

where the third order terms can be eliminated by the proper selection of  $c$  as discussed previously. So when the full two-step Runge-Kutta method is used the order can be as high as three even after the IMEX extension.

Since the each reaction term has no underlying spatial grid connectivity, each of the stages in (7.6) consists of  $M$  decoupled systems of  $r$  equations where  $M$  is the number of grid points and  $r$  is the number of reactants. Obviously this is a clear advantage over the need to solve the entire system at each step, which is the reason for using the IMEX method.

The overall result is that the IMEX procedure can be integrated into the complete two-step Runge-Kutta Chebyshev method with no change in the stability domain for portion of the system that is handled explicitly. As long as the implicit portion has negative eigenvalues, the method will remain stable.

### 7.1.5 Error Control

Error control is necessary for two broad reasons. Firstly, it is necessary to determine if a time step has caused the solution to become unstable, usually indicated by a large change in the solution vector between two successive time steps. Ideally this is avoided by the use of a method that adjusts the time step based on the time step as the one outlined above does. However, even with this type of error accounted for there is still the possibility of error due to the need for an accurate approximation for the integral. For example, implicit Euler is stable for all time step sizes, but it is unlikely to get an acceptable solution by simply taking one step over the entire interval of integration.

The fundamental concepts of error control were discussed in Section 2.2.5. For the two-step method it suffices to consider the error at each time step so the error analysis can be evaluated identically to the one-step Runge-Kutta Chebyshev method.

The first consideration is the expansion of the Runge-Kutta Chebyshev function for one time step. The most straightforward way to do this is to start from the approximate expansion of  $R_s$  in equation (7.7) which can be written more simply as

$$R_s \eta z_E, \eta z_I = 1 + \eta z + \tilde{\mu}_1 \eta^2 z_I z + \frac{1}{2} (\eta z)^2 + \frac{1}{2} \eta z \tilde{\mu}_1 \eta^2 z_I z + \frac{1}{6} c (\eta z)^3 + O(\eta^4 z^4, \eta^4 z^3 z_I, \eta^4 z^2 z_I^2, \eta^4 z z_I^3, \eta^4 z_I^4)$$

Recall that  $w(t_n)$  means the exact correct solution at  $t = t_n$ . Expansion about a step of size  $\tau_n$  where  $\tau_n \rightarrow 0$  can be reckoned from the above equation by noting a few relationships:

$$w_{n+1} = R_s w_n, \quad z^k w_n = \tau^k w^{(k)}(t_n), \quad \frac{d}{dt} F_I(t_n, w(t_n)) = \frac{\partial F_I}{\partial t} + \frac{\partial F_I}{\partial w} \frac{dw}{dt}$$

Then by multiplying the expansion for  $R_s$  by  $w_n$  becomes

$$w_{n+1} = w(t_n) + \eta \tau_n w'(t_n) + \tilde{\mu}_1(\eta \tau_n)^2 \left( \frac{\partial F_{I,n}}{\partial t} + \frac{\partial F_{I,n}}{\partial w} F_n \right) + \frac{1}{2} (\eta \tau_n)^2 w''(t_n) + \frac{1}{2} \tilde{\mu}_1(\eta \tau_n)^3 \frac{d^2}{dt^2} F_I + \frac{1}{6} c(\eta \tau_n)^3 w'''(t_n) + O(\eta^4 z^4, \eta^4 z^3 z_I, \eta^4 z^2 z_I^2, \eta^4 z z_I^3, \eta^4 z_I^4)$$

To ensure that the error approximation is accurate for both linear and non-linear systems, a more conservative approach is taken wherein error is assumed to occur at the second order and higher terms. Then the error at each step is approximated as

$$est_{n+1} = \frac{1}{2} (\eta \tau_n)^2 w''(t_n) + \tilde{\mu}_1(\eta \tau_n)^2 \left( \frac{\partial F_{I,n}}{\partial t} + \frac{\partial F_{I,n}}{\partial w} F_n \right) + O(\eta^3 \tau_n^3).$$

This estimation for error brings up several points. First of all, the error terms above must be approximated in some form. The simplest way to do so is as follows:

$$w''(t_n) = \frac{d}{dt} F \approx \frac{1}{\tau_n} F(t_{n+1} w_{n+1}) - F(t_n w_n)$$

$$\left( \frac{\partial F_{I,n}}{\partial t} + \frac{\partial F_{I,n}}{\partial w} F_n \right) = \frac{d}{dt} F_I \approx \frac{1}{\tau_n} F_I(t_{n+1} w_{n+1}) - F_I(t_n w_n)$$

The next consideration is the application of a filter for stiff reaction term. A filter is needed due to the potential for very large  $z$  values resulting from the large eigenvalues in stiff problems.

The general design of error estimators is based around the assumption that the  $z$  value goes to zero. When this is not the case (*e.g.* stiff problems) the error will often be overestimated. When such relatively large values are possible, the error estimation should be adjusted so that unnecessarily small time steps are not imposed. [Shampine & Baca, 1994] discuss an approach to remedy this problem. Their idea is fairly straightforward.

Most error estimators compare a test time step to a step based on some other method and use the difference to estimate the error (see Section 2.2.5). In general this gives

$$est_n = w_{n+1}^* - w_{n+1}$$

where  $w_{n+1}^*$  is a step taken with an alternate method. The difference between two methods can be expressed in terms of their stability polynomials:

$$est_n = R_{est}(z) w_n$$

where  $R_{est}(z) = R^*(z) - R(z)$ . This can then be compared to the actual error due to the difference between the actual time step and the real solution (recalling that the approximation should match the exponential up to the order of the method), which leads to the definition

$$R_{err}(z) = e^z - R(z).$$

Clearly  $R_{err}$  and  $R_{est}$  should have the same behavior for the error estimate to be meaningful. This is always achieved for the case  $z \rightarrow 0$  for every error method used. But for the case  $|z| \rightarrow \infty$ , the agreement is often not met. For example, if the solution method is backward Euler and the error estimation is implicit trapezoid, the real and estimated errors are

$$R_{err}(z) = e^z - \frac{1}{1-z}$$

$$R_{est}(z) = \frac{1+z/2}{1-z/2} - \frac{1}{1-z}$$

Both of these go to zero as  $z \rightarrow 0$  as expected. But for the case when  $|z| \rightarrow \infty$ ,  $\text{Re}(z) < 0$ , the equations become

$$R_{err}(z) \sim \frac{1}{z}$$

$$R_{est}(z) \sim -1$$

indicating that the estimated error is much greater than the actual. Indeed any case where the ratio

$$\frac{R_{est}}{R_{err}} \sim kz^m, \quad |z| \rightarrow \infty, \quad \text{Re}(z) < 0$$

has a positive integer  $m$  the error will be over-estimated for large steps. The solution to this problem proposed by the authors is to employ a filter to the error estimation of the form  $(I - \gamma\tau_n A)^{-m}$  where  $\gamma$  is generally chosen to correspond to the solution method so that the same decompositions can be reused for the filter. This filter effectively lowers the order of  $z$  in the above ratio so that the error estimate scales with the real error even for large  $z$ .

In the context of the IMEX RKC method the filter has the form  $(I - \gamma\tau_n F'_{I,n})^{-1}$  where  $F'_{I,n}$  is the Jacobian of the implicit function,  $F'_{I,n} = \frac{\partial}{\partial w} F_I(t_n, w(t_n))$ . The coefficient  $\gamma$  is a free parameter and is used to ensure the error will be of moderate size. It can be chosen by considering the scalar test equation (7.4). The error estimate becomes

$$\begin{aligned}
est_{n+1} &= \frac{1}{2} \tau_n (\lambda_E + \lambda_I) w_{n+1} - (\lambda_E + \lambda_I) w_n + \tilde{\mu}_1 \tau_n \lambda_I w_{n+1} - \lambda_I w_n \\
&= \frac{1}{2} z_E + (1 + 2\tilde{\mu}_1) z_I w_{n+1} - w_n
\end{aligned}$$

and by applying the filter it is

$$est_{n+1} = \frac{1}{2} \frac{z_E + (1 + 2\tilde{\mu}_1) z_I}{1 - \gamma z_I} w_{n+1} - w_n .$$

To decide on a value for  $\gamma$  it is important to consider the behavior of the stability function for large  $z_I$

$$R_s \eta z_E, -\infty = \lim_{z_I \rightarrow -\infty} 1 - b_s T_s(\omega_0) + b_s T_s \left( \frac{\omega_0 + \omega_1 \eta z_E}{z_I} \right) = 1 - b_s T_s(\omega_0) - T_s(0) .$$

All of these terms remain of moderate size, so it is important that the error estimate remains so also. Setting  $\gamma = \tilde{\mu}_1$  is tempting since then the matrix inversion of the filter would be able to use the same LU decomposition as for the system being solved at each stage. However, using this value for  $\gamma$  can be problematic since  $\tilde{\mu}_1$  scales with  $1/s^2$ .

Consider the error estimate with  $z_I \rightarrow -\infty$ :

$$est_{n+1} = \frac{1}{2} \frac{1 + 2\tilde{\mu}_1}{\gamma} b_s T_s(\omega_0) - T_s(0) w_n .$$

If  $\gamma = \tilde{\mu}_1$  then the error will increase with the number of stages which will obviously cause problems since having many stages is often desirable. The value chosen by [Shampine *et al.*, 2005] was  $\gamma = 1$  which leads to a bounded error and thus fulfills the purpose of filtering the stiff components in the error estimate.

The final form of the error estimation is then

$$I - \eta \tau_n F'_I(t_n y_n) est_{n+1} = \frac{1}{2} \eta \tau_n F(t_{n+1} y_{n+1}) - F(t_n y_n) + \tilde{\mu}_1 \eta \tau_n F_I(t_{n+1} y_{n+1}) - F_I(t_n y_n)$$

where this system must be solved for each error estimation. However, as with the formulae for the stages, this system actually needs to be solved only at each grid point, so each system is only this size of the number of species.

Now the actual step size can be selected based on the error estimate. First of all, the absolute and relative error tolerances (*atol* and *rtol*, respectively) must be specified. The absolute error could be the same for all points in the vector or it could change at every

point. The relative error tolerance is a scalar. It should be noted that the error tolerances cannot be very small since this method is at best third order accurate.

As mentioned above the error can be calculated one block at a time. Hereafter the number of species is denoted  $q$  and the number of grid points  $M$ . The error at each grid point  $m$  is based on the by the  $L_2$  norm of the error estimates for all species at that point with each error estimate normalized by the respective  $w$  value. That is to say it is the square root of

$$err_m = \sum_{j=1}^q \left( \frac{est_{n+1,m}^{(j)}}{atol + rtol \max |w_{n,m}^{(j)}|, |w_{n+1,m}^{(j)}|} \right)^2.$$

The total error estimate can then be represented by the norm

$$\|est_{n+1}\|_2 = \sqrt{\frac{1}{q} \sum_{m=1}^M err_m}.$$

The acceptance of the current step is determined by this norm. If it is greater than unity then the step is rejected and recalculated with a smaller time step size  $\tau_n$ . To determine the step size such that the number of rejected steps is minimized the method of [Sommeijer *et al.*, 1997] is employed, with slight modifications. It determines the step size based on the restriction

$$0.1, fac \leq \frac{\tau_{new}}{\tau_n} \leq 6$$

where the term  $fac$  is determined by comparing the previous error norms and time steps as

$$fac = 0.8 \left( \frac{\|est_n\|^{1/2} \tau_n}{\|est_{n+1}\|^{1/2} \tau_{n-1}} \right) \frac{1}{\|est_{n+1}\|^{1/2}}.$$

The upper bound for the ratio of step size increase has been decreased from 10 to 6. This is because the ratio has a large effect on the size of the stability domain for the two-stage method.

The size of the initial step is also of significance. The initial guess is determined by the limitations by the eigenvalues and the stability domain as discussed above. This is actually an important point. Often times the initial step is very difficult to obtain efficiently, but with the stability domain and maximum eigenvalues well established the very first guess for the initial time step is often accepted.

For comparison, a different error estimation method has been implemented based on [Zhang, 2004]. Recalling the expansion for the explicit version

$$\tilde{w}_{n+\eta} = w + \eta\tau_n w' + \frac{1}{2}\eta^2\tau_n^2 w'' + \frac{1}{6}c\eta^3\tau_n^3 + O(\tau_n^4), \quad w = w(t_n)$$

the difference between this and a full Taylor series expansion of  $w$  around  $t_n$  is clearly just dependent on the  $c$  value. This value can be adjusted to allow for agreement up to third order for the linear case but the error estimation in this case will be conservative and allow for  $c$  to be as high as unity. The error estimate is then

$$est_n = \frac{1}{6}c\eta^3\tau_n^3 w'''$$

which can be approximated as

$$est_n = \frac{1}{6}c \left[ 12(w_n - w_{n+1}) + 6\eta\tau_n (F(t_n, w_n) + F(t_{n+1}, w_{n+1})) \right] .$$

This considers the explicit portion. To consider the implicit portion depends on its derivative multiplied by a factor dependent on the number of stages. The total estimated error then becomes

$$est_n = \frac{1}{6}c \left[ 12(w_n - w_{n+1}) + 6\eta\tau_n (F(t_n, w_n) + F(t_{n+1}, w_{n+1})) \right] - \frac{3}{s^2 - 1} F'_I(t_n, w_n) F(t_n, w_n)$$

This error estimation vector is then used similarly to the one outlined above. The only difference is the exponents are now adjusted to become 1/3 instead of 1/2 due the increase in order consistency of one in the explicit case.

This estimation method suffers from the shortcomings mentioned above for methods that do not have any sort of filter to compensate for relatively large time steps in the implicit portion. However, it is still useful as a point of comparison since it avoids the necessity of an extra LU decomposition at every step and so may be worth the price of more conservative time steps.



## 8.0 Actual Implementation and Results

### 8.1.1 Two-Step RKC IMEX Function

The current implementation of the two-step IMEX Runge-Kutta Chebyshev method has been completed in the MATLAB file `rkc2stepIMEX5.m`. The program can solve problems both in purely explicit or IMEX format using either the two-step or one-step methods. There are several inputs to the program and most of them depend on the external function that defines the problem to be solved. They are summarized in the following table:

**Table 8.1: Inputs for `rkc2stepIMEX5.m`**

Input	Description
<code>odefcn_exp</code>	portion of the function to be solved explicitly
<code>odefcn_imp</code>	portion of the function to be solved implicitly
<code>Jacfcn_imp</code>	function of the analytical solution of the Jacobian of the implicit portion
<code>lambdafcn</code>	function returning the largest real and imaginary eigenvalues
<code>tspan</code>	interval in time
<code>w0</code>	initial condition vector
<code>q</code>	number of different species
<code>opt</code>	structure containing several options

There are several options passed to the program given by a list of logicals in the structure `opt`. `timeron` is toggled to report the elapsed time for many of the internal operations in the function. `largescale` causes the LU decomposition to be solved grid point by grid point. This is needed since it turns out that MATLAB's built-in LU decomposition function is more efficient at solving the whole matrix at once despite the block nature of the matrix. But at a certain size the storage requirements become too great so the decomposition must occur on a block by block basis. `onestep` determines if the basic one-step method is used and `progind` displays the progress after each time step. `keepallw` allows for the storage of all intermediate vectors instead of just the initial and final conditions.

The option `errcorr` identifies the error correction method, defined from a list. "0" results in no error correction, *i.e.* relies only on the stability bounds. "1" results in choosing the time steps from a prescribed vector, `tauvect`, irrespective of any other assignments within the program. "2" uses the error method of [Zhang, 2004] that follows a fairly standard Richardson-type approach. Finally, "3" employs the method of [Shampine *et al.*, 2005] that "filters" the error estimate to offset the effect of large  $z$  values in the implicit portion due to stiffness.

The relative and absolute error tolerances are represented by `rtol` and `atol`, respectively and `atol` can be a scalar or a vector the same length as `w0`. The default values of 1/100 for the relative tolerance and 1/1000 for the absolute tolerance are usually a good starting

point. Setting them too much lower is not advisable as the method is at best third-order accurate.

A few of the next variables require some further explanation.  $S\_Im$  is the imaginary stability bound limit as discussed in Section 7.1.3.  $maxr$  is the largest jump in time step size allowed. The variable  $Mg$  is the number of gridpoints being described by the  $w$  vector.  $smax$  is the largest number of stages allowable by the machine precision.  $taumax$  is arbitrarily set to be 1/10 of the time span and  $taumin$  is limited by machine precision. Next the time parameters are initialized and the function evaluation counters are defined.

The first major step is the determination of the initial step size. The overall goal is to select a step size as large as possible without resulting in massive instabilities or errors that will take the error correction many trials to correct. The concept here is to compare the maximum  $z$  value to the stability domain of an explicit Euler step. This conservative approach allows for the fact that the first true step must be a one-step RKC step which has a less favorable stability domain. Additionally this accounts for the potential for a very stiff implicit portion. This is achieved by first determining the maximum real and imaginary eigenvalues via the function  $lambdafcn$  (which is discussed in more detail in the next section). These values along with the  $\infty$ -norm of the Jacobian of the implicit portion are then multiplied by  $taumax$  to determine if any of them will result in any instability. The time step is then reduced to achieve this.

With the initial step defined the main solution loop can begin. First the current time step is truncated if it is large enough to exceed the end of the time interval. Then the largest real and imaginary eigenvalues are determined. If the eigenvalues are purely real then the one-step method is used thereafter since it is more efficient in that case. In that case the necessary number of stages can be determined from a simple polynomial that approximates the relationship between  $z$  and  $s$ .

For the two-step case the value of  $\tau$  is first restricted by the limited  $S\_Im$  defined above. Then the number of stages and the parameter  $\varepsilon$  is determined via a slightly more complicated formula owing to the fact that two parameters are used and the dependence on  $r$ , the ratio of the current to previous time steps. This formulation works as follows. Obviously the minimum number of stages that can still contain  $z_{Re} = \tau \lambda_{Re,max}$  is desired. But there is still one more parameter,  $\varepsilon$ , that can still be used to adjust the shape of the stability domain. With the height of the stability domain set at  $i\sqrt{3}$  there is an optimal extension of the stability domain along the real axis that can be achieved by adjusting  $\varepsilon$ . The results have been calculated by [Sommeijer & Verwer, 2006] and are presented in the following table:

**Table 8.2: Optimal stability boundary for a given number of stages**

$s$	3	4	5	6	7	8	9	10	>10
$\varepsilon$	$\infty$	4	2	1.5	1.5	1.0	0.9	0.9	0.9
$z_{Re}(s)/s^2$	0.14	0.37	0.34	0.43	0.40	0.45	0.44	0.45	0.45

Unfortunately this is only accurate if  $r$  is unity. As long as  $r$  is less than one the bounds are still acceptable but as  $r$  increases the stability domain becomes more restrictive. An amended formula has been incorporated for  $r$  up to 2. Any larger values (up to  $maxr$ ) require a decrease in  $\tau$  and  $s$  being set to 6.

With  $s$  and  $\varepsilon$  determined, the definition of the parameters of the Runge-Kutta Chebyshev method can proceed. Subsequently the parameters corresponding to the two-step portion are calculated as well.

For methods with implicit portions the Jacobian must next be determined. If the Jacobian of the implicit part varies with time it is calculated and expanded to the full block diagonal matrix. The function to be solved involves this term,  $I - \tilde{\mu}_1 \tau Jac$ . Therefore the LU decomposition of this full matrix (recall that it is generally more efficient to solve the whole matrix at once in MATLAB and that this can be controlled with *largescale*) is then completed. The LU composition is done at this point since this is a reasonably costly operation and the values can be used repeatedly at each iteration of the solver discussed below.

Next follows the calculation of the vectors for the internal stages. If there is an implicit portion, each stage requires the vector be obtained by using a simple solver based on a modified Newton's method.

The solver can evaluate the full implicit matrix or solve for each grid point separately based on the value of *largescale*. In the solver the initial guess for the solution vector is based on the value at the previous time step. The solver initially performs an LU decomposition if the values have not been passed to the function. The main equation being solved is

$$res \cdot I - \tilde{\mu}_1 \tau Jac = \tilde{\mu}_1 \tau F_1(t_n, w_{newt}) + rhsconst - w_{newt}$$

where the parenthetical term has already been decomposed into  $L$  and  $U$ , *rhsconst* is the constant terms (that do not depend on the current value of  $w$ ) for the stage and *res* is the residual error which is normalized in the RMS sense and compared to some desired minimum norm. (This is set to be 0.5 as suggested by [Shampine *et al.*, 2005].) Once this norm is achieved the function is considered solved. If the solution cannot be achieved in a set number of iterations (the default is 5) then the solver returns a flag that indicates a failure to the error correction method and additionally returns a string indicating where the failure occurred.

Once all of the internal vectors have been calculated this gives the value for the partial Runge-Kutta Chebyshev step which can be combined with the two-step parameters to achieve a value for  $w$ .

The  $w$  value is then subjected to an error test as outlined above. After the comparison of the old and new vectors a step is either repeated with a smaller  $\tau$  or another step is taken

with a  $\tau$  that may be larger or smaller depending on the size of *fac*. All of the values are recorded and the loop repeats.

The final output is a matrix that contains the solution vector at the initial and final conditions (or at every time step, depending on *keepallw*), a vector containing the value of every time step, and a report that contains various statistics about the solution evaluation.

Throughout the program the number of evaluations of both the explicit and implicit functions is tabulated. Note that the implicit function evaluations are counted based on the entire system (so as to be comparable with the explicit function evaluations) so when the implicit function is evaluated on a per grid point basis it must be multiplied by the number of grid points. In addition the output report contains the number of stages used at each step, the number of Newton iterations needed at each stage, the number and location of rejected steps, and more.

### **8.1.2 Problem Setup Function**

The two-step RKC IMEX solver is designed to be used on a wide range of inputs in the same vein as the built-in MATLAB functions such as *ode45* and *ode15s*. But this solver is clearly designed first and foremost for advection-diffusion-reaction problems. To design a program that can handle a wide a range of these types of problems so they can be solved via the solver described above is a task in itself. It was desired to construct the program to allow for the depiction of systems with multiple species described by multiple reactions, variable velocities, diffusion coefficients, and reaction constants in multiple dimensions. In addition it should allow for Diriclet and Neumann or periodic boundary conditions. To achieve all of this in one program is desirable as it efficiently allows for the evaluation of the RKC solver in a wide variety of practical situations.

The function *advDiffRxnMultiCompMultiDim1.m* takes a large number of inputs described in the table below. It also requires a separate program (*probTemp.m*) that specifies the initial conditions, and any of the coefficients or boundary conditions that are too elaborate to be expressed as simple inputs to the main program.

**Table 8.3: Inputs for advDiffRxnMultiDim1.m**

Input	Description
v	velocity for each dimension (vector)
d	diffusion coefficient for each component (vector)
K	reaction matrix with first $q$ columns for reaction order of each component and the final column for reaction coefficients, e.g. $\begin{bmatrix} 1 & 2 & -k_1 & 0 & 0 & 0 \\ 1 & 2 & k_1 & 0 & .5 & -k_2 \end{bmatrix} \rightarrow \begin{matrix} c_1 = -k_1 \cdot c_1 \cdot c_2^2 \\ c_2 = k_1 \cdot c_1 \cdot c_2^2 - k_2 \cdot c_2^{1/2} \end{matrix}$
h	space step size for each dimension
xspan	intervals in space; matrix with each row corresponding to a dimension
tspan	interval in time (vector)
bctype	boundary condition on each side; matrix with each line corresponding to a dimension ([0 0] = periodic, 1 = Diriclet, 2 = Neumann)
bcs	value of boundary condition on each side; matrix with each line corresponding to a dimension
spdisc	spatial discretization type (0 = 1 <sup>st</sup> order upwind, 1 = 2 <sup>nd</sup> order central, 2 = 2 <sup>nd</sup> order upwind, 3 = 3 <sup>rd</sup> order upwind biased, 4 = 4 <sup>th</sup> order central)
intmeth	time integration method (from list)
splitimex	logical, "true" means the implicit & explicit portions are handled separately
pname	string referring to a specific problem template. This can be a described in the "probTemp.m" function or some other specified function
icond	reference to a specific initial condition (from list, defined in "probTemp")

First the options for the problem setup are defined as well as the various options that are sent to the solver. *filteron* determines if the non-linear filter is employed to maintain positivity. *fulljacalc* when on calculates the full Jacobian of the explicit portion (which is normally unavailable in the filtered version) and *dispmatrix* causes the calculation and display of an analytical version of the final  $A$  matrix. Next the function determines if all of the inputs to the program are consistent.

The number of grid points in each dimension is then calculated based on the spatial step size. The boundary conditions are compared to the assignment from the problem template and then the parameters for the positivity filter are set up. The  $p$ -dimensional initial condition space is then put forth and all of the initial parameters are compared with those from the problem template. Note that the problem template can override nearly all of the inputs defined above. They are mainly available for assignment up front as a convenience for solving simple problems quickly.

Assigning the boundary conditions to the initial conditions array is rather complicated due to the fact that the array can have any number of dimensions. The challenge is handled using strings to build a function that can accept the correct number of inputs for the corresponding number of dimensions. This general approach is employed for all of the cases when the  $p$ -dimensional array is needed. It is not especially efficient from a runtime context but is only used in situations where it is run once, never in an iterative setting.

After the initial condition array is fully defined it is unpacked into a vector that will be fed to the chosen time integration method. The method for this assignment in, for example, the three dimensional case, is

$$w_0 \cdot q \cdot (i-1)M_x M_z + (j-1)M_y M_z + k - q + m = c_0 \quad i, j, k, m \quad (8.1)$$

where  $q$  is the total number of reactants and  $m$  is the specific reactant, and  $M_x$ ,  $M_y$  and  $M_z$  are the number of grid points in each spatial dimension with  $i$ ,  $j$ , and  $k$  being their respective iterators. This is based on equation (6.2).

The function is general enough that it can be solved with several different time integration methods. The `ode45` and `ode15s` are the familiar explicit Runge-Kutta and Backward Differentiation Formula (BDF) functions packaged in MATLAB. `rungeKutter` is a function designed to employ any Runge-Kutta method that can be defined with a Butcher array (see Section 2.2.8). And `rkc2stepIMEX3` is the time integration scheme outlined above.

The actual subfunctions that are fed to the solvers of course are where all the work really happens. `odefcn_imp` is by far the largest. It first checks all the previously defined parameters and defines a few new ones.  $Mg$  is the total number of grid points,  $r$  is the maximum number of reactions for a given component and `advcoefs` defines the coefficients used in each advection scheme. The maximum velocity in each dimension is then calculated as a limiting value.

The first technique defined is the filtered method described in Chapter 6.0. Determining which grid point is adjacent in the vector defined by (8.1) requires several iterators to be defined upfront. These basic iterators are employed throughout the function in many different instances. Also the distinction between positive and negative velocities requires the definition of two different versions of the filter. The  $A$  matrix is then defined based on a basic upwind method multiplied by the  $\gamma$  vector just defined. Note that each velocity is extracted from the problem template if it is variable in time or space. Note also that the  $A$  matrix defined here is not the familiar Jacobian in the basic formulation

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

since it has some  $w$  dependency due to the non-linear  $\gamma$  vector.

It is possible to define the more standard  $A$  as above using the filtered method however. This is accomplished in the next section (for the one-dimensional case only) by taking to account which values are multiplied and divided by which in the algorithm and reassigning them to a matrix. Note that the values for the analytical display version of  $A$  are also assigned in this section.

In both of these cases the extreme rows are assigned via the simple upwind method as is required by the filtered method.

For the unfiltered case, the  $A$  matrix is built using the advection coefficient matrix one row at a time. This takes into account variable velocities and the effect of negative velocity on the upwind direction. Here too the extreme rows are assigned using the upwind scheme if Diriclet boundary conditions are used. In the other boundary condition cases this is unnecessary.

The boundary condition vector is then assigned in a manner similar to the  $A$  matrix. Each case (periodic, Diriclet, Neumann) requires a different assignment method.

If the entire equation is to be handled explicitly (as determined by *splitimex*), the reaction vector is defined within *odefcn\_exp*. The  $A$  matrix is then multiplied by  $w$  and added to the boundary condition vector and the reaction vector.

*odefcn\_imp* is a much smaller function. First it determines if the implicit function even needs to be used. Then it uses either the  $K$  matrix defined as an input or some larger more complicated function described in the problem template to determine the effect on  $w$ . It outputs both the reaction vector and a logical variable indicating if it needs to be used.

*Jacfcn\_imp* determines the analytical Jacobian of the reaction portion based on the  $K$  matrix defined as an input or some larger more complicated function described in the problem template. It also outputs logicals indicating if it is a function of time or space.

The final function, *lambdafcn*, determines the largest real and imaginary eigenvalues of the explicit portion. If the filter is on, safest option is to use Gershgorin's method since there is no regular pattern to the  $A$  matrix. However, using the maximum possible eigenvalues from all of the different schemes is another option here. And of course if the unfiltered method is used then the maximum eigenvalues can be well described with Fourier analysis. These methods are described in Section 7.1.3.

All of these functions are used iteratively by the chosen time integration method. Once the final solution is achieved the final condition array is reconstructed and the final boundary conditions are assigned to the array. Error estimates are available if a known solution is specified in the problem template.

Plotting the output is the final task. For each dimension up to three a different technique is used. For the one-dimensional case each species is assigned a color and the initial condition is shown a solid line while the final answer is shown with a dashed one. If an exact solution is known another plot is output with the initial condition replaced by the exact solution. The two-dimensional case assigns the initial condition in black for each species and gives the final output of each species a different color. Similarly to the one-dimensional case the exact solution plot replaces that solution with the initial condition. The three-dimensional case represents each grid point as a ball in a graph with each axis corresponding to a spatial dimension and the concentration determined by ball color.

## 8.2 Results from New Techniques

This section discusses the results from the positivity preserving spatial discretization discussed in Chapter 6.0 and the Runge Kutta-Chebyshev method discussed in Chapter 7.0.

There are several standard examples that will be used to demonstrate the performance of the program. Some of them have been mentioned in previous sections but they are all restated here. The basic initial conditions are the square and triangular pulses on grids extending from 0 to 1.

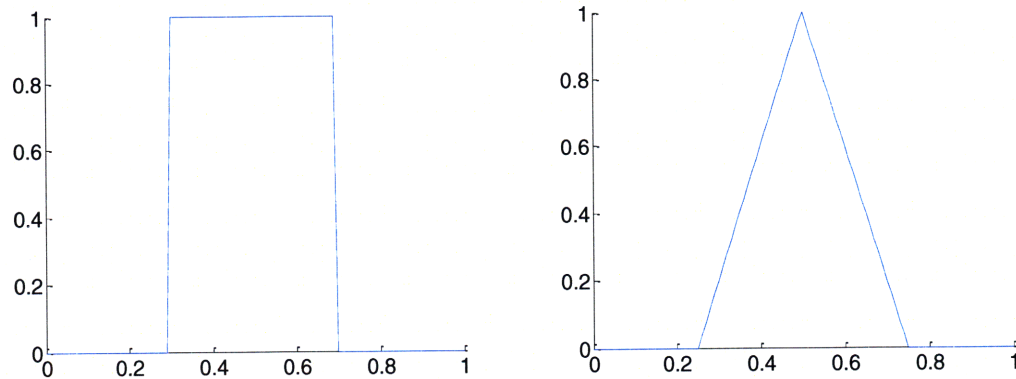


Figure 8.1: Typical Initial Conditions; concentration vs. position

The main examples use periodic boundary conditions for simplicity; other BC types are demonstrated later on. The problems considered include strong diffusion, strong advection, and combined situations. The default spatial discretization method is 3<sup>rd</sup> order upwind biased. Multiple reacting species are combined with the various transport situations as well.

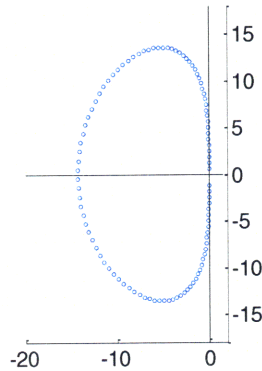
For the purpose of comparison between different methods and conditions, several factors are measured, including the number of time steps, function evaluations (both explicit and implicit, if split), matrix decompositions, and iterations of the modified Newton solvers.

### 8.2.1 Two-Step RKC

The basic Runge-Kutta Chebyshev method was designed for pure diffusion problems where the eigenvalues lie along the real axis. The later modified versions were developed to allow for the inclusion of eigenvectors that lie farther from the imaginary axis. To illustrate the advantage of using a more advanced method, consider the 1-D square pulse described above. Solving with the two-step method is more efficient in cases with advection as expected.

Consider the simple advection problem with  $\nu = 0.10$  and a 3<sup>rd</sup> order upwind biased spatial discretization over a time interval of 10. The eigenvalues are in a ring-like pattern centered on the real axis as shown below.





**Figure 8.2: Eigenvalues for 3rd order upwind biased discretization**

The two-step RKC method evaluates this in 82 time steps and requires 570 function evaluations while the classical RKC method requires 200 time steps and 1139 function evaluations. This is of course due to the smaller stability domain of the classical method in the imaginary direction.

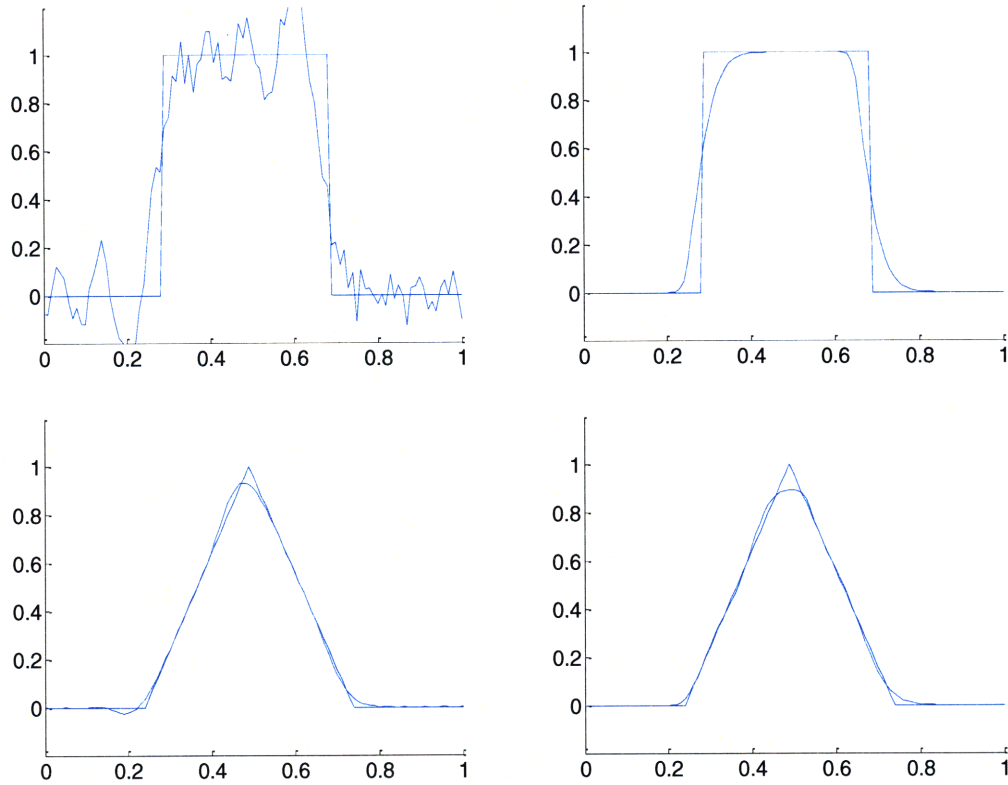
However, the classical RKC still has the advantage for pure diffusion cases. For the same problem as above but with zero advection and a diffusion coefficient of  $1E-4$  the classical RKC requires 15 time steps and 136 function evaluations. The two-step RKC method is slightly less efficient requiring 16 time steps and 192 function evaluations. This is as expected since the two-step method is broader in the imaginary direction and requires more stages to achieve stability along the real axis than does the classical RKC.

The greatest advantage of the two-step RKC method is for situations with eigenvalues along the imaginary axis. The classical RKC method has effectively no stability domain near the imaginary axis while the two-step RKC method's stability domain runs along the axis. For the same advection problem as above but with the second order central spatial discretization the two-step RKC solves the problem in 59 time steps and 297 function evaluations. The classical RKC requires 207 time steps and 1664 function evaluations and still does not achieve as accurate a solution as the two-step method. Forcing it to use 10,000 time steps still cannot result in the same accuracy. This makes sense as there will always be some instability due to the fact that the only stability the classical RKC method has along the imaginary axis is at the origin. The extraneous oscillations regardless of decreasing time step size reflect this.

### 8.2.2 Positivity Preservation

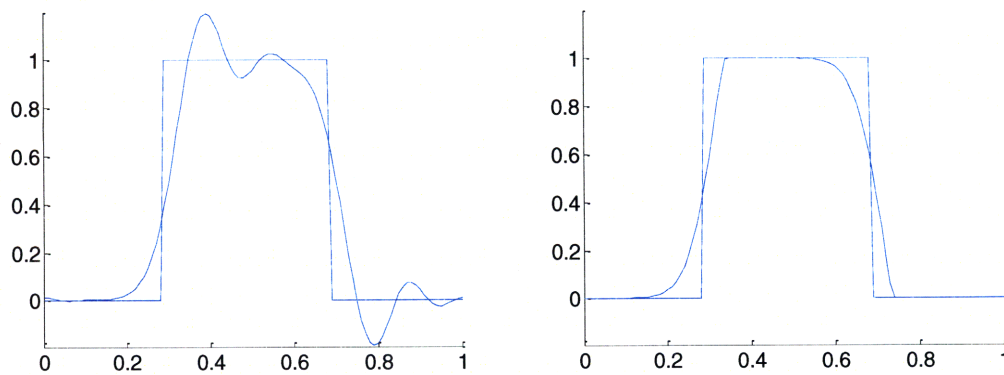
A positivity preserving filter is often necessary in spatial discretizations of advection problems. The filter used in the programs discussed is described in Section 6.1. Due to the nature of the eigenvalues of the system after the filter is applied, there is generally a tighter restriction on the time step (see below), but the need for a much greater number of spatial grid points is avoided by the use of the filter.

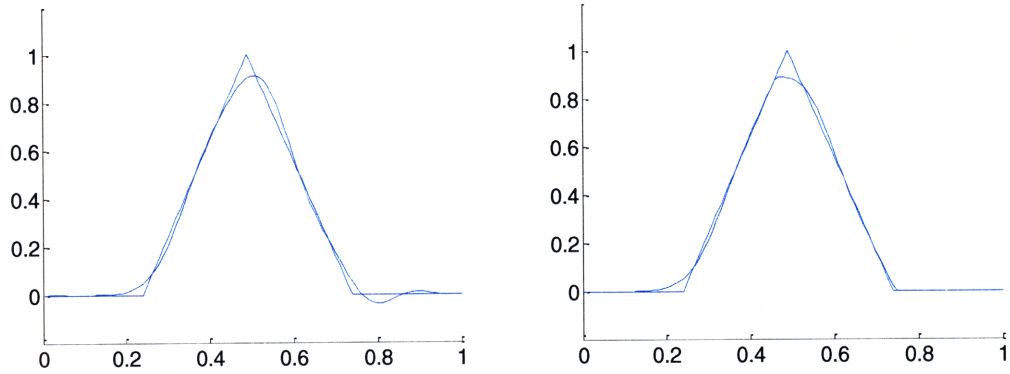
First consider the simple 1-D advection problem with  $\nu = 0.10$  and 100 grid points over a time interval of 10. The triangular and square pulses each gain accuracy differently from the filter. All spatial discretizations of order two or greater can exhibit some unphysical oscillations and can benefit from the positivity filter. Four examples are shown below with and without the filter:



**Figure 8.3: Advection with 2<sup>nd</sup> order central discretization**

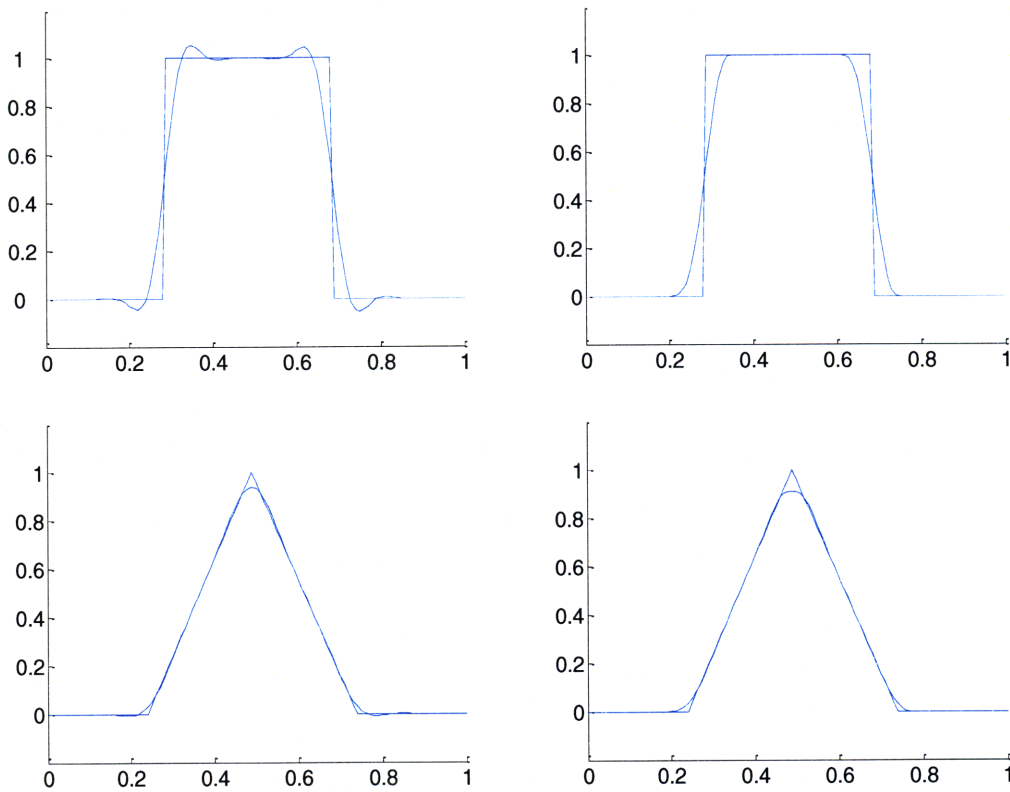
Unfiltered on left, filtered on right; exact solution (solid line) and approximation (dashed line).





**Figure 8.4: Advection with 2<sup>nd</sup> order upwind discretization**

Unfiltered on left, filtered on right; exact solution (solid line) and approximation (dashed line).



**Figure 8.5: Advection with 3<sup>rd</sup> order upwind biased discretization**

Unfiltered on left, filtered on right; exact solution (solid line) and approximation (dashed line).

Recall the definition of errors put forth in Section 2.2.5. The absolute and relative errors are defined with vector norms for  $m$ -length  $w$  vectors as

$$\text{absolute error} \equiv \frac{1}{m} \|w - w_{\text{exact}}\|_1$$

$$\text{relative error} \equiv \frac{1}{m} \frac{\|w - w_{\text{exact}}\|_2}{\|w_{\text{exact}}\|_2}$$

and the peak error is defined as the difference between the highest point of the exact solution and the corresponding approximate point, divided by the exact value. They are presented in the following table along with the number of time steps and function evaluations required.

**Table 8.4: Advection Test Error Results**

	Filter Used	Absolute Error	Relative Error	Peak Error	Time Steps	Function Evals
Square Pulse						
2nd order central	No	0.1232	0.0027	0.3023	140	1270
2nd order central	Yes	0.052	0.0020	0	132	1324
2nd order upwind	No	0.09079	0.0026	0.6021	133	1332
2nd order upwind	Yes	0.0602	0.0021	0	152	1744
3rd order upwind biased	No	0.0443	0.0017	0	92	924
3rd order upwind biased	Yes	0.0421	0.0018	0	135	1376
Triangular Pulse						
2nd order central	No	0.0094	0.000403	0.0708	61	530
2nd order central	Yes	0.0098	0.000468	0.1069	128	1282
2nd order upwind	No	0.0188	0.000678	0.0986	129	1286
2nd order upwind	Yes	0.0115	0.000525	0.1114	128	1282
3rd order upwind biased	No	0.0048	0.000234	0.0592	82	816
3rd order upwind biased	Yes	0.0052	0.000324	0.0884	128	1282

First off it is readily apparent that the square pulse benefits much more overall from the positivity filter. In all cases the purported goal of positivity is achieved but the sharp changes in the square pulse more greatly benefit from the reduced oscillations. The triangular pulse solution only benefited inasmuch as the negative portions were eliminated. The errors were actually somewhat higher in some of the filtered cases. As

expected the third order upwind biased discretization was the most accurate and saw the smallest improvement after filtration. However, it did too contain negative portions that were successfully eliminated.

The square pulse saw a reasonably small increase in the number of time steps and (more importantly) function evaluations for the filtered case. In most cases the triangular pulse versions required a larger increase in time steps and function evaluations. The benefit in these cases was then very small when considering that the only change is a reduction of small areas of negativity.

The difference in the number of time steps and function evaluations need is due partially to the way that the bounds on the eigenvalues are characterized. As mentioned in Section 7.1.3 there are two principle ways that the eigenvalues are characterized in the program. Gershgorin's method bounds the eigenvalues for any matrix, but the bounds are not very tight. Using Fourier analysis on the matrices that have the regular patterns resulting from the standard spatial discretizations can put a more exact bound on the eigenvalues. Recall that these bounds are actually applicable to the filtered versions as well.

This difference in boundaries on the real and imaginary eigenvalues generally causes an increase in the necessary number of time steps or the number stages (and therefore function evaluations) in cases where stability rather than accuracy restrict the time step size. In the above examples the error correction was employed to restrict the step size. So to demonstrate the effect of different eigenvalue bounding methods consider the same problem solved with Fourier or Gershgorin methods to characterize the eigenvalues and the error correction disengaged. Using the same sample problem as above, with a 2<sup>nd</sup> order central spatial discretization, the results are that the Fourier analysis version took 59 time steps and 297 function evaluations and the Gershgorin method version needed 59 time steps and 469 function evaluations. And for a 3<sup>rd</sup> order spatial discretization, the Fourier analysis version required 82 time steps and 570 function evaluations and the Gershgorin method version took 88 time steps and 705 function evaluations. Overall the tighter bounds are the main reason for the advantage of the non-filtered method in most cases.

The results in this section demonstrate that it is important to consider both the desired application for the output and the final form of the solution. If positivity is an essential feature of the solution then the filter should be employed as necessary. If, however, some non-positivity can be tolerated, the solution can be achieved with similar accuracy and more efficiently in some cases without the filter engaged. Finally, it is important to recall that the non-positivity only results from imaginary eigenvalues and therefore is only relevant in advection-dominated problems.

### **8.2.3 RKC IMEX**

The use of the implicit-explicit method offers great advantages on problems exhibiting stiffness due to the reaction terms. The best way to observe this is with a simple example. Consider a reaction-diffusion problem with two components. Assume both

have a diffusion coefficient of 1E-4 and take a time interval of 10 and a spatial grid of 100 points, as above. The 1<sup>st</sup> order reaction has the form

$$w_1 = -k_1 w_1(t) + k_2 w_2(t)$$

$$w_2 = k_1 w_1(t) - k_2 w_2(t)$$

where the  $k$ 's and the reaction constants and the subscripts on the  $w$ 's refer to each component. This reaction occurs at each point on the grid.

The overall stiffness of this problem can depend on either the diffusion discretization or the values of the reaction constants. Assuming the diffusion is set as stated above, the eigenvalues will be distributed along the negative real axis with a maximum value of -4. Since the eigenvalues are also present back near the origin the stability region must encompass all of these values.

For the linear system of equations above it is easy to characterize the eigenvalues in a similar manner. Considering the two by two matrix that represents the two reactions occurring at each grid point it is clear that the eigenvalues are 0 and  $-k_1 - k_2$ . If the reaction constants remain small then the eigenvalues from the diffusion discretization dominate. But as the reaction terms become large, they will result in larger and larger eigenvalues and make the overall problem increasingly stiff.

When the IMEX method is employed the reaction terms are solved with an implicit method (possessing an infinite stability domain) so that only the spatial discretizations affect the explicit stability domain (which is defined by the two-step RKC method).

The example described above is solved under several conditions both with and without using IMEX.  $k_1$  is set at 1 and  $k_2$  is changed to various values listed below.

**Table 8.5: Standard RKC vs. IMEX RKC; reaction-diffusion problem**

$k_2$	Standard			IMEX					
	Time Steps	Explicit Function Evals	Failed Steps	Time Steps	Explicit Function Evals	Failed Steps	Implicit Function Evals	LU Decomps	Newton solver iterations
1	22	174	2	22	174	2	346	21	137
10	45	447	18	33	260	3	517	33	205
100	294	3284	170	46	342	2	682	45	281
1000	2643	29925	1632	50	391	5	779	52	304

As these results show, there is no advantage in the IMEX when the reaction coefficients are small. Since there are the same number of time steps for both methods it is clearly the eigenvalues that are due to the diffusion discretization that are determining the required stability domain. In this case the extra computation necessary to perform LU decompositions and solve the system of equations make the IMEX method unattractive.

But as the reaction constants become increasingly large the advantage of the IMEX method becomes apparent. For a modest increase in the number of time steps and function evaluations the solution can be obtained for reaction constants that are orders of magnitude larger. The standard RKC method requires a number function evaluations that grows with the magnitude of the reaction constants. The extra computation needed to evaluate the reaction portion implicitly is easily justified in these cases.

Looking at the values of the time steps as the solution progresses, both the standard and IMEX methods need increasingly small time steps at the beginning as the reaction coefficients increase. This is due to the fact that the solution changes rapidly at early times for fast reactions and accuracy requirements restrict the time step size. But as the solution levels off to a more steady state, the IMEX method is able to adjust the time step down to the stability limits imposed by the spatial discretization while the standard method still needs very small time steps to contain the large eigenvalues of the reaction terms.

As expected using the IMEX method shows great advantages for stiff reactive systems. While only linear reaction portions were discussed here the same results apply; it is just the eigenvalues of the Jacobian of the system of equations describing the reactions that must be considered.

#### **8.2.4 Error Correction**

In general, the use of error correction in the time integration is needed to ensure stability and achieve a desired level of accuracy. The RKC methods as they are implemented here take stability bounds as an input so accuracy is the only concern of the error correction. Indeed it is possible to turn off the error correction and still obtain reasonable results since there are none of the large deviations associated with instability.

Built into the RKC program are four different error correction options. The first option disengages the error control and the second one allows the input of a set series of time steps. The other two options are the true error control techniques discussed in Section 7.1.5. The Sommeijer method works well for explicit problems where the eigenvalues are not too large and spread out. But when the step sizes can become large Shampine's method becomes more useful.

Of importance to the error correction techniques is the value for the error tolerance. The absolute tolerance is a scalar while the absolute tolerance can be a scalar or vector if there are expected to be significantly greater errors present in one region relative to the rest of the solution grid. Since the RKC methods are either second or third order, the tolerances cannot be too low. In general, the relative error tolerance is set to 1E-2 and the absolute to 1E-3 unless otherwise noted.

To demonstrate the effectiveness of each error correction technique several examples are evaluated and compared to an exact solution. When the exact solution is not available a solution is calculated with very small time steps and tighter error tolerance to use for

comparison. The problem was similar to the one used throughout with a square pulse and a time interval of 10. The first reaction constant is 1 where necessary and 3<sup>rd</sup> order upwind biased discretization with the positivity filter was used in the pure advection case. All other parameters are as noted below.

**Table 8.6: Error Correction Results**

Error Cor	$d$	$\nu$	$k_2$	Time Steps	Explicit Function Evals	Failed Steps	Implicit Function Evals	LU Decomps	Newton solver iterations
0	1E-4	0		11	55	0			
2	1E-4	0		15	136	1			
3	1E-4	0		19	153	2		21	
0	0	0.1		128	898	0			
2	0	0.1		135	1376	3			
3	0	0.1		257	2296	1		258	
0	1E-4	0	1	11	55	0	110	3	66
2	1E-4	0	1	50	426	2	694	51	309
3	1E-4	0	1	22	174	2	346	45	137
0	1E-4	0	1000	11	55	0	110	3	66
2	1E-4	0	1000	$\infty$	$\infty$		$\infty$		
3	1E-4	0	1000	50	391	5	779	107	304

Error Cor	$d$	$\nu$	$k_2$	Abs Error	Rel Error
0	1E-4	0		2.80E-03	1.29E-04
2	1E-4	0		6.28E-05	2.07E-06
3	1E-4	0		5.71E-05	1.62E-06
0	0	0.1		9.73E-04	3.39E-05
2	0	0.1		9.10E-04	3.22E-05
3	0	0.1		2.09E-04	7.54E-06
0	1E-4	0	1	2.10E-03	6.42E-05
2	1E-4	0	1	3.53E-05	6.73E-07
3	1E-4	0	1	3.86E-05	7.30E-07
0	1E-4	0	1000		
2	1E-4	0	1000		
3	1E-4	0	1000	3.98E-05	7.51E-07

Several interesting items can be noted. As expected, disengaging the error control leads to the worst results. However, in the pure advection case there was only a relatively small advantage. Moreover, the results are never completely incorrect (except in the second reaction case). To the eye there is almost no distinction between the correct solution and the sans error control solution. This demonstrates that maintaining stability is generally the most important function of error correcting algorithms and if stability is ensured, error correction is mainly needed for fine tuning. The main difficulty lies in the fact that it can occasionally fail on very fast reactions that effectively occur between the relatively large time steps.



The method of Shampine shows a slight advantage in accuracy in most of the cases, but that is largely due to its slightly more restrictive error control. The accuracy scales roughly with the number of time steps in the non-reactive cases. The disadvantage of Shampine's method is that it requires LU decompositions at every time step (in addition to the LU decompositions in the main body of the function).

Where the advantage of Shampine's method is greatest is in the reactive cases. Even for the non-stiff reaction case this method requires fewer time steps for nearly the same level of accuracy. And for the stiff case the Sommeijer method required a very large number of time steps while the Shampine method needs only to double the number of time steps it used in the non-stiff case. This easily justifies the extra computation need for the LU decompositions.

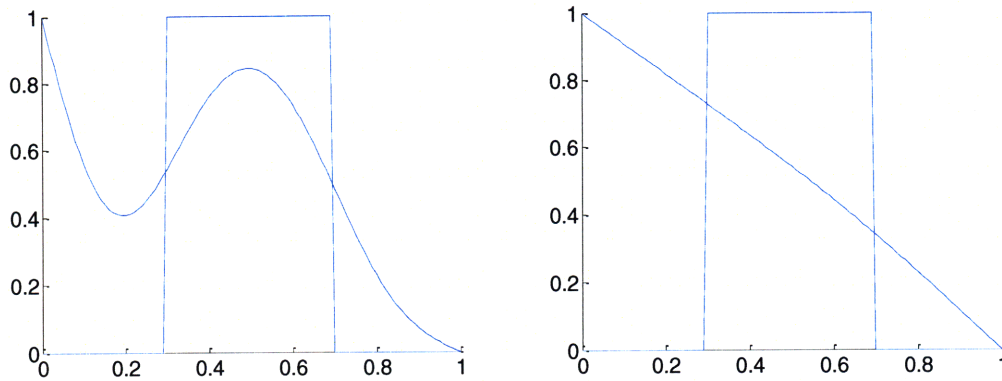
Clearly different situations call for different error control schemes. The no error correction scheme always bears some consideration as it can produce passable results in most cases in a very short amount of time steps. Sommeijer's method is the winner for non-reactive cases. For both stiff and non-stiff reactions Shampine's method has the clear advantage. Of course there may be cases in which these generalities are not correct so it is worthwhile considering each of the error correction methods.

### **8.2.5 Other Problem Types**

Thus far only one-dimensional cases with periodic boundary conditions have been considered. However, the input function can handle many different types of problems. Several of them are demonstrated here.

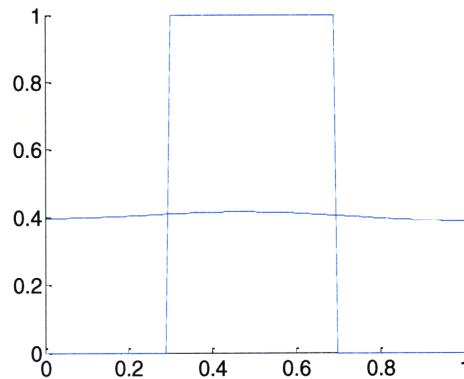
Boundary conditions are fairly simple to consider mathematically but quite a challenge to implement in a program in a general context, especially in multi-dimensional cases. A few examples demonstrate the results from non-periodic boundary conditions.

Both Diriclet and Neumann boundary conditions can be input with the function statement and more exotic space- or time-dependent boundary conditions can be defined in the problem template function. First consider a constant input source of the left-hand side of the system with a sink on the right hand side. If the square pulse is again used as the initial condition with  $d=1E-4$ , the results show what happens as time progresses:



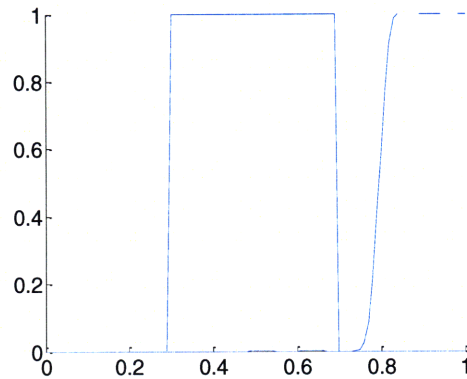
**Figure 8.6: Dirichlet Boundary Conditions,  $t=100$  (left) &  $t=1000$  (right)**

The solution is the dashed line. Now consider the equilibration of concentration in a vessel with no output (zero flux at the sides) and the same initial conditions and diffusion as above. Over time the concentration becomes uniform as expected. Note also that mass is conserved after the time integration.



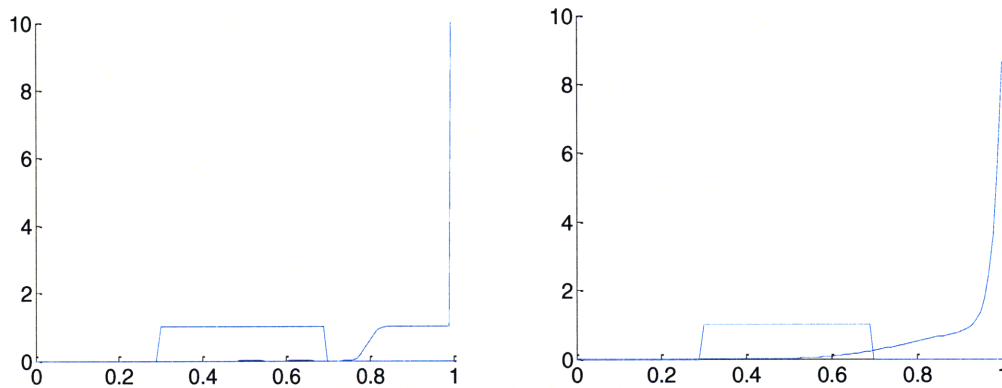
**Figure 8.7: Neumann Boundary Conditions,  $t=1000$**

Advection problems can be solved with boundary conditions as well. However, they do suffer the inherent limitations of the underlying mathematical model inasmuch as there can only be one boundary condition since the equation is first order. Also, Neumann boundary conditions cannot be applied for the same reason. This is demonstrated by the fact that when no-flux conditions are applied, the solution travels through as if the condition were not present and mass leaves the system. Consider an advection problem with  $v=0.05$  and a time interval of 10 solve with positivity filtered 3<sup>rd</sup> order upwind discretization over 100 grid points.



**Figure 8.8: Advection with no flux BC; Initial condition (solid line) and solution**

This is the correct behavior based on the underlying model but it is not physically correct. A simple change in the boundary condition portion of the problem template function corrects this to get a somewhat odd-looking but more physically correct result. In the absence of diffusion all of the mass will simply accumulate on the right hand side due to the constant advection in that direction.



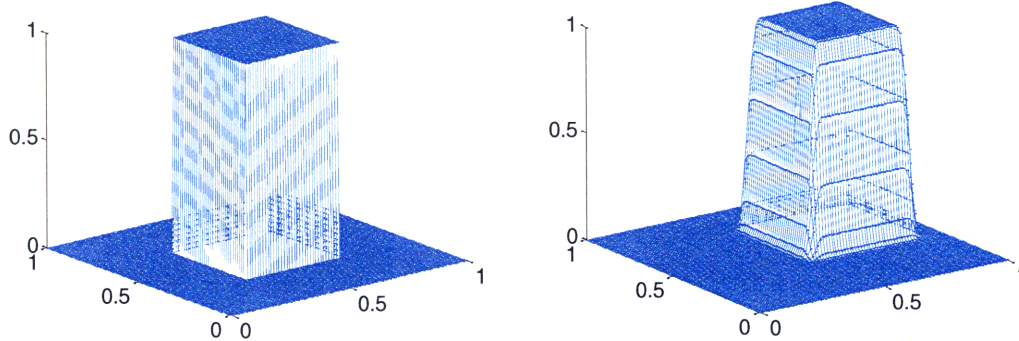
**Figure 8.9: Advection with no flux BC, corrected solution and solution with diffusion**

And adding diffusion gives the somewhat more reasonable solution on the right. This is just the surface of what can be done with boundary conditions.

One of the other major features is the ability to consider multi-dimensional problems with the same framework as the one-dimensional case. Any of the problems outlined above can be adapted to a two- or three-dimensional analog with relative ease. And higher dimensions can be handled as well if there is some desired quantity that can be represented as another dimension.

For demonstrative purposes a few simple two- and three-dimensional problems are solved. The output of the 2-D case is easy to interpret. The x-y plane represents space and the z-axis denotes concentration. For an example problem, consider a 100 by 100 point grid and a square pulse as the initial condition. Now consider advection in one

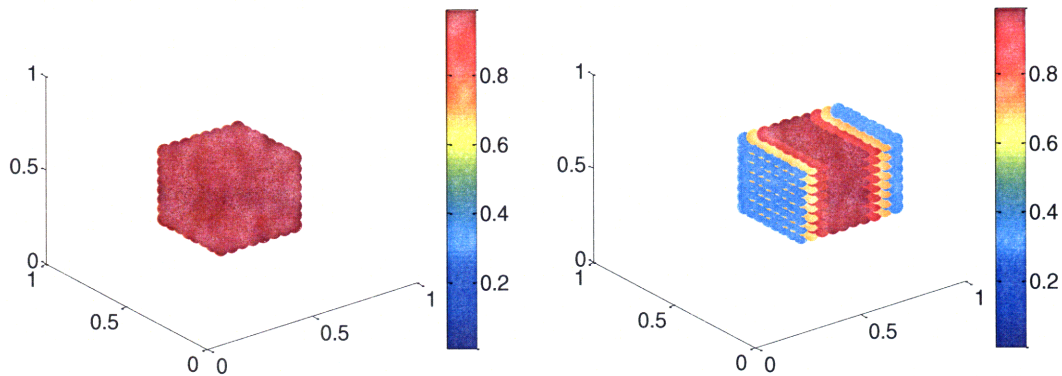
direction with a velocity of 0.2 and 0.1 in the  $x$  and  $y$  directions, respectively over a time period of 10. The third-order upwind biased discretization with positivity filter is used.



**Figure 8.10: 2-D Advection**

The solution behaves as expected and remains positive. There is only a slight spreading due to artificial diffusion. It requires 10 time steps and 50 function evaluations to achieve this.

Now consider a 3-D problem with a 20 by 20 by 20 grid. Consider a velocity of 0.2 in the  $x$  direction with a time interval of 10. The discretization is the same as above as well. The results in the 3-D case require a bit more explanation. All three of the axes now represent a spatial dimension. The color of the ball at each grid point indicates the concentration (indexed by the scale on the side) unless there is zero concentration at that point, in which case it is blank.



**Figure 8.11: 3-D Advection**

Once again the results are as expected. Positivity is maintained but there is a slight artificial diffusion in the  $x$  direction. The solution took 11 time steps and 73 function evaluations. It is possible to add reactions, different boundary conditions, etc. but the results have no surprises.

While the number of time steps and function evaluations are similar to the one-dimensional counterparts the time to evaluate each function is much greater. This typifies the greatest challenge of multi-dimensional problems. Such problems are fundamentally more complex and the solution time scales with the number of points in the grid. This inherent difficulty can only be avoided by employing a completely different solution method, such as Monte Carlo simulation.

### 8.2.6 Comparison with Other Methods

There are a plethora of different methods for solving systems of differential equations. It is therefore worthwhile to consider the performance of the RKC methods alongside some different techniques. The easiest two to start with are the built-in MATLAB functions, ode45 and ode15s. ode45 is a 5<sup>th</sup> order four-stage Runge-Kutta method and ode15s is a modified 5<sup>th</sup> order BDF method.

A few caveats are necessary before any comparisons begin. Both of the MATLAB functions mentioned are fully developed pieces of software. In addition, they are optimized to take advantage of the manner in which MATLAB runs. Also, they are both up to 5<sup>th</sup> order methods whereas the RKC methods are only 2<sup>nd</sup> or 3<sup>rd</sup> order.

With these issues it makes the most sense to compare methods based on the number of time steps, function evaluations, Newton solver iterations, etc. These performance indicators give a general idea how efficient a method is when used on a given problem.

The problem input function is quite flexible. It allows for a given problem to be solved with any time integration method desired. This allows for all of the discretizations, positivity preservation, reaction term splitting (if necessary), and other features to be uniform across different solution methods.

First off, consider pure diffusion problems. For the example problem define a grid of 100 points and a square pulse as the initial condition. Solving over a time interval of 10 gives the following results for varying diffusion coefficients:

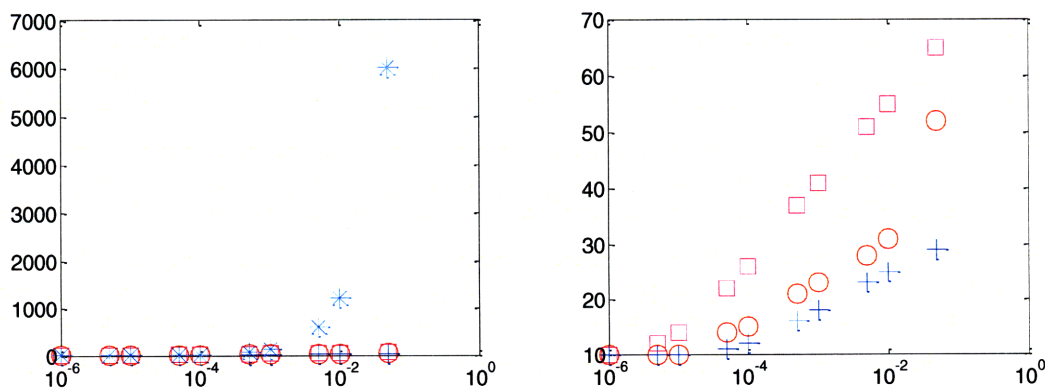


Figure 8.12: Time Steps vs. Diffusion Coef

Blue + = RKC (no error correction), Red circle = RKC, Cyan star = ode45, Magenta square = ode15s.

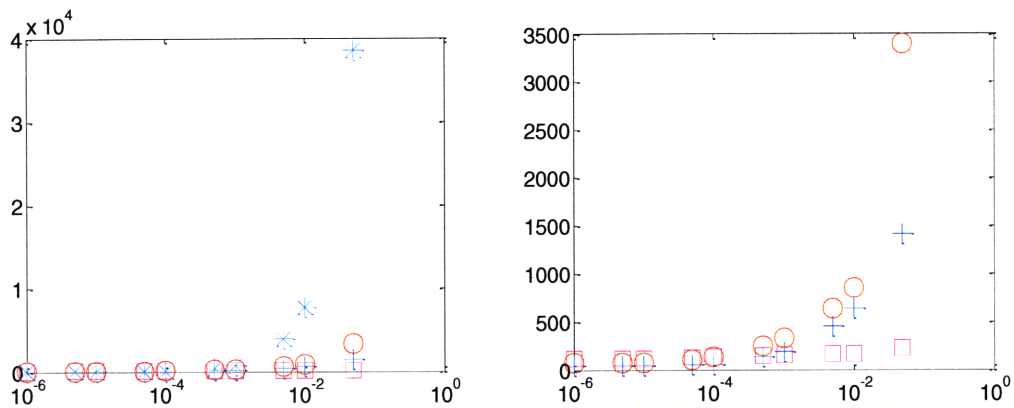


Figure 8.13: Function Evaluations vs. Diffusion Coef

Blue + = RKC no err, , Red circle = RKC err corr, Cyan star = ode45, , Magenta square = ode15s.

At low diffusion there is no clear advantage for any method. As the diffusion becomes stronger, the advantage of the RKC method over the ode45 method becomes apparent. Eventually the stiffness caused by very large diffusion coefficients makes the ode15s method more attractive. The relatively low number of function evaluations overcomes the cost of matrix operations in this implicit method. Even the non-error corrected RKC method (which remains quite accurate) eventually becomes less attractive than ode15s.

For a pure advection test consider the same system as above with 3<sup>rd</sup> order upwind biased spatial discretization with positivity filtering. Varying the velocity gives the following results:

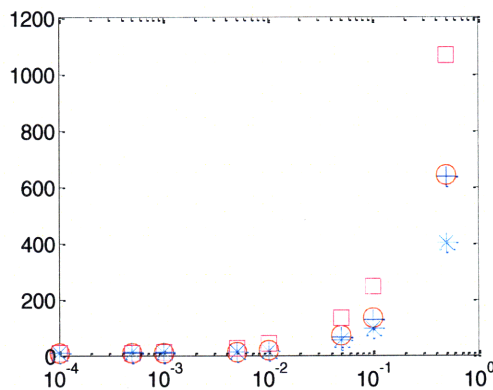


Figure 8.14: Time Steps vs. Velocity

Blue + = RKC (no error correction), Red circle = RKC, Cyan star = ode45, Magenta square = ode15s.

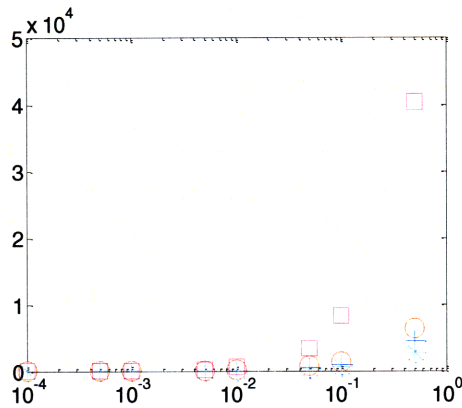


Figure 8.15: Function Evaluations vs. Velocity

Blue + = RKC (no error correction), Red circle = RKC, Cyan star = ode45, Magenta square = ode15s.

ode45 maintains a general advantage over the RKC method for most of the range. The location of eigenvalues spread in the imaginary space near the imaginary axis is somewhat better suited to the stability domain of standard Runge-Kutta methods. ode15s is never a viable option for these cases as it requires a large number of function evaluations in addition to its inherent required matrix operations. It seems that its stability domain is not very large near the imaginary axis.

To test the IMEX performance of RKC reaction-diffusion and reaction-advection problems were solved using the conditions above. For the diffusion problems a diffusion coefficient of 1E-4 was used and a system of two reactions,

$$w_1 = -k_1 w_1(t) + k_2 w_2(t)$$

$$w_2 = k_1 w_1(t) - k_2 w_2(t)$$

was used with  $k_1$  set to unity and  $k_2$  varied from 1 to 1000.

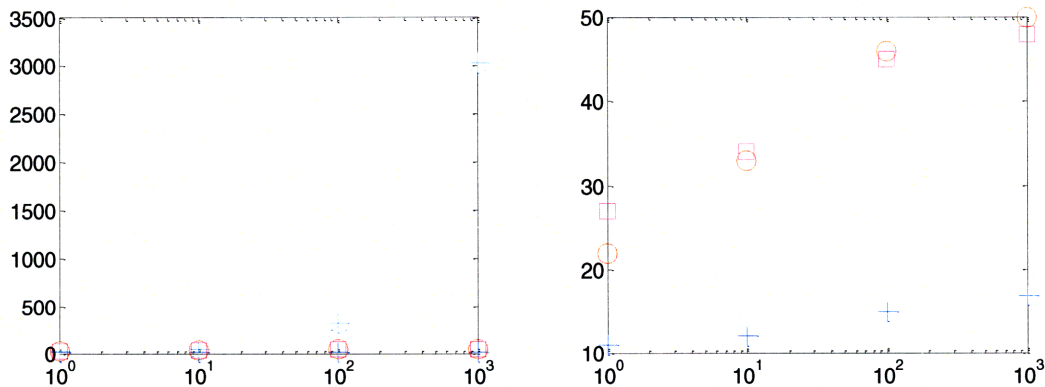


Figure 8.16: Time Steps vs.  $k_2$

Blue + = RKC (no error correction), Red circle = RKC, Cyan star = ode45, Magenta square = ode15s.

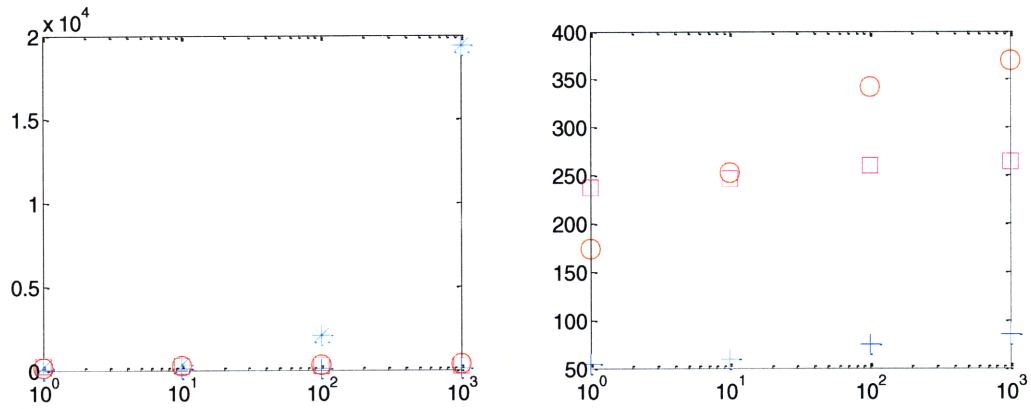


Figure 8.17: Function Evaluations vs.  $k_2$

Blue + = RKC (no error correction), Red circle = RKC, Cyan star = ode45, Magenta square = ode15s.

As expected, ode45 performs well until the stiffness becomes large forcing a great increase in function evaluations needed. The RKC method performs similarly to ode15s overall. It eventually requires more function evaluations but only the uncoupled grid points need to be solved implicitly rather than the entire Jacobian, resulting in simpler matrix operations necessary for the IMEX case. The no-error-correction RKC method requires a relatively small number of function evaluations and still remains quite accurate in this case.

For the reaction-advection problem the velocity is set to 0.1 and the same reaction system and values as above are used.

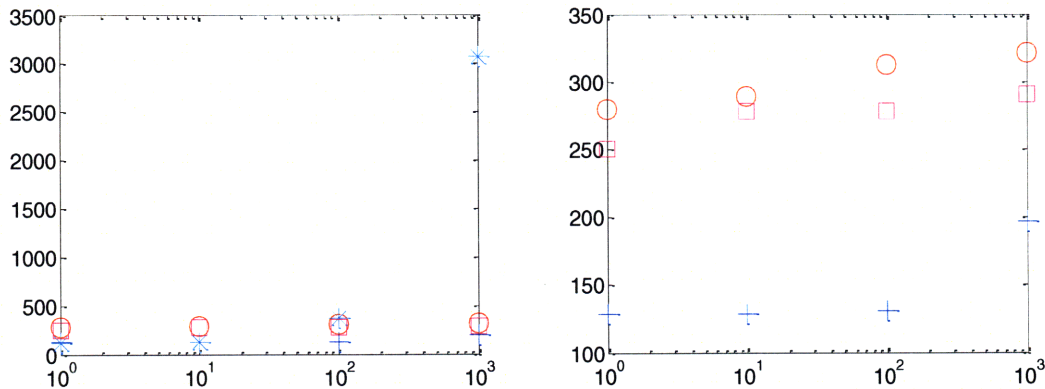


Figure 8.18: Time Steps vs.  $k_2$

Blue + = RKC (no error correction), Red circle = RKC, Cyan star = ode45, Magenta square = ode15s.



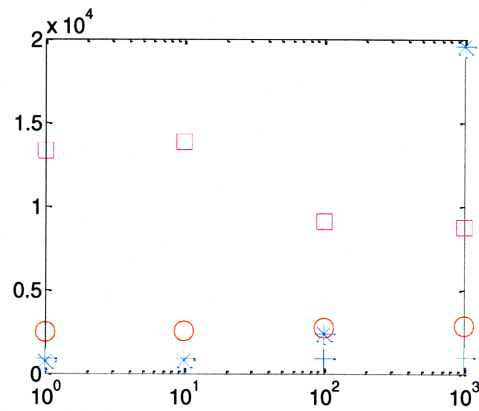


Figure 8.19: Function Evaluations vs.  $k_2$

Blue + = RKC (no error correction), Red circle = RKC, Cyan star = ode45, Magenta square = ode15s.

Similarly to before, ode45 struggles as the stiffness in the reaction portion increases. The advantage of the RKC methods becomes apparent as it requires very little increase in function evaluations to handle the increased stiffness. ode15s shows the previously observed difficulty with the strong advection portion and actually decreases the number of function evaluations as the reaction portion becomes stiffer. This is likely due to the eigenvalues moving away from the imaginary axis for higher reaction constants.

Overall, different problems and situations can be handled most efficiently by different methods; it impossible to determine which one is unilaterally the “best”. Indeed, that is why MATLAB includes several different functions.

The principle advantage of the RKC methods is that they can take as an input the problem domain information. This allows the calculation of the largest time step possible for stability. This advantage is most greatly evident in the no-error-correction examples which have the largest possible time steps allowed by stability. Several factors can mitigate this advantage, however. Using this time step often gives in acceptable results but if high accuracy is needed the error control will restrict the time step and decrease this advantage.

RKC methods show the greatest advantage versus the explicit, Runge-Kutta ode45 method in pure diffusion problems which is the purpose for which they were originally constructed. The performance is less competitive when it comes to advection dominated problems. This is because the eigenvalues are spread far from the real axis which decreases the advantage of RKC relative to standard RK methods that already show stability in that region – knowing the location of the eigenvalues is irrelevant if they are already in the stability domain of another method. When compared to the implicit, BDF ode15s method there is an advantage until the number of function evaluations become so high as to offset the cost of performing matrix operations and Newton’s method solver iterations.

Another advantage of the RKC method is the IMEX splitting. The advantage over explicit methods was discussed in Section 8.2.3 but there is also an advantage over purely implicit methods. By only considering the implicit portion of the problem, the system is simpler to solve. Additionally, each grid point can be solved for individually which avoids the need to decompose the entire matrix at once. Unfortunately, several issues decrease the advantage of the factors in MATLAB. It turns out that, counterintuitively, MATLAB is significantly more efficient at decomposing and evaluating an entire matrix at once rather than in small subsections. This is despite the coupling between points that is present in the advection-diffusion portion. In addition, MATLAB is far more efficient at performing linear algebra operations than at evaluating functions. Altogether this means that the standard cost associated with IMEX methods of more function evaluations is not offset by the simpler matrix to be evaluated when implemented in MATLAB.

While the RKC methods do not have a clear advantage in all of the tests, they do show acceptable performance across many different conditions. This is especially advantageous for advection-diffusion-reaction problems with time- or space-dependent coefficients. Even as the eigenvalues change as the solution progresses, RKC will be able to determine the optimal stability domain at each time step.

There are other advantages to the RKC functions. For very large systems, the memory requirements between and even within each time step becomes a concern. Such systems occur both when extreme precision is needed and when multiple dimensions are involved. `ode15s` eventually returns an error do to memory issues. RKC is still able to solve them though the solution time can be quite long. Another advantage is with integro-differential equation systems. The explicit nature in which RKC handles the non-reaction portion becomes very advantageous when the problem matrix is non-sparse and has many more interactions between points.

### 8.2.7 Summary

These results demonstrate several important points. The two-stage method offers a definite advantage for advection problems and adds overall flexibility to the program. The positivity preserving filter requires more computation but is necessary when a reasonable amount of advection is present. Employing the IMEX method for reactive transport problems offers a strong advantage when stiffness due to the reaction coefficients is significant. The optimal choice of error correction varies on the problem type. Relying on only stability rather than actual error correction can often result in acceptable solutions and at least offers a reasonable first guess. In comparison with existing methods the RKC program demonstrates a distinct advantage in several situations but it cannot claim to be the best in all categories.

Beyond just the RKC methods implemented the problem definition function has some significance itself. It can be used to solve any basic advection-diffusion-reaction problem with various spatial discretizations with many other features that can be engaged as necessary. And the problem template can be easily modified to input more complex

reaction equations, boundary conditions, etc. This flexibility allowed the above examples to be quickly tested and compared.

The multidimensional structure that is adaptable to any number of dimensions is one of the most substantial parts of the code. It has the potential to be employed in other contexts, further increasing the utility of the program. The programs developed in this section form the basis for much of the later work.

## 9.0 Population Balance Systems

Population balance systems can describe many phenomena from polymer formation to aerosols that are generated in numerous natural and industrial processes. This section describes how the various phenomena are modeled numerically and discusses how the techniques developed above can be applied to solve such problems.

### 9.1 Notation

First off it is important to describe the notation and carefully define each term that is used. The notation employed here is roughly based on that used in [Seinfeld & Pandis, 2006]. It is often helpful to consider the units when introducing each quantity; they are expressed here with generic abbreviations, L for length, M for mass. The number concentration,  $N$ , refers to the total number of particles that are contained in some control volume. For this definition the units are  $1/L^3$  but the number concentration can actually refer to several different quantities. Technically each particle is composed of some finite number of molecules so the number concentration does have a rigorous meaning. In practice, though, it is infeasible to consider the size of each particle in this way as there could be from thousands to billions of molecules in each particle.

One way to handle this difficulty is to divide the distribution into arbitrarily size ranges. These sizes could be determined by length, volume, mass, or some other factor, which we can generically denote as  $s$  (and assume that it has units of S which is understood to represent one of the abovementioned units). Then  $N$  can be specified as  $N(s)$  to indicate the independent size variable. But if the bins are different sizes comparison becomes difficult. Thus it seems wise to scale the number concentration by the span of the bins,  $\Delta s$ . This can be made even more useful by taking the limit as  $\Delta s$  goes to zero to give us  $ds$ . Now a new term can be defined,  $n_N(s)$ , the number distribution (units  $1/L^3 1/S$ ). Then  $n_N(s)ds$  is the number of particles per unit volume from size  $s$  to  $s+ds$ . This can be integrated over all sizes to give the cumulative number concentration,

$$N(s) = \int_0^s n_N(s') ds'$$

which is defined as the number of particles of size smaller than  $s$  per unit volume. And if this is integrated over all possible sizes it gives the total particle concentration.

Besides using a distribution based on the number of particles, other bases can be used as well. Volume and mass are two popular choices and they are represented as  $n_V(s)$  ( $L^3/L^3 1/S$ ) and  $n_m(s)$  ( $M/L^3 1/S$ ) respectively. These can of course be converted back into the number concentration by their relationship via the independent variable  $s$ . A few small examples can help clarify these ideas. Consider a number distribution based on the particle diameter,  $D_p$ ; this would be written  $n_N(D_p)$ . If this were to be converted to a volume distribution we would consider the particles in the infinitesimal size range

$n_N(D_p)dD_p$ . Each particle in the range has a volume of  $1/6 \pi D_p^3$  so the total volume they occupy is simply  $1/6 \pi D_p^3 n_N(D_p)dD_p$ . Thus the conversion is simply

$$n_V(D_p) = \frac{1}{6} D_p^3 n_N(D_p)$$

where  $n_V(D_p)dD_p$  represents the volume occupied by particles per unit volume in the size range from  $D_p$  to  $D_p+dD_p$ . Now consider if we have a number distribution based on particle mass,  $m$ , written  $n_N(m)$  (units  $1/L^3 1/M$ ). To convert this to a mass distribution with mass as the independent variable as well we first note that in the infinitesimal size range  $n_N(m)dm$  the particles have a mass of  $m$  so the conversion is just

$$n_m(m) = mn_N(m)$$

where  $n_m(m)dm$  represents the mass of particles per unit volume in the size range from  $m$  to  $m+dm$ .

Additionally the sizes are often expressed in terms of the logarithm of the size parameter. For example  $n_N(\ln D_p)d\ln D_p$  represents the number of particles in the size range from  $\ln D_p$  to  $\ln D_p+d\ln D_p$ . Note that technically the logarithm must be taken of a dimensionless quantity so there is an implied reference size of 1 (with the same units as the size quantity) for the quantities in the logarithm. This also means that  $n_N(\ln s)$  has units  $1/L^3$ . These quantities can be converted to other functions of the independent variable by the following relationship: Suppose  $u$  and  $v$  are functions of  $s$ . Then the size distribution functions are related by

$$n(u) = n(v) \frac{dv/ds}{du/ds}$$

So for example we have

$$n_N(\ln m) = mn_N(m)$$

These different types of distributions are used in various situations to display information as clearly as possible. There are several common types that appear in the literature. Hereafter four different distributions are considered that encompass most of the mathematical situations that arise in solving population balance systems. Abbreviated notations are used to enhance the clarity in the later sections. The four types are summarized in Table 9.1 below.

**Table 9.1: Standard Distributions**

Distribution	Full Notation	Distribution Basis	Size Variable	Units
$n$	$n_N(m)$	number	$m$ , mass	$1/L^3 1/M$
$q$	$n_m(m)$	mass	$m$ , mass	$1/L^3$
$v$	$n_N(\mu)$	number	$\mu$ , $\ln(\text{mass}/m_{\text{ref}})$	$1/L^3$
$p$	$n_m(\mu)$	mass	$\mu$ , $\ln(\text{mass}/m_{\text{ref}})$	$M/L^3$

The  $m_{\text{ref}}$  normalizing parameter is assumed to be unity and is omitted unless otherwise noted. Mass is often used as a variable since it is conserved. The key point to remember is that they all represent an amount of particles per volume over a given size range.

## 9.2 Basic Phenomena

Some of these concepts were discussed briefly in Section 3.2.1. Here they are developed more fully. The standard array of phenomena that particles undergo includes condensation/evaporation and coagulation/splitting in addition to all of the normal transport phenomena and sources/sinks.

### 9.2.1 Condensation/Evaporation

All particles are composed of some large number of molecules and there may be many different species in each particle. Species surrounding the particles can condense upon them and the species in the particle can evaporate into the surrounding fluid. These phenomena will change the size of each particle and thus change the size distribution,  $n(s, t)$  (where  $s$  is some size variable and  $t$  is time). The actual transport mechanisms can be quite varied but the key features are that the driving force is the difference in concentrations or partial pressures and that the area available for flux is dependent on the particle size.

A brief summary of the important mass transfer properties follows. The key when considering particles of finite size is the mean free path of the surrounding fluid,  $\lambda$ , relative to the particle radius,  $R_p$ , which are related by the Knudsen number,

$$Kn = \frac{\lambda}{R_p}.$$

If the Knudsen number is much greater than one, we are in the kinetic regime where the particle moves around such that the surrounding molecules are discrete objects. As the Knudsen number goes to zero, the continuum regime is approached where the particle is so large that the surrounding molecules act as one continuous fluid.

The simpler case is the continuous regime which can be described by the familiar transport equations. For a spherical particle the unsteady-state diffusion of species A is described by

$$\frac{\partial c}{\partial t} = -\frac{1}{r^2} \frac{\partial}{\partial r} r^2 J_{A,r}$$

where  $J_{A,r}$  represents the molar flux at position  $r$ . The molar flux through the air under dilute conditions is simply

$$J_{A,r} = -D_A \frac{\partial c}{\partial r}.$$

If  $c_\infty$  represents the bulk concentration far from the particle and  $c_s$  the vapor phase concentration at the surface, then the above differential equation can be solved using these concentrations as boundary conditions. At steady-state the familiar relation is found,

$$\frac{c(r) - c_\infty}{c_s - c_\infty} = \frac{R_p}{r}.$$

Using the steady-state turns out to be acceptable to use since the diffusional flux is orders of magnitude larger than the rate of particle shrinkage. The profile near the surface of the particle retains the steady state character throughout any actual condensation/absorption processes.

The flow of A toward the particle (using the convention of a normal vector pointing outward) is given multiplying the flux by the surface area,

$$F = -4\pi R_p^2 J_{A,r} \Big|_{r=R_p}$$

which becomes, using the pseudo-steady-state approximation described above,

$$F_{continuum} = 4\pi R_p D_A (c_\infty - c_s).$$

This expression is valid only in the continuum regime but the same basic driving force is present in all cases. The adjustment to other regimes can be made by employing a function of the particle size and an accommodation coefficient,  $\alpha$ . The values for  $\alpha$  range from zero to one and it accounts for the probability of a molecule sticking to a particle that it encounters. In the kinetic regime the adjusted equation for molecular flow is

$$F_{kinetic} = \left( \frac{8k_B T}{\pi m_A} \right)^{1/2} \pi \alpha R_p^2 (c_\infty - c_s)$$

where  $k_B$  is Boltzmann's constant and  $m_A$  is the mass of one molecule of A. This equation is based on molecules moving randomly and striking a given area. For Knudsen numbers in the transition regime between fully continuous and fully kinetic regimes, there are many adjustment factors that are generally expressed as functions of Kn and  $\alpha$ . Several examples can be found in [Seinfeld & Pandis, 2006]. The overall expression for molecular flow then becomes

$$F = \frac{dm}{dt} = 4\pi R_p D_A f \text{ Kn}, \alpha (c_\infty - c_s) = \frac{4\pi R_p D_A}{RT} f \text{ Kn}, \alpha (p_A - p_{eq,A}). \quad (9.1)$$

The partial pressure version is also commonly used, where  $p_A$  is the bulk partial pressure of A,  $p_{eq,A}$ , is the equilibrium concentration at the surface, and  $T$  and  $R$  are the temperature and ideal gas constant, respectively.

With a general equation describing the mass transfer in place, the actual growth of particles can be considered. The growth rate, denoted  $I_s$ , where  $s$  indicates the size variable, is simply the rate of change of that  $s$ . Thus  $I_m$  is simply equation (9.1) and the RHS of it can be rewritten in terms of the mass to give

$$I_m(m, t) = \frac{dm}{dt} = \frac{2 \cdot 6^{1/3} \pi^{2/3} D_A}{\rho^{1/3} RT} m^{1/3} f \text{ Kn}, \alpha (p_A - p_{eq,A}). \quad (9.2)$$

where the density,  $\rho$ , will remain constant as long as there is only one component in the particle. In addition there is often one factor that affects the concentration near the surface known as the Kelvin effect [Seinfeld & Pandis, 2006]. This factor takes into account the curved nature of the particles and has the form

$$p_{A,eq} = p_{A,eq}^{flat} \exp\left(\frac{2\sigma M_w}{R_p \rho RT}\right)$$

where  $p_{A,eq}^{flat}$  is the equivalent pressure for a flat surface,  $\sigma$  is the surface tension,  $M_w$  is the molecular weight of the condensing species.

To a first approximation the Kelvin effect can be neglected. If this is combined with a constant concentration near the surface then the driving force is constant. Overall this means that the growth equation can be considered in its simplest form: a constant multiplied by a size parameter raised to some power. To limit the number of possible configurations let us just consider the mass growth factor with mass as the size variable.

$$I_m(m, t) = A_d m^{1/3}$$

where  $A_d$  is a constant.

There are a couple of other forms that the growth equations can take. The most common are surface reaction controlled growth and volume reaction controlled growth. They can be seen in detail in Gelbard & Seinfeld, 1978 and references cited therein. In comparison the above reaction can be referred to as diffusion controlled. Boiled down to the simplest form, the surface and volume reaction controlled growth laws are, respectively,

$$I_m(m, t) = A_{sr} m^{2/3} \quad \text{and} \quad I_m(m, t) = A_{vr} m.$$



Certainly the most important point when developing a solution method is the functional relationship to the size parameter. These simplified growth equations are useful as a first approximation and also because they generally have analytical solutions available for comparison with numerical approximations. These numerical methods can then be used to attack the more realistic equations.

If we now consider an infinitesimal slice of some size distribution,  $\Delta s$ , the number of particles contained in that slice at time  $t$  is  $n(s,t)\Delta s$ . If there is some growth of particles then the number of particles in that slice will be increasing due to the growth of smaller particles and decreasing due to the growth of particles out of that size range. This depends upon the rate of growth multiplied by the amount of particles at either boundary of the slice effectively resulting in a flux along the size dimension. This all changes with time so that after a time of  $\Delta t$  there will be a change in the amount of particles to  $n(s,t+\Delta t)\Delta s$ . Since this change is entirely due to the flux of particles at the boundaries of the slice the complete equation can be written as

$$n(s,t+\Delta t)\Delta s - n(s,t)\Delta s = I_s(s-\frac{1}{2}\Delta s,t) n(s-\frac{1}{2}\Delta s,t) \Delta t - I_s(s+\frac{1}{2}\Delta s,t) n(s+\frac{1}{2}\Delta s,t) \Delta t$$

After rearrangement and taking the limits as  $s$  and  $t$  go to zero we have

$$\frac{\partial n(s,t)}{\partial t} = -\frac{\partial}{\partial s} I_s(s,t)n(s,t)$$

which is known as the condensation equation. This can be written in terms of any of the distributions and size variables discussed in the previous section.

In the literature the following notation is often used for growth when mass is the size variable

$$\frac{dm}{dt} = I_m(m,t) = H(m,t)m.$$

When  $H$  is a constant, this becomes the law for simple volume (or mass) controlled growth.

### 9.2.2 Coagulation

Another important phenomenon that particles undergo is coagulation which considers particles of any size coming together to form larger ones. As discussed in Section 3.2.1 the relationship between two particles that results in their coagulation can be summarized with a coagulation coefficient  $\beta$ . The basic form of this can be developed by once again considering the Knudsen number-determined regimes that define the system. If we return to the definition of  $N$  as the number concentration of particles then the continuum regime can be described by the diffusion equation,

$$\frac{\partial N(r,t)}{\partial t} = \frac{\partial}{\partial r} \left( \frac{D}{r^2} \frac{\partial N(r,t)}{\partial r} \right) = D \left( \frac{\partial^2 N(r,t)}{\partial r^2} + \frac{2}{r} \frac{\partial N(r,t)}{\partial r} \right)$$

where  $D$  is the diffusion coefficient (assumed constant) and  $r$  is the distance from the center of one particle. This model is based on assuming that one particle remains stationary and another particle will diffuse toward it and stick to it. Also this only allows for one size of particle. This equation can be solved with appropriate boundary conditions to find  $N(r,t)$ . With this the rate at which particles collide is effective surface area multiplied by the flux,

$$J = D \frac{\partial N}{\partial r} \Big|_{r=2R_p}$$

Note that the critical radius is  $2R_p$  since this is the effective area where two particles of radius  $R_p$  would interact. Thus the collision rate is

$$F_{collision} = 8\pi R_p D N_0 \left( 1 + \frac{2R_p}{\sqrt{\pi D t}} \right)$$

where  $N_0$  is the initial particle concentration. The steady state value is often adequate in most situations of interest. This can be expanded to the general case where both particles are moving and may be of different sizes. The effective diffusion coefficient can be shown to be simply the sum of the diffusion coefficients of each particle. Also on this case the effective radius becomes the sum of the two particle radii. Combining these facts the steady state collision rate becomes

$$F_{coagulation} = 4\pi(R_{p1} + R_{p2})(D_1 + D_2)N_1N_2$$

Note that this rate depends on both concentrations. The rate in the first equation is based on type 2 particles hitting one single particle of type 1. This rate needs to be multiplied by the total number of type one particles so the final form is achieved. The first few terms in the equation can be lumped into a coagulation coefficient,  $\beta$ , to give the familiar coagulation rate expressed in general form as

$$F_{coagulation} = \beta(s_1, s_2)N_1N_2 \tag{9.3}$$

For the case above we have

$$\beta(D_{p1}, D_{p2}) = 2\pi(D_{p1} + D_{p2})(D_1 + D_2)$$

In the kinetic regime ( $Kn \rightarrow \infty$ ) the interactions follow from basic kinetic theory give

$$\beta_{12}(R_p, t) = \pi(R_{p1} + R_{p2})^{1/2} (\bar{c}_1^2 + \bar{c}_2^2)^{1/2}$$

where  $\bar{c} = 8k_B T / \pi m$  <sup>1/2</sup> is the mean kinetic velocity. The transition regime between the two extremes is generally represented by multiplying the continuum coagulation coefficient by some parameter.

The coagulation equation was derived from the basic coagulation rate previously in Section 3.2.1 but the main points are reiterated here for completeness. Coagulation results in two main effects on a given particle size,  $s$ : production due to smaller particles colliding and creating a particle of size  $s$  and depletion due to  $s$  size particles colliding with other particles. The continuous approximation is again made as discussed in the previous section. The critical quantity is then the concentration of particles in the size range  $s$  to  $s+ds$  which is  $n(s,t)ds$  where  $n$  can have any basis and  $s$  is any size variable. Note that for the coagulation portion this needs to be a conserved quantity (e.g. diameter is not appropriate).

The rate of production of an  $s$  size particle results from one particle of size  $a$  and one of size  $s-a$ . The total rate is then the integral over the range of all sizes  $a$  smaller than  $s$ . To eliminate double counting a factor of  $1/2$  must be used. The production is then

$$F_{coag,prod} = \frac{1}{2} \int_{s_{min}}^{s-s_{min}} \beta(s-a, a)n(s-a, t)n(s, t)da .$$

Here  $s_{min}$  is some minimum sized particle determined by the physics of the specific particle type. The decrease in  $s$  size particles due to coagulation is the rate that  $s$  size particles agglomerate with all other size particles. This is

$$F_{coag,depl} = n(s, t) \int_{s_{min}}^{\infty} \beta(a, s)n(a, t)da .$$

These two terms are combined to give the coagulation equation,

$$\frac{\partial n(s, t)}{\partial t} = \frac{1}{2} \int_{s_{min}}^{s-s_{min}} \beta(s-a, a)n(s-a, t)n(s, t)da - n(s, t) \int_{s_{min}}^{\infty} \beta(a, s)n(a, t)da .$$

### 9.2.3 Other Phenomena

There are several other phenomena that can be important to population balance systems. The most complicated to describe mathematically is fragmentation which was explained in Section 3.2.1 and is less common than the other effects listed here.

Source and depletion terms are fairly easy to handle. They are usually considered to have some specified function and are denoted  $S(s,t)$  and  $R(s,t)$  respectively. They may have the form of some type of reaction.

Nucleation refers to the formation of particles of a certain size in a given medium, often enhanced due to interaction with foreign material. There are many mechanisms that explain this phenomenon for different systems but for the current purpose they will be quantified by one term,  $Q_0(s)$ . This results in particles that are of some minimum size,  $s_{\min}$ , and only of this size. This means that the term must be multiplied by a delta function that only has value when the size variable is exactly  $s_{\min}$ .

In addition, particles are subject to the standard transport effects, diffusion and advection. They generally take the same form as the molecular analogs developed in previous sections.

Combining all of the population balance phenomena results in what is known as the general dynamic equation [Seinfeld & Pandis, 2006]:

$$\begin{aligned} \frac{\partial n(s,t)}{\partial t} = & \frac{1}{2} \int_{s_{\min}}^{\infty-s_{\min}} \beta(s-a,a)n(s-a,t)n(s,t)da - n(s,t) \int_{s_{\min}}^{\infty} \beta(a,s)n(a,t)da \\ & - \frac{\partial}{\partial s} I_s(s,t)n(s,t) + Q_0(s)\delta(s-s_0) + S(s) - R(s) \end{aligned} \quad (9.4)$$

The transport terms are not included here. They effectively add other dimensions to the problem as they result in changes in spatial coordinates and not along the size distribution. These issues will be discussed in later sections. Also note that if the distribution changes to mass or the size variable undergoes a transformation the form of the equation will change to some extent.

### 9.3 Equation Forms & Analytical Solutions

The basic form of (9.4) changes if different distribution types and size variables are used. Also the multi-component case alters the mathematical form and the solution techniques used. Using the four distribution/size variable systems mentioned above in Table 9.1 the forms and available solutions for the parts of the general dynamics equation are discussed.

#### 9.3.1 Condensation Equation

The standard condensation equation describes the change in particle size number distribution due to the evaporation or condensation of species onto the existing particles. The basic form is repeated here:

$$\frac{\partial n(s,t)}{\partial t} = - \frac{\partial}{\partial s} I_s(s,t)n(s,t) \quad (9.5)$$

Recall that  $n$  can be a distribution with a given basis (e.g. number or mass) but the form of (9.5) will change.  $s$  is a size parameter (e.g. mass or diameter) over which the distribution varies and it can alter the form as well. This is the simplest form and the four specific cases will be considered after this basic equation.

It is important to note that the particle can be composed of several different species each possessing a unique growth term. However, it is worth analyzing the one component case thoroughly to see what can be learned.

Equation (9.5) is a first order hyperbolic differential equation. Such equations can generally be solved using the method of characteristics. To see this, first expand the equation:

$$\frac{\partial n(s,t)}{\partial t} + I_s(s,t) \frac{\partial n(s,t)}{\partial s} + \frac{\partial I_s(s,t)}{\partial s} n(s,t) = 0.$$

Now assume that the growth function is differentiable in  $s$  and notate it as  $I_s'$ . Also define the initial distribution by some function

$$n(s, 0) = n_0(s).$$

The goal of the method of characteristics is to change the coordinates to a set in which the PDE becomes an ODE along certain curves, which are known as characteristic curves. For this example, consider the new variables to be  $(s_0, a)$ . As one might suspect by the notation, the  $s_0$  variable will remain constant along the characteristics.

To begin, it is noted that if

$$\frac{ds}{da} = I_s(s,t) \quad \text{and} \quad \frac{dt}{da} = 1$$

(which are known as the characteristic equations) then we have

$$\frac{dn}{da} = \frac{ds}{da} \frac{\partial n}{\partial s} + \frac{dt}{da} \frac{\partial n}{\partial t} = \frac{\partial n(s,t)}{\partial t} + I_s(s,t) \frac{\partial n(s,t)}{\partial s} = -I_s' n(s,t)$$

so we have the ODE and initial condition

$$\frac{dn}{da} + I_s' n = 0, \quad n(0) = n_0(s). \quad (9.6)$$

Now the two characteristic equations are simple ODEs with respective initial conditions

$$s(a = 0) = s_0 \quad \text{and} \quad t(a = 0) = 0.$$

Note that the  $s = s_0$  points are points where  $t = 0$  in the original coordinates. Solving the ODE for  $t$  gives simply

$$t = a$$

and the other ODE is solved with the given  $I_s$  to give the relationship between  $s$  and  $s_0$ . Now the ODE (9.6) can be solved and the original coordinates can be substituted back in to give the solution,  $n(s,t)$ .

A simple example will help clarify this. Using the abbreviated notation introduced in Section 9.1, consider a number distribution based on mass,  $n(m,t)$  and a growth law that is directly proportional to mass,  $I_m(m,t) = Hm$ . If  $H$  is constant, the expanded condensation equation and initial condition has the form

$$\frac{\partial n(m,t)}{\partial t} + Hm \frac{\partial n(m,t)}{\partial m} + Hn(m,t) = 0, \quad n(m,0) = n_0(m).$$

After switching the coordinates,  $(m,t) \rightarrow (m_0,a)$ , the characteristic equations can be solved to reveal

$$t = a \quad \text{and} \quad m = m_0 e^{Ht}.$$

The ODE in  $a$ ,

$$\frac{dn}{da} + Hn = 0, \quad n(0) = n_0(m),$$

is then solved to give

$$n(m_0, a) = n_0(m_0) e^{-Ha} \quad \Rightarrow \quad n(m, t) = n_0(m e^{-Ht}) e^{-Ht}. \quad (9.7)$$

The other three types of distributions mentioned above each have slightly different forms but can be solved using the same methods as above. All four equations and their solutions for constant  $H$  are summarized below in Table 9.2. The subscript zero on the distribution variables indicates the initial distribution function.

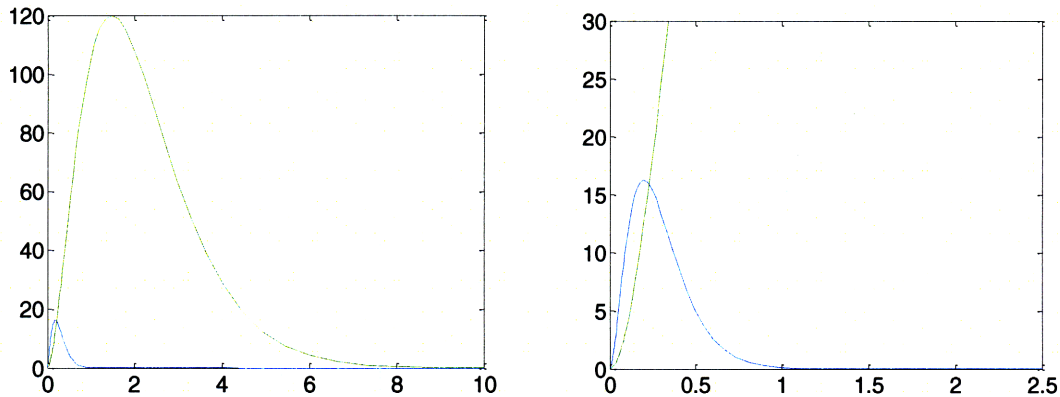
**Table 9.2: Solutions for basic forms of the Condensation Equation**

Condensation Equation		Solution, $H=\text{const}$
$\frac{\partial n}{\partial t} = -\frac{\partial}{\partial m} Hmn = -m \frac{\partial Hn}{\partial m} - Hn$	(9.8)	$n = e^{-Ht} n_0(me^{-Ht})$
$\frac{\partial q}{\partial t} = -m \frac{\partial}{\partial m} Hq$	(9.9)	$q = q_0(me^{-Ht})$
$\frac{\partial v}{\partial t} = -\frac{\partial}{\partial \mu} Hv$	(9.10)	$v = v_0(\mu - Ht)$
$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial \mu} Hp + Hp$	(9.11)	$p = e^{Ht} p_0(\mu - Ht)$

It is worthwhile to briefly observe how these functions actually evolve over time. Consider the commonly used exponential initial distribution,

$$n_0(m) = \frac{N_0}{m_{\text{ref}}} e^{-m/m_{\text{ref}}}$$

$N_0$  is the initial number of particles in the system and  $m_{\text{ref}}$  is some reference size, often unity. With this initial distribution the results look as follows plotted against particle diameter.



**Figure 9.1: Condensation Growth, Mass Distribution, with closeup**

The initial Condition is the dashed line and the final condition is the solid line.

This is completed over a time interval of 1.0 using 100 as the initial number of particles. Note the large change in size range over a small time. This hints that it may be advantageous to employ logarithmic coordinates for the size variable.

Things become more interesting when there are multiple species in each particle. Before beginning it is important to point out the assumption involved in all of the multi-component mathematics is that all particles of the same size have the same composition

of species. This is often referred the internally well-mixed assumption [Diaz *et al.*, 1999].

First off, denote  $n_i(s,t)$  as the number distribution of component  $i$ . The condensation equation for number distribution with mass as the size variable has the familiar form

$$\frac{\partial n_i(m,t)}{\partial t} = -\frac{\partial}{\partial m} I_{m,i}(m,t)n_i(m,t) .$$

Note that the growth law also is specific to a particular species. By definition the relationship between the component mass distribution,  $q_i(m,t)$ , and the total mass distribution is

$$q_i(m,t) = \frac{m_i}{m} q(m,t)$$

where  $m_i$  is the fraction of  $i$  in particles of size  $m$ . The size variable must be either mass or volume (if the density is constant) so that the fraction is consistent. This of course alters the condensation equation again. Consider the growth law used previously, modified for the individual species case,

$$I_{s,i} = \frac{ds_i}{dt} = H_i s .$$

Note that since the sum of the growth of  $r$  individual species must equal the total growth we have

$$\sum_{i=1}^r H_i = H$$

and this  $H$  is the same as the one used in the one-species example. Now using the above growth law and starting from (9.9) the multi-component mass distribution can be found:

$$\begin{aligned} \frac{\partial q_i}{\partial t} &= \frac{m_i}{m} \frac{\partial q}{\partial t} = -m_i \frac{\partial}{\partial m} \left( H \frac{m}{m_i} q_i \right) \\ \frac{\partial q_i}{\partial t} &= -m_i \left( \frac{m}{m_i} \frac{\partial H q_i}{\partial m} - H q_i m \frac{1}{m_i^2} \frac{\partial m_i}{\partial m} + \frac{H q_i}{m_i} \right) \end{aligned}$$

and since

$$\frac{\partial m_i}{\partial m} = \frac{dm_i / dt}{dm / dt} = \frac{H_i m}{H m} = \frac{H_i}{H}$$

we have



$$\frac{\partial q_i}{\partial t} = -m \frac{\partial H q_i}{\partial m} + H_i q - H q_i. \quad (9.12)$$

Note that if these equations are summed over all  $i$  or if there is only one component the equation reduces to (9.9) as expected. Also it is important to observe that nowhere was  $H$  assumed to be constant. Indeed it could be any function of size or time.

Similar expansions can be completed on the other types of distributions and all four are put forth in below:

$$\frac{\partial n_i}{\partial t} = -m_i \frac{\partial}{\partial m} \left( H \frac{m}{m_i} n_i \right) - H n_i = -m \frac{\partial}{\partial m} H n_i + H_i n - 2 H n_i. \quad (9.13)$$

$$\frac{\partial q_i}{\partial t} = -m_i \frac{\partial}{\partial m} \left( H \frac{m}{m_i} q_i \right) = -m \frac{\partial}{\partial m} H q_i + H_i q - H q_i. \quad (9.14)$$

$$\frac{\partial v_i}{\partial t} = -e^{\mu_i - \mu} \frac{\partial}{\partial \mu} H e^{\mu - \mu_i} v_i = -\frac{\partial}{\partial \mu} H v_i + H_i v - H v_i. \quad (9.15)$$

$$\frac{\partial p_i}{\partial t} = -e^{\mu_i - \mu} \frac{\partial}{\partial \mu} H e^{\mu - \mu_i} p_i + H p_i = -\frac{\partial}{\partial \mu} H p_i + H_i p. \quad (9.16)$$

These PDEs represent systems of  $r$  equations if  $r$  is the number of species. A paper by [Diaz *et al.*, 1999] demonstrated that analytical solutions of this problem are possible. These serve as an excellent point of comparison for the numerical methods.

Diaz *et al.* employed the method of characteristics to solve this system for any general form of  $H(s,t)$ . The expression they derived has the form

$$p_i(\mu, t) = \left( p_i(\mu_0, t_0) + p(\mu_0, t_0) e^{-\mu_0} \int_{\mu_0}^{\mu} e^{\mu'} \frac{H_i(\mu', \tau(\mu_0, t_0, \mu'))}{H(\mu', \tau(\mu_0, t_0, \mu'))} d\mu' \right) \cdot \tilde{H}(\mu_0, t_0, \mu). \quad (9.17)$$

with

$$\tilde{H}(\mu_0, t_0, \mu) = \exp \left( - \int_{\mu_0}^{\mu} \left( \frac{\partial \ln H(\mu', t)}{\partial \mu'} \right)_{t=\tau(\mu_0, t_0, \mu')} d\mu' \right).$$

The variable  $\mu_0$  corresponds to the solution to one of the characteristics as discussed at the beginning of this section. Recall that the two characteristics (modified to use the current variables) were

$$\frac{d\mu}{da} = H(\mu, t) \quad \text{and} \quad \frac{dt}{da} = 1$$

with respective initial conditions

$$\mu(a=0) = \mu_0 \quad \text{and} \quad t(a=0) = t_0$$

The latter ODE can be trivially solved while the former ODE of course depends upon the form of  $H$  which must be integrated. Ultimately the four variables,  $t$ ,  $t_0$ ,  $\mu$ , and  $\mu_0$  are all interrelated and this can be expressed by the following functions,

$$\mu = \zeta(\mu_0, t_0, t), \quad \mu_0 = \zeta^{-1}(\mu, t_0, t), \quad t = \tau(\mu_0, t_0, \mu)$$

which depend on the form of  $H$ .

The integrals in equation (9.17) may need to be numerically evaluated but the equation can provide very accurate solutions for comparison with other methods.

### 9.3.2 Coagulation Equation

The coagulation equation introduces the integral components. As described above the coagulation of particles results in two integrals due to the gain of particles of a given size by smaller particles colliding and loss due to collision of particles of a given size colliding with other particles to make larger particles. The basic equation is as follows with number concentration,  $n$  based on particle mass,  $m$ :

$$\frac{dn(m, t)}{dt} = \frac{1}{2} \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m', m} n(m-m', t) n(m', t) dm' - n(m, t) \int_{m_{\min}}^{\infty} \beta_{m, m'} n(m', t) dm'. \quad (9.18)$$

Note that the first term is effectively a convolution whereas the second integral is linear in  $n$ .

To modify this equation for the other standard bases mentioned above there are a few important points to be noted. The coagulation term,  $\beta$ , must be adjusted if the underlying size variable changes. Hereafter for any bases other than mass the new  $\beta$  is indicated with a parenthetical superscript for the new underlying variable. In addition, in the convolution integral is based on the difference in mass,  $(m - m')$  since that is the only quantity that is preserved upon collision. For all other size variables it is the difference in mass that must be transformed rather than the difference between two masses, i.e. for  $\mu = \ln(m/m_{ref})$  we have

$$m - m' \Leftrightarrow \ln e^\mu - e^{\mu'}$$

Finally, the symmetry of the convolution integral can allow for a slight simplification by noting that

$$\frac{1}{2} m \int_{m_{\min}}^{m-m_{\min}} \frac{1}{m'} \frac{dm'}{(m-m')} = \int_{m_{\min}}^{m-m_{\min}} \frac{dm'}{(m-m')}$$

However, both forms are equally tractable when handled numerically so it is largely a matter of preference which form to use.

With those preliminaries the original and three transformed versions of the coagulation equation are as follows. Recall that  $\mu = \ln(m/m_{ref})$ ,  $n(m,t)$  and  $v(\mu,t)$  are number distributions, and  $q(m,t)$  and  $p(\mu,t)$  are mass distributions.

$$\begin{aligned} \frac{dn(m,t)}{dt} &= \frac{1}{2} \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} n(m-m',t) n(m',t) dm' - n(m,t) \int_{m_{\min}}^{\infty} \beta_{m,m'} n(m',t) dm' \\ \frac{dq(m,t)}{dt} &= \frac{1}{2} m \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} \frac{q(m-m',t)}{(m-m')} \frac{q(m',t)}{m'} dm' - q(m,t) \int_{m_{\min}}^{\infty} \beta_{m,m'} \frac{q(m',t)}{m'} dm' \\ &= \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} \frac{q(m-m',t)}{(m-m')} q(m',t) dm' - q(m,t) \int_{m_{\min}}^{\infty} \beta_{m,m'} \frac{q(m',t)}{m'} dm' \\ \frac{dv(\mu,t)}{dt} &= \frac{1}{2} e^{\mu} \int_{\mu_{\min}}^{\ln(e^{\mu}-e^{\mu_{\min}})} \beta_{\ln(e^{\mu}-e^{\mu'}),\mu'} \frac{v(\ln(e^{\mu}-e^{\mu'}),t)}{e^{\mu}-e^{\mu'}} v(\mu',t) d\mu' - v(\mu,t) \int_{\mu_{\min}}^{\infty} \beta_{\mu,\mu'} v(\mu',t) d\mu' \\ &= \int_{\mu_{\min}}^{\ln(e^{\mu}-e^{\mu_{\min}})} \beta_{\ln(e^{\mu}-e^{\mu'}),\mu'} \frac{v(\ln(e^{\mu}-e^{\mu'}),t)}{e^{\mu}-e^{\mu'}} \frac{v(\mu',t)}{e^{\mu'}} d\mu' - v(\mu,t) \int_{\mu_{\min}}^{\infty} \beta_{\mu,\mu'} v(\mu',t) d\mu' \\ \frac{dp(\mu,t)}{dt} &= \frac{1}{2} e^{2\mu} \int_{\mu_{\min}}^{\ln(e^{\mu}-e^{\mu_{\min}})} \beta_{\ln(e^{\mu}-e^{\mu'}),\mu'} \frac{p(\ln(e^{\mu}-e^{\mu'}),t)}{(e^{\mu}-e^{\mu'})^2} \frac{p(\mu',t)}{e^{\mu'}} d\mu' - p(\mu,t) \int_{\mu_{\min}}^{\infty} \beta_{\mu,\mu'} \frac{p(\mu',t)}{e^{\mu'}} d\mu' \\ &= e^{\mu} \int_{\mu_{\min}}^{\ln(e^{\mu}-e^{\mu_{\min}})} \beta_{\ln(e^{\mu}-e^{\mu'}),\mu'} \frac{p(\ln(e^{\mu}-e^{\mu'}),t)}{(e^{\mu}-e^{\mu'})^2} p(\mu',t) d\mu' - p(\mu,t) \int_{\mu_{\min}}^{\infty} \beta_{\mu,\mu'} \frac{p(\mu',t)}{e^{\mu'}} d\mu' \end{aligned}$$

An analytical solution is available in the case of constant coagulation coefficient. Equation (9.18) then simplifies to

$$\frac{dn(m,t)}{dt} = \frac{1}{2} \beta \int_0^m n(m-m',t) n(m',t) dm' - \beta n(m,t) N(t)$$

recalling that  $N(t)$  is the total number of particles in the system. Integrating over all  $m$  then gives

$$\int_0^{\infty} \frac{dn(m,t)}{dt} dm = \int_0^1 \beta \int_0^m n(m-m',t)n(m',t)dm'dm - \int_0^{\infty} \beta n(m,t)N(t)dm$$

$$\frac{dN(t)}{dt} = \frac{1}{2} \beta N(t)^2 - \beta N(t)^2 = -\frac{1}{2} \beta N(t)^2$$

This simple ODE is solved to give:

$$N(t) = \frac{N_0}{1 + \frac{1}{2} \beta N_0 t}$$

where  $N_0$  is the initial number of particles. Returning to the simplified coagulation equation we have

$$\frac{dn(m,t)}{dt} = \frac{1}{2} \beta \int_0^m n(m-m',t)n(m',t)dm' - \beta n(m,t) \frac{N_0}{1 + \frac{1}{2} \beta N_0 t}$$

$$n(m,0) = \frac{N_0}{m_0} e^{-m/m_0}$$

where the exponential initial condition is used so that the solution is achievable. This non-linear integro-differential equation is solvable. First the integrating factor is determined by taking the exponential of the coefficient of the  $n(m,t)$  term

$$IF = \exp \left( \int_0^t \frac{\beta N_0}{1 + \frac{1}{2} \beta N_0 t'} dt' \right) = 1 + \frac{1}{2} \beta N_0 t^2 .$$

Now both sides are multiplied by the integrating factor and terms are combined to give

$$\frac{\partial}{\partial t} \left( 1 + \frac{1}{2} \beta N_0 t^2 \right) n(m,t) = \frac{1}{2} \beta \left( 1 + \frac{1}{2} \beta N_0 t^2 \right) \int_0^m n(m-m',t)n(m',t)dm' .$$

A change of variables can help to elucidate the problem. Assign  $y = 1 / N_0 (1 + \frac{1}{2} \beta N_0 t)$  and  $w(m,y) = \left( 1 + \frac{1}{2} \beta N_0 t^2 \right) n(m,t)$  to give

$$\frac{\partial w(m,y)}{\partial y} = - \int_0^m w(m-m',y)n(m',y)dm'$$

Now if we assume a solution of the form  $w(v,y) = a \exp(-bmy)$  we get

$$-abme^{-bmy} = -a^2 me^{-bmy} \Rightarrow w(m,y) = ae^{-amy} .$$

Putting back in the original variables gives

$$1 + \frac{1}{2} \beta N_0 t^{-2} n(m, t) = a \exp\left(-\frac{am}{N_0 \left(1 + \frac{1}{2} \beta N_0 t\right)}\right)$$

and solving for when  $t = 0$  gives

$$n(m, 0) = a \exp\left(-\frac{am}{N_0 \left(1 + \frac{1}{2} \beta N_0 \cdot 0\right)}\right) = \frac{N_0}{m_0} e^{-m/m_0}$$

so  $a = N_0/m_0$ . Putting everything together gives the solution

$$n(m, t) = \frac{N_0}{m_0} \frac{1}{1 + \frac{1}{2} \beta N_0 t^{-2}} \exp\left(-\frac{m}{m_0 \left(1 + \frac{1}{2} \beta N_0 t\right)}\right)$$

When multiple components are involved some care must be taken in modifying the coagulation expressions. It is best illustrated by starting from the standard coagulation equation to derive the mass-based partial component version.

$$\frac{dn(m, t)}{dt} = \frac{1}{2} \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m', m'} n(m-m', t) n(m', t) dm' - n(m, t) \int_{m_{\min}}^{\infty} \beta_{m, m'} n(m', t) dm'$$

Using  $q_i = m_i n$  on the LHS and multiplying by  $m_i$  gives

$$\frac{dq_i(m, t)}{dt} = \frac{1}{2} m_i \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m', m'} n(m-m', t) n(m', t) dm' - q_i(m, t) \int_{m_{\min}}^{\infty} \beta_{m, m'} n(m', t) dm'$$

The  $m_i$  can be brought into the convolution integral by defining it as the sum of the mass of  $i$  in each of two coagulating particles,

$$m_i = m_i|_{m'} + m_i|_{m-m'}$$

where  $m_i|_m$  is the mass of  $i$  in a particle of size  $m$ . Each  $n$  is then converted to  $q$  by  $q(m) = m n(m)$

$$\frac{dq_i(m, t)}{dt} = \frac{1}{2} \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m', m'} \left( m_i|_{m'} + m_i|_{m-m'} \right) \frac{q(m-m', t)}{m-m'} \frac{q(m', t)}{m'} dm' - q_i(m, t) \int_{m_{\min}}^{\infty} \beta_{m, m'} \frac{q(m', t)}{m'} dm'$$

Considering just the convolution integral, the  $m_i$ 's can be converted by  $q_i = \frac{m_i}{m} q$  and then the simplification proceeds as

$$\begin{aligned}
& \frac{1}{2} \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} m_i|_{m'} + m_i|_{m-m'} \frac{q(m-m',t)}{m-m'} \frac{q(m',t)}{m'} dm' \\
&= \frac{1}{2} \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} \left( m' \frac{q_i(m',t)}{q(m',t)} + (m-m') \frac{q_i(m-m',t)}{q(m-m',t)} \right) \frac{q(m-m',t)}{m-m'} \frac{q(m',t)}{m'} dm' \\
&= \frac{1}{2} \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} \frac{q(m-m',t)}{m-m'} q_i(m',t) dm' + \frac{1}{2} \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} \frac{q(m',t)}{m'} q_i(m-m',t) dm'
\end{aligned}$$

by symmetry these two integrals are the same so the entire coagulation equation for  $q_i$  becomes

$$\frac{dq_i(m,t)}{dt} = \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} \frac{q(m-m',t)}{m-m'} q_i(m',t) dm' - q_i(m,t) \int_{m_{\min}}^{\infty} \beta_{m,m'} \frac{q(m',t)}{m'} dm' .$$

Converting this to the other distributions yields

$$\begin{aligned}
\frac{dn_i(m,t)}{dt} &= \frac{1}{2} \int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} n(m-m',t) n_i(m',t) dm' - n_i(m,t) \int_{m_{\min}}^{\infty} \beta_{m,m'} n(m',t) dm' \\
\frac{dv_i(\mu,t)}{dt} &= \frac{1}{2} e^{\mu} \int_{\mu_{\min}}^{\ln(e^{\mu} - e^{\mu_{\min}})} \beta_{\ln(e^{\mu} - e^{\mu'}), \mu'} \frac{v(\ln(e^{\mu} - e^{\mu'}), t)}{e^{\mu} - e^{\mu'}} v_i(\mu', t) d\mu' - v_i(\mu, t) \int_{\mu_{\min}}^{\infty} \beta_{\mu, \mu'} v(\mu', t) d\mu' \\
\frac{dp_i(\mu,t)}{dt} &= e^{\mu} \int_{\mu_{\min}}^{\ln(e^{\mu} - e^{\mu_{\min}})} \beta_{\ln(e^{\mu} - e^{\mu'}), \mu'} \frac{p(\ln(e^{\mu} - e^{\mu'}), t)}{(e^{\mu} - e^{\mu'})^2} p_i(\mu', t) d\mu' - p_i(\mu, t) \int_{\mu_{\min}}^{\infty} \beta_{\mu, \mu'} \frac{p(\mu', t)}{e^{\mu'}} d\mu'
\end{aligned}$$

There are no known analytical solutions to multi-component coagulation systems but as with the condensation equation the sum of the final solutions can be used to check consistency with known one-component solutions.

## 9.4 Numerical Solutions

There are a large number of methods for solving parts of the general dynamic equation. As discussed above, analytic methods are only possible in a few special instances but they serve as a useful point of comparison. Numerical methods can be broadly divided into two types. Sectional methods divide the entire size range into discrete sections in which the various distribution functions are assumed to remain constant. The other approach is to attack the equations directly using the techniques for discretization and time integration discussed throughout this work. Both methods have advantages and

drawbacks. Sectional methods have difficulty with the discrete jump in conditions between sections and can result in artificial diffusion and other issues. The direct approach can often result in very large systems that can be impractical to handle with conventional solution techniques. It will be shown how the methods developed in this work can be applied to the direct continuous approach.

### 9.4.1 Condensation Equation

First off it is noted that the condensation equation has a similar form as the one-dimensional advection equation,

$$\frac{\partial c}{\partial t} + \frac{\partial vc}{\partial x} = 0$$

where  $v$  is the velocity and may be a function of time ( $t$ ) and position ( $x$ ). Therefore the same techniques will be applied where possible.

The solution procedure outlined in previous sections is summarized here: The distribution function ( $n$ ,  $q$ , etc.) is discretized according to a given step size,  $h$ . Call this discretized vector  $w$ . A function is then applied to this vector that incorporates the spatial discretization method, the boundary conditions, and any other properties unique to the solution. This results in a function,  $f(w(s,t))$ , that represents the RHS of the condensation equation. This function is then integrated over time via a prescribed method beginning from the discretized initial distribution,  $w_0$ .

For the spatial discretization we once again have the need for positivity preservation so the technique developed in Section 6.1 will be used. For the time integration the 2-stage Runge-Kutta-Chebyshev method can be applied. The solution of the condensation equation actually requires very little modification of the methods as compared to the transport equations. The important points in implementation are now discussed.

The boundary conditions are actually fairly straightforward. For the lower limit there is a Diriclet condition that sets the concentration of particles at this limit to zero. This effectively states that there are no particles below a certain size. For the upper limit it is proper to apply an open boundary condition. This is because the condensation equation is first order in the size variable so there can only be one boundary condition.

Recall from the earlier chapters that the stability bounds of a given time integration method (along with the error control methods) and the eigenvalues of the spatially discretized portion determine the time step size and ultimately the solution time. As such, considering the eigenvalue arrangement is always important.

To this end, consider the following versions of the condensation equation. Depending on the type of distribution the form of the condensation equation changes and it changes again when it is applied to individual species. The four versions are restated here for convenience.

**Table 9.3: Condensation Equation Forms**

Distribution	Individual species	Total
$n_i(m,t)$	$\frac{\partial n_i}{\partial t} = -m \frac{\partial}{\partial m} H n_i + H_i n - 2H n_i$	$\frac{\partial n}{\partial t} = -m \frac{\partial}{\partial m} H n - H n$
$q_i(m,t)$	$\frac{\partial q_i}{\partial t} = -m \frac{\partial}{\partial m} H q_i + H_i q - H q_i$	$\frac{\partial q}{\partial t} = -m \frac{\partial}{\partial m} H q$
$v_i(\mu,t)$	$\frac{\partial v_i}{\partial t} = -\frac{\partial}{\partial \mu} H v_i + H_i v - H v_i$	$\frac{\partial v}{\partial t} = -\frac{\partial}{\partial \mu} H v$
$p_i(\mu,t)$	$\frac{\partial p_i}{\partial t} = -\frac{\partial}{\partial \mu} H p_i + H_i p$	$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial \mu} H p + H p$

The additional terms are roughly analogous to first order reaction terms in a standard transport equation. These terms shift the eigenvalues and this effect is easily seen when considering the set form of the equations at each grid point. Consider a system with three components, A, B, and C and three different values for  $H_i$ . For the  $n$  distribution this amounts to the matrix

$$\begin{bmatrix} H_A - 2H & H_A & H_A \\ H_B & H_B - 2H & H_B \\ H_C & H_C & H_C - 2H \end{bmatrix}$$

where  $H$  is the sum of the  $H_i$ 's. For the  $q$  and  $v$  distributions the matrix has each  $-2H$  replaced by  $-H$  and the  $p_i$  omits the  $H$ . This matrix is multiplied by the vector  $w$  at every grid point recalling that  $w$  is arranged such that the values for the different species at each grid point are adjacent, i.e.

$$w = \left[ \dots \quad w_j^A \quad w_j^B \quad w_j^C \quad w_{j+1}^A \quad w_{j+1}^B \quad w_{j+1}^C \quad \dots \right]^T$$

where the  $j$ 's represent the grid points. It is then the eigenvalues of the above  $H$  matrix that are of importance. The eigenvalues due to the discretization of the first derivative portion will have the same arrangement but will be translated by the eigenvalues of the  $H$  matrix.

Initially it may seem that this factor could become substantial if there is a large difference between the individual  $H_i$ 's relative to their sum (i.e. the components are of similar magnitude but of opposite sign). However, closer analysis of the matrix reveals that the largest magnitude eigenvalues scale with the sum,  $H$ , not the individual  $H_i$ 's. Indeed the eigenvalues for the abovementioned  $H$  matrices are simply

Distribution	$H$ matrix eigenvalues
$n_i(m,t)$	$-H, -2H$
$q_i(m,t)$	$0, -H$

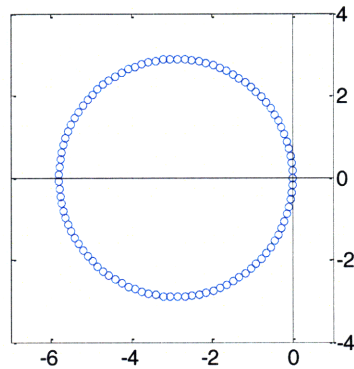


$v_i(\mu, t)$	$0, -H$
$p_i(\mu, t)$	$0, H$

The  $H$  matrix for each grid point can be put along the diagonal of a larger matrix by which the entire  $w$ -vector can be multiplied. Of course for constant  $H$ 's the eigenvalues of the  $H$  matrix for each grid point become the eigenvalues of the larger matrix.

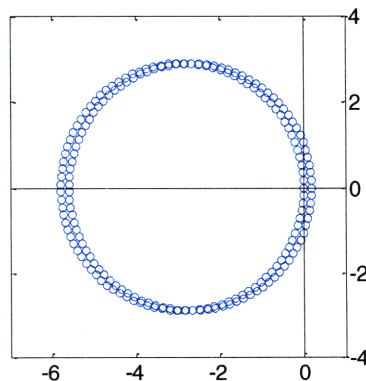
The advective portion scales with  $H$  as well so the “reaction” term associated with the  $H$  matrix does not add any extraordinary stiffness.

To see how these factors affect the stability, consider the spectrum of eigenvalues for a simple system. Take a  $p$  distribution with uniform grid spacing of  $h = 0.069$  ranging over masses  $[0.00001 \ 0.01]$ . This is a two-component system with growth rates of  $H_A = -0.1$  and  $H_B = 0.3$  giving  $H = 0.2$ . For the advective portion consider a third-order upwind biased discretization. The eigenvalues can be seen in Figure 9.2 below.



**Figure 9.2: Eigenvalues for Condensation Equation, "Advective" Portion**

This is the familiar arrangement from the eigenvalues is of course the same as the upwind examples in previous chapters. Now when the “reactive” term is added the eigenvalues change a relatively small amount as can be seen below.



**Figure 9.3: Eigenvalues for Condensation Equation, Both Portions**

The same effect is seen regardless of the individual  $H$  values as long as the sum is the same. We can see that the eigenvalues are merely shifted by the two eigenvalues of the “reactive” portion,  $-H$  ( $= -0.2$ ) and  $0$ .

To summarize, the eigenvalues from the advective portion extend in both the real and imaginary directions and scale with  $H/h$  (where  $h$  is the grid spacing) and the eigenvalues from the reaction portion are all real and scale with  $H$ . From a practical standpoint this means that the advective portion is the primary concern in terms of stability.

The next issue becomes how to characterize the eigenvalues for the different distributions and spatial discretizations. For the purposes considered here there are effectively two different cases. For the  $n$  and  $q$  distributions the form of the advection operator is

$$m \frac{\partial}{\partial m}$$

and for the  $v$  and  $p$  versions the form is

$$\frac{\partial}{\partial \mu}$$

For this second case the form is identical to the standard advection operator so the eigenvalue spectrum for each spatial discretization is the same as that discussed in previous sections using Fourier-Von Neumann analysis. In the first case, however, there is a significant difference. In the second case it is advantageous to use variable grid spacing in some situations (see next section). In the case of logarithmic grid spacing the second case actually becomes identical to first.

One way to estimate the eigenvalues in the first case would be to treat the leading  $m$  like a “velocity” term. To do this one would characterize the eigenvalues using the same methods as in Section 7.1.3 and then multiply them by the largest value that  $m$  attains (i.e. the “right hand bound” on the size vector). This would be sufficient to bound the eigenvalues but is more than is necessary.

It is possible to characterize the eigenvalues more precisely but it cannot be done as neatly. The Fourier-Von Neumann analysis is no longer feasible since this is effectively a variable velocity case. Ideally we would like to be able to determine the maximum real and imaginary eigenvalues with similar parameters as in the constant velocity case.

Numerical experiments have revealed that it is possible to use the size variable range and the number of grid points. While the results do not apply to all conceivable conditions they are valid over the range of interest for the types of problems solved. This is acceptable since all that is desired is an approximation for use in determining stability. The results are summarized in Table 9.4 below. This is comparable to the constant velocity results from Table 7.1.

**Table 9.4: Approximate eigenvalue bounds for discretized variable velocity operator**

	Maximum Real Eigenvalue	Maximum Imaginary Eigenvalue
1 <sup>st</sup> Order Upwind	$\lambda_{\text{Re}}^{\text{max}} = M \cdot \left( 1 + 2 \frac{m_{\text{min}}}{m_{\text{max}} - m_{\text{min}}} \right)$	$\lambda_{\text{Im}}^{\text{max}} = M \cdot \left( \frac{1}{2} + \frac{m_{\text{min}}}{m_{\text{max}} - m_{\text{min}}} \right)$
2 <sup>nd</sup> Order Central	$\lambda_{\text{Re}}^{\text{max}} = 0$	$\lambda_{\text{Im}}^{\text{max}} = M \cdot \left( 1 + \frac{m_{\text{min}}}{m_{\text{max}} - m_{\text{min}}} \right)$
2 <sup>nd</sup> Order Upwind	$\lambda_{\text{Re}}^{\text{max}} = M \cdot \left( \frac{11}{5} + 4 \frac{m_{\text{min}}}{m_{\text{max}} - m_{\text{min}}} \right)$	$\lambda_{\text{Im}}^{\text{max}} = M \cdot \left( 1 + \frac{11}{5} \frac{m_{\text{min}}}{m_{\text{max}} - m_{\text{min}}} \right)$
3 <sup>rd</sup> Order Upwind-biased	$\lambda_{\text{Re}}^{\text{max}} = M \cdot \left( \frac{1}{2} + \frac{7}{5} \frac{m_{\text{min}}}{m_{\text{max}} - m_{\text{min}}} \right)$	$\lambda_{\text{Im}}^{\text{max}} = M \cdot \left( 1 + \frac{7}{5} \frac{m_{\text{min}}}{m_{\text{max}} - m_{\text{min}}} \right)$

The  $\lambda$ 's refer to the maximum magnitude eigenvalues in the real and imaginary directions and  $M$  is the number of grid points.

Now that the eigenvalues have been characterized for the basic cases the positivity preservation can be considered. The method developed in Chapter 6.0 can be applied to the advective portion. Of course the use of the filter will change the eigenvalue spectrum. However, as was discussed earlier the filtered version results in an eigenvalue spectrum that is less than or equal to that of the unfiltered version so the above estimates are acceptable in all cases.

#### 9.4.2 Coagulation Portion

The coagulation portion of the general dynamics equation presents some new challenges. The numerical solution consists of discretizing the integrals in the size coordinate and then integrating through time as in the previous sections. The key difference between the earlier problems is of course the non-local nature of the integral equations.

Looking at the coagulation equation,

$$\frac{dn(m,t)}{dt} = \frac{1}{2} \int_{m_{\text{min}}}^{m-m_{\text{min}}} \beta_{m-m',m'} n(m-m',t) n(m',t) dm' - n(m,t) \int_{m_{\text{min}}}^{\infty} \beta_{m,m} n(m',t) dm'$$

it is clear that the value of  $n$  at each grid point will depend on the values over the entire range of the size variable. To handle the integrals a couple approaches can be employed. Any type of quadrature could be used to generate a sum that approximates the integral. There are many techniques that can obtain an arbitrarily high order of accuracy. However, known values of  $n$  are restricted to a grid as defined initially. Since the shape of the function being integrated changes at each time step anyway it is most efficacious to base the integration sum upon the standard grid points. This also makes the method far easier to combine with the condensation equation solution methods described above.

Using this simple method for the integration the second integral simply becomes the sum of the  $n$  values at each point in the vector multiplied by the grid step size, so for a given size  $m$  the integral becomes

$$\int_{m_{\min}}^{m_{\max}} \beta_{m,m} n(m', t) dm' \approx \sum_{i=1}^M h_i \beta_{j,i} w_i$$

where  $j$  corresponds to the size of interest,  $m$ , and  $i$  is the indexing variable over the size range.  $w$  is again the discretized version of  $n$ ,  $\beta$  is now a matrix of coagulation values at each size and  $h_i$  is the  $i^{\text{th}}$  spatial step size.

The first integral is a bit more involved as it depends on the product of two different  $n$  values. For a given  $m$  it is approximated as

$$\int_{m_{\min}}^{m-m_{\min}} \beta_{m-m',m'} n(m-m', t) n(m', t) dm' \approx \sum_{i=1}^{j-1} h_i \beta_{j-i,i} w_{j-i} w_i.$$

There are some important details as to how best to set up these sums and this is discussed in the following section.

The spectrum of eigenvalues is again a concern for the stability of the solution. The problem is a bit more challenging now as there is a non-linear equation involved. As in the linear cases it is the Jacobian matrix that is of interest, but in this case it must be calculated. First consider the full discretized version of the equation

$$\frac{\partial w_j}{\partial t} = \sum_{i=1}^{j-1} h_i \beta_{j-i,i} w_{j-i} w_i - w_j \sum_{i=1}^M h_i \beta_{j,i} w_i$$

For each grid point  $j$  the  $\beta$  and  $h$  values in the summations can be condensed into a matrix  $B^j$ . Then for every  $j$  we have

$$\frac{\partial w_j}{\partial t} = w^T B^j w$$

The details of the  $B$  matrices are discussed in the next section but for now knowing just the form suffices. From this the Jacobian can then be derived. It turns out that each row of the Jacobian is dependent on the sum of the  $B$  matrix and its transpose so that the full Jacobian is

$$Jac = \begin{pmatrix} w^T B^1 + (B^1)^T \\ \vdots \\ w^T B^M + (B^M)^T \end{pmatrix}$$

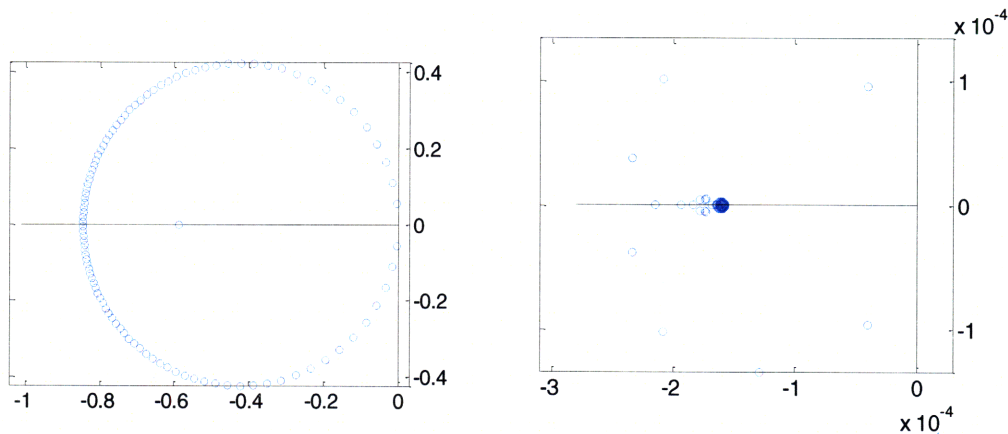
The key feature of this Jacobian is that it is dependent upon the value of  $w$  so it changes at every timestep even if the coefficients are constant in time. Unfortunately this can pose a challenge when attempting to efficiently characterize the eigenvalues.

Characterizing the eigenvalues now becomes very dependent upon the initial distribution. This means that some preliminary effort needs to be put in once a given initial distribution is selected. Unfortunately there are no general rules that can be applied as there were with the condensation equation for bounding the maximum eigenvalues.

As an example consider the coagulation equation with constant  $B = 0.01$  using a step size of  $h = 0.069$  and masses over the range  $[0.00001 \ 0.01]$ . Start with the exponential initial distribution discussed earlier,

$$q_0(m) = m \frac{N_0}{m_{ref}} e^{-m/m_{ref}}$$

As well as a square pulse function. Now consider the initial eigenvalue configuration for these two cases using the mass-based mass distribution,  $q(m,t)$ .



**Figure 9.4: Eigenvalues, Coagulation Eqn.; Exponential (left) and Square Pulse IC**

Clearly they are very dissimilar due to the dependence on the  $w$  vector. However, there is one key feature that gives hope for bounding the eigenvalues over the time integration: The magnitude of the  $w$  vector decreases over time. This occurs because the amount of particles at the smaller sizes decreases as they conglomerate to form larger particles. And the amount of larger particles increases by a smaller amount since there are less of them than the original smaller particles. Since  $B$  is positive this can only result in the eigenvalues of the Jacobian decreasing over time. Thus the initial bounding of the eigenvalues is sufficient for the time integration.

Though this may seem to be a significant issue, it is important to consider the eigenvalues of the entire general dynamics equation. For the advective portion they scale with  $1/h$

while for the coagulation portion they scale with  $h$ . Overall this means that the coagulation portion has a small effect on stability in the full GDE in most cases.

## 9.5 Implementation and Results

### 9.5.1 Condensation Examples

From the basic description of the discretization of the major portions of the GDE a practical implementation can be considered. The program was completed in MATLAB and is designed similarly to the routines discussed in Chapter 8.0.

The parameters of interest within the program are as follows:

**Table 9.5: Basic variables for MATLAB Population Balance function**

variable	Description
M	Number of grid points
x	size vector
B	coagulation coefficient (if constant)
H	growth function
tspan	interval in time
q	number of different species

There are two main points that differ from implementations mentioned earlier. The first is the grid spacing. Due to the nature of the evolution of the solution there are often more substantial changes in one region of the grid than in the rest. This can be handled more efficiently by using either variable grid spacing or by changing to another coordinate basis. Technically, using variable grid spacing reduces the order of the solution. The actual implications of this are discussed below.

The other important point is the manner in which the integral terms are handled. All of the values for the kernel at each grid are input as a matrix,  $B\_full$ . Recalling equation (9.18),

$$\frac{dn(m,t)}{dt} = \frac{1}{2} \int_{m_{min}}^{m-m_{min}} \beta_{m-m',m'} n(m-m',t)n(m',t)dm' - n(m,t) \int_{m_{min}}^{\infty} \beta_{m,m} n(m',t)dm'$$

the first integral is set up by assigning the values of  $B\_full$  to an array,  $B\_arr$ . The array consists of  $q \cdot M$  matrices and each matrix has the  $B\_full$  values assigned such that each element resulting from the evaluation of the first integral is assigned by

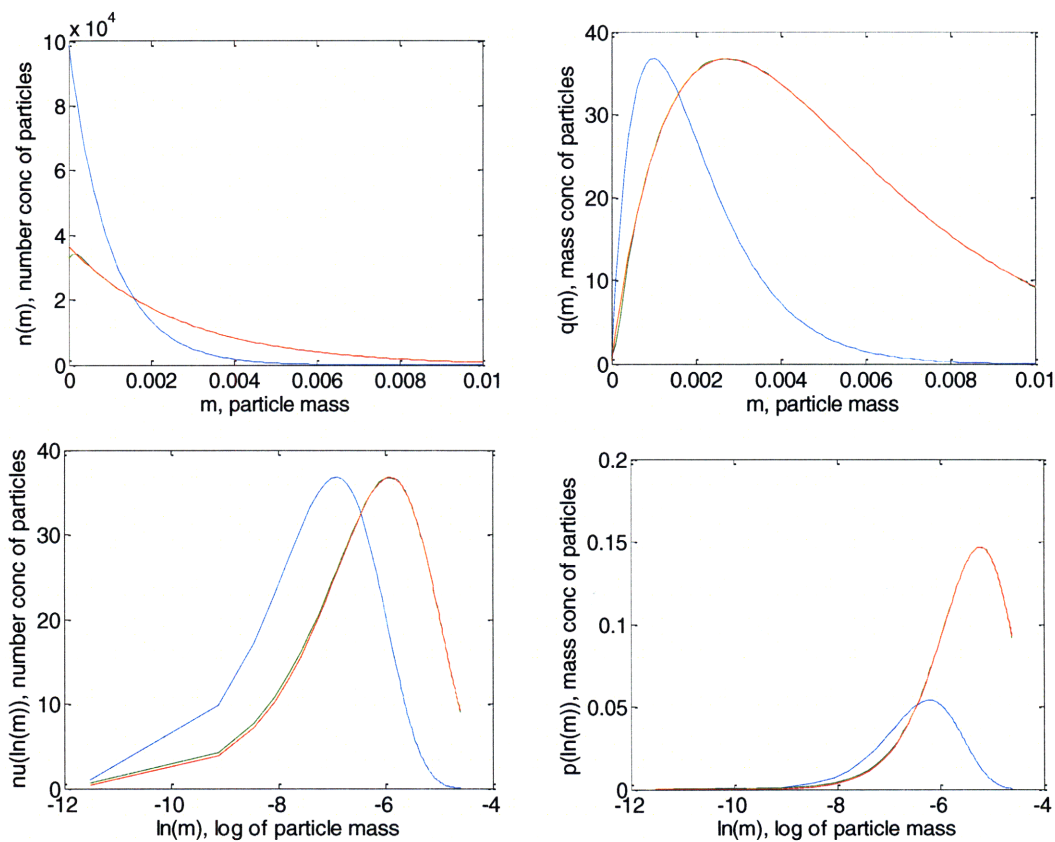
$$w\_out(i) = w^T \times B\_arr(:, :, i) \times w \cdot h$$

where the colons indicate all elements in that direction. The second integral is easier to evaluate; the values assigned to  $B\_mat$  just expand the values of  $B\_full$  over multiple points to incorporate the fact that there can be multiple components,  $q$ .

The positivity preserving filter is maintained in the same form as in Chapter 8.0. The only change is that the result now may be multiplied by the position vector,  $x$ , depending on the form of the equation (see Table 9.3).

For the solution the four different representations discussed above will be considered: number distribution based on mass,  $n_i(m,t)$ ; mass distribution based on mass,  $q_i(m,t)$ ; number distribution based on log mass,  $v_i(\mu,t)$ ; and mass distribution based on log mass,  $p_i(\mu,t)$ . For the initial condition, an exponential distribution is used. Despite its relative simplicity, it actually represents the conditions found in many real situations [Seinfeld & Pandis, 2006].

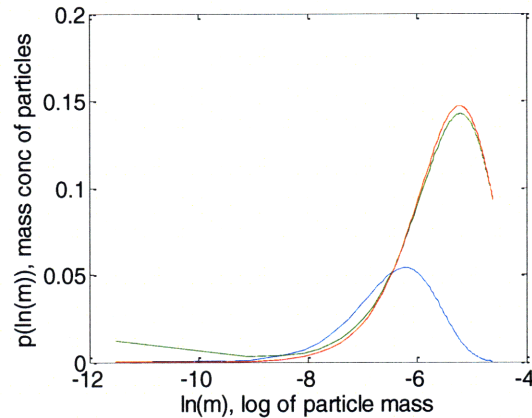
For the first case consider a simple one component non-coagulating system with a growth factor  $H$  of 0.1. The integration is over 10 time units using RKC with the positivity preserving filter. With 100 grid points the solutions for the four cases are as follows.



**Figure 9.5: Condensation equation example results**

The blue dashed line is the initial condition, the red dashed line is the analytical solution and the green solid line is the numerical solution.

In general we see that the agreement is very good. For a comparison, consider the plot of  $p_i(\mu, t)$  without the positivity preserving filter.



**Figure 9.6: Condensation equation, non-filtered version**

While positivity is still maintained, the error is much greater (see Section 2.2.5 for definitions):

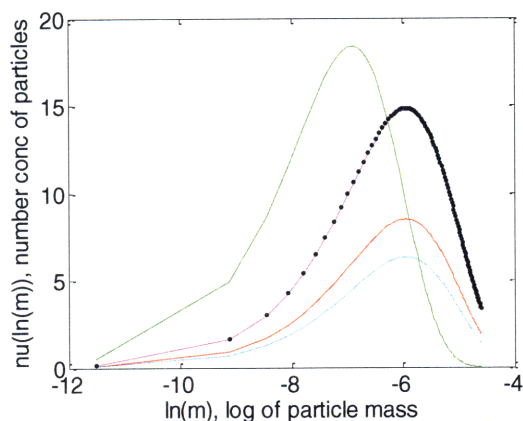
**Table 9.6: Condensation equation error comparison**

	$p_i(\mu, t)$ , no filter	$p_i(\mu, t)$ , filter
Absolute Error	0.0027	0.00018
Relative Error	0.00029	0.000024
Relative Peak Error	0.0314	0.0028

### 9.5.2 Coagulation Examples

Now let us consider a more complicated example. In this case there are two components, A and B, with growth factors of 0.1 and -0.05 respectively. Also let the coagulation coefficient have a constant value of 0.03. Consider the other parameters to be the same as above except now the time only goes to 1.





**Figure 9.7: Condensation and coagulation, 2 species system**

The green dashed line represents the initial conditions for both species. The red line is the solution for A and the cyan line for B. In this case an analytical result is still possible for the sum of the particle concentration. It is plotted with the purple dashed line. For comparison the sum of the concentrations of A and B is plotted with black dots.

Physically we see that even in this shorter time horizon the coagulation effect causes significant spreading of the distribution as particles now have another means for changing their size through interactions with other particles.

The error still remains relatively small, with Absolute Error of 0.0602, Relative Error of 0.000074 and Relative Peak Error of 0.0084.

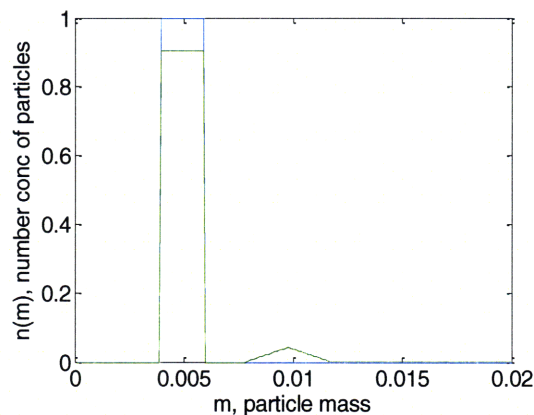
The solution efficiency performance is also quite favorable. In the previous few examples the problem was solved using modified RKC as well as the built-in ode45 and ode 15s functions. Only the  $n_i(m,t)$  and  $p_i(\mu,t)$  problem types are used as there is no substantial variation among the results.

**Table 9.7: Solution efficiency results from Condensation and Coagulation examples**

Method	Parameters	Prob Type	Fcn Evals	Time Steps	Soln Time (s)
ode45	H = 0.1 B = 0 t = 10	$n_i(m,t)$	292	41	14
ode15s			229	22	16
RKC			243	33	13
ode45		$p_i(\mu,t)$	292	41	15
ode15s			230	22	16
RKC			243	33	13
ode45	$H_A = 0.1$ $H_B = -0.05$ B = 0.03 t = 1	$n_i(m,t)$	67	11	39
ode15s			230	17	153
RKC			57	11	31
ode45		$p_i(\mu,t)$	67	11	41
ode15s			233	17	155
RKC			57	11	33

Predictably, the RKC method was superior in general. There is some stiffness, largely due to the condensation terms. The adaptive functionality of the RKC allowed for more efficient solutions than ode45. ode15s lagged behind substantially in the coagulation models because of the need to decompose and solve the dense non-linear matrix at each time step.

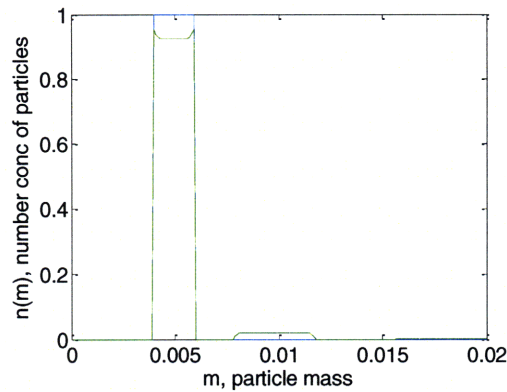
With the efficacy of the solution method on a coagulation equation demonstrated, let us consider a few more examples. First off consider a simple case to offer some intuition on the coagulation problem. For this example the  $n(m,t)$  distribution best illustrates the point. The initial condition is an equal number concentration of particles for the size range [0.004, 0.006]. If coagulation is the only physical process that occurs and the coagulation coefficient is constant across size ranges, consider what this means physically. Particles within the size would begin to collide with each other: some small particles would collide to make particles a bit larger than 0.008 and some large ones would collide to make particles a bit smaller than 0.012. But, combinatorially, the most frequently generated particles would be of size 0.010. The simulation bears this out as seen in the following figure. In this case the (uniform) coagulation coefficient has a value of 50 and the time is 1. The green solid line is the solution.



**Figure 9.8: Coagulation only, uniform**

Of course the particles would continue to combine and the distribution will eventually drift more upward as time passes and this is borne out in the simulation. However this example confirms our basic intuition.

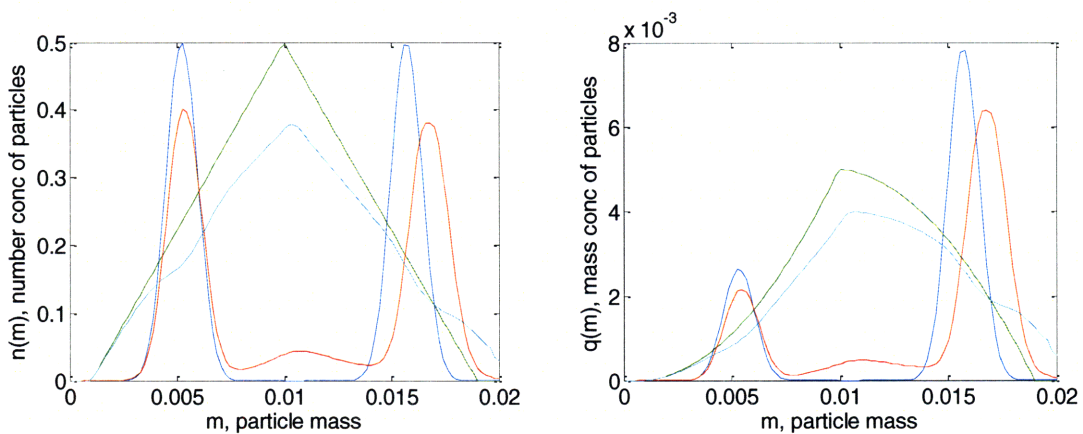
Now consider if we use a coagulation coefficient that differs for different particle sizes. Specifically consider a case where particles of similar sizes collide and coagulate at a high rate and particles of different sizes have decreasing coagulation coefficients. For a coagulation matrix  $B$  this would be represented by large values along the diagonal with decreasing values in a symmetric band about the diagonal. Note that symmetry is important for any physically realistic scenario. Otherwise a particle of size 1 coagulating with one of size 2 would be different than a particle of size 2 coagulating with one of size 1. Under the described conditions with the same situation as the previous example the results are as follows.

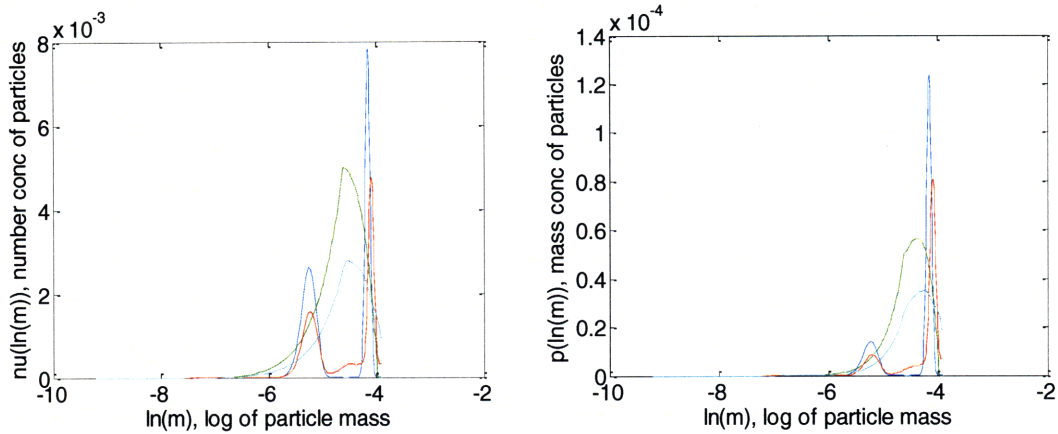


**Figure 9.9: Coagulation only, similar-size dominated**

Note that the particles in the middle of the initial distribution coagulated the most as would be expected since those on the “edge” of the distribution have fewer particles of similar size with which to coagulate. Otherwise the results are similar to the previous example.

With the performance and intuition considered it is worthwhile to consider a few variations on a more complicated example. This example commences with a bimodal distribution for one species and a triangularly peaked distribution for a second species, in terms of the  $n(m)$  distribution. The condensation coefficient is 0.03 for the first species and -0.01 for the second; each vary linearly with the grid position. The coagulation coefficients follow the “similar size dominated” pattern outlined above. The simulation is run over an interval of two time units. Overall this example is not designed to represent any actual physical condition. Indeed it is unlikely that the initial distribution and composition of particles would follow such a pattern. However, the point is mainly to show that the solution techniques can handle an arbitrary problem. The results are presented below for the four standard cases.





**Figure 9.10: and coagulation, 2 species system, variable parameters**

The blue dashed line is the initial concentration of species 1 and the green dashed line is that of species 2. The red line is the final distribution of species 1 and the cyan line is that of species 2.

In this situation no analytical solution is available for comparison. Exploring such problems is of course the reason for considering numerical solutions. However it is possible to consider the validity of the solution. The  $n(m)$  example is visually the most convenient for discussion. Most importantly we note that positivity is preserved. The sharpness of the peak degrades slightly but this is more due to the coagulation than any numerical diffusion. Otherwise the results make sense given the conditions of the example. The greater magnitude of the condensation (and evaporation) on the right side is obviated by the greater size-increasing shift of the distribution on the right side (or size-decreasing for the second component). The coagulation causes the decrease around the peaks noted above though the symmetry is lost in the other effects.

In terms of performance the results were similar to the first coagulation-condensation example. This demonstrates that even more complicated examples can be handled with efficiency.

Overall this chapter demonstrated the effectiveness of the RKC-based methods, the positivity-preserving filter, and the problem solving approach of this thesis in general. The basic examples considered here can be easily scaled to systems of more species and different types of coagulation relationships.

## 10.0 Other Examples

As mentioned in Section 3.2 there are countless systems that can be represented by PIDEs. Below are a couple of relatively simple examples that demonstrate the efficacy of problem-design solution approach of this thesis.

### 10.1 Neural Example

#### 10.1.1 Problem Background and Setup

Cells serve as an excellent example of systems which are affected by phenomena over large length scales. Neurons specifically represent complex systems where the long range interactions can be significantly different from the short range interactions captured by normal diffusion models. Murray [Murray, 1989] discusses several examples of neural interactions that can be described by partial integro-differential equations.

The firing of nerve cells is determined by both its autonomous rate as well as excitatory and inhibitory input from neighboring cells. The inducing cells can be some distance away and still exert an influence, hence the need to incorporate long-range diffusion. Practically this type of model is important in describing pattern formation. These systems range from mapping the regions of the visual cortex responsible to receiving input from each eye to the various patterns that form on shells. More recently they have been applied to memory cells, as discussed below.

Consider a system of cells that is only a function of position,  $x$ , and time,  $t$ . Denote the firing rate of cells as  $n(x,t)$ . For this simple model it is assumed that in the absence of external stimuli cells either do not fire or fire autonomously at a constant rate, normalized to one. This is also referred to as the synaptic drive or synaptic input in other situations, as described below. Perturbation from the steady state is described in terms of the rate of change of the firing rate,

$$\frac{dn}{dt} = f(n). \quad (10.1)$$

The functional form of  $f$  will determine where the steady states of the firing rate are. Perturbations due to external stimuli will cause temporary changes in the firing rate that may result in a new steady state. Consider a third order polynomial for  $f$  as shown in Figure 10.1. If the cell is temporarily excited to a firing rate above the meta-stable state value it will eventually settle at the upper steady state; if it is inhibited to below the meta-stable point it will eventually settle at the lower steady state. Thus we see the importance of external stimuli: even temporary effects of non-adjacent cells can change the long term behavior of a given cell.

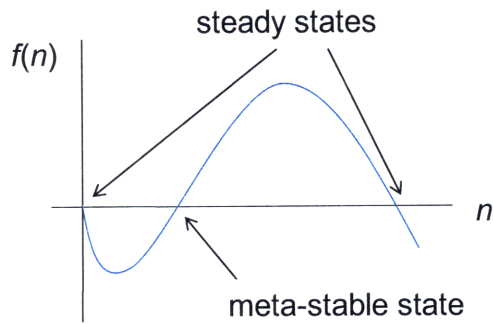


Figure 10.1: Rate of change of the firing rate

For a system of cells it is apparent how complicated the interactions could become. We need to consider how the neurons are affected over distances; this is accomplished by the following modification to (10.1):

$$\frac{\partial n}{\partial t} = f(n) + \int_S K(x-y) n(y,t) - 1 \, dy. \quad (10.2)$$

$S$  is the spatial domain over which the effects of other cells are relevant. The first term is the kernel which is a function of the distance between the particles of interest. The 1 in the integrand is an arbitrary constant that determines the value at which the influence from neighboring neurons is positive (in this case, if  $n > 1$ ) or negative. For the purposes of this example the kernel is assumed to be symmetric.

A typical cell behavior exhibits local activation and long range inhibition as shown by Figure 10.2 below. Consider the case with a steady state at  $n = 1$ . In the region around  $x$  the effect of the integral will increase the firing rate in the immediate region while inhibiting neighbors past a certain distance.

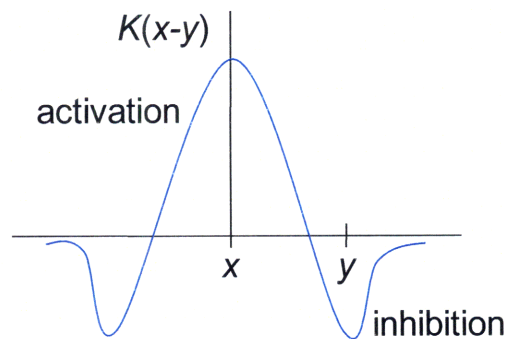


Figure 10.2: Symmetric activation kernel

When equations of the form of (10.2) are used in practice, relatively simple kernel functions are used. Additionally, small perturbations around the steady state are assumed so that  $f$  is approximately linear. With these simplifications the functional form of  $n$  can be assumed to be as

$$n(x,t) \propto \exp(\lambda t + ikx)$$

so that the Fourier transform can be used to get a solution in terms of the growth factor,  $\lambda$ .

While much can be done with these versions it is more interesting to consider the distribution of firing rates and expression that can result from more complex functions.

A problem discussed by Elvin [Elvin & Laing, 2005] deals with such a function. Their example applies to memory cells. Previous work cited in the paper describes how neurons in the prefrontal cortex have elevated firing rates during the period in which an animal is “remembering” the spatial location of an event.

Similar to above, we have  $n(x,t)$  which we will refer to as synaptic drive for the remainder of the example and a PIDE,

$$\frac{\partial n(x,t)}{\partial t} = d \frac{\partial^2 n(x,t)}{\partial x^2} - n(x,t) + \int_{\mathcal{S}} K(x-y) g(n(y,t)) dy. \quad (10.3)$$

where the most noticeable differences are the addition of a diffusional term and the more complicated function,  $g$ , in the integral. Note that when we have  $g(n(x,t)) > 0$  the neurons are active at  $x$ .

Problems in the form of (10.5) are especially interesting as diffusion equation-type problems benefit the most from the Runge-Kutta Chebyshev methods; indeed [Elvin & Laing, 2005] discuss the need for implicit solution methods for some of the larger values for  $d$ .

Solution of the non-diffusive ( $d=0$ ), stationary version have been completed in e.g. [Laing et al., 2002]. A key feature of the solution shape is a characteristic number of “bumps” in the solution profile which depends strongly upon the initial conditions. Such solutions are useful for examining the long-term behavior when examining solution accuracy. As such, it is worth briefly considering them.

Hereafter the kernel function,  $K(\cdot)$ , and the activation function,  $g(\cdot)$ , will be defined as

$$K(x) = e^{-b|x|} b \sin(|x|) + \cos(x)$$

$$g(n) = 2e^{-r/(n-\theta)^2} H(n-\theta)$$

Where  $b$ ,  $r$ , and  $\theta$  are parameters and  $H(\cdot)$  is the Heaviside function. These functions are depicted in Figure 10.3 below.

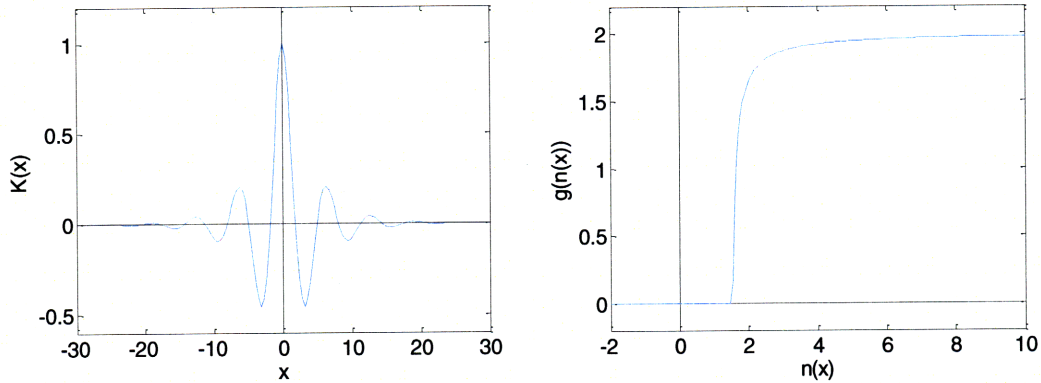


Figure 10.3: Kernel function (left) and activation function;  $b=0.25$ ,  $r=0.1$ ,  $\theta=1.5$

The initial condition is defined by the smooth, decaying function

$$n(x, 0) = \cos\left(\frac{Lx}{R}\right) \exp\left(-\left(\frac{Lx}{R}\right)^2\right). \quad (10.4)$$

where  $L$  and  $R$  are parameters.

As it turns out the steady-state solution can have an arbitrary number of bumps. The bumps refer to the number of central peaks in the solution profile. Mathematically, this can be defined by considering the stationary problem

$$n(x, t) = \int_S K(x-y) g(n(y, t)) dy + k$$

where  $k$  is some constant. Define the region of excitement by the range  $S' = \{x | n(x) > \theta\}$  which corresponds to some finite range where  $n$  is excited (above the threshold value  $\theta$ ). Due to the homogeneity of the stationary problem it is apparent that  $n(x)$  is a solution (that is, equal to zero) whenever  $n(x-a)$  is so the region to be considered can be  $S' = (a_1, a_2)$  where  $n(a_1) = n(a_2)$ .

If this region is connected then the solution is a one bump solution. A two bump solution could then be defined as

$$S' = \begin{cases} n > \theta & \text{on } (a_1, a_2) \cup (a_3, a_4) \\ n(a_1) = n(a_2) = n(a_3) = n(a_4) \\ n < \theta & \text{otherwise} \end{cases}$$

and this definition can be extended for  $N$ -bumps. This can be seen in the figures below.



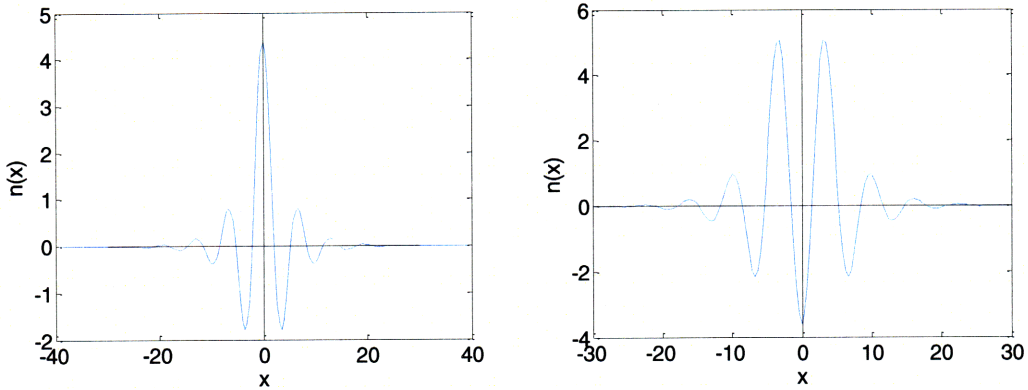


Figure 10.4: One and two bump solutions to the stationary problem

The zero point is of course shifted by the value of  $k$ , but the main point is that there are primary peaks and smaller peaks that die away below some threshold value.

An analytical solution can be found for the case with no diffusion and where  $r=0$  so that  $g(\cdot)$  reduces to a pure step function. In this case we just have

$$n(x,t) = \int_S K(x-y)H(n-\theta)dy$$

which reduces to

$$n(x,t) = \int_{S'} K(x-y)dy$$

where  $S'$  is the region of excitation. Thus we can just integrate the kernel over some specified excitation region. Defining this region as  $(-a_1, a_2)$ , performing the integration on the above-specified kernel function yields

$$n(x, \infty) = \begin{cases} \frac{e^{b(a_1-x)} - 2b \cos(a_1-x) - \sin(a_1-x) + b^2 \sin(a_1-x)}{b^2+1} & a_1 \leq x \\ \frac{e^{-b(a_1-x)} - 2b \cos(a_1-x) - \sin(a_1-x) + b^2 \sin(a_1-x) - 4b}{b^2+1} & x < a_1 \\ \frac{e^{b(a_2-x)} - 2b \cos(a_2-x) + \sin(a_2-x) - b^2 \sin(a_2-x)}{b^2+1} & a_2 \leq x \\ + \frac{e^{-b(a_2-x)} - (2b \cos(a_2-x) + \sin(a_2-x) - b^2 \sin(a_2-x)) - 4b}{b^2+1} & x < a_2 \end{cases}$$

This function can be evaluated over discontinuous regions  $S'$  to obtain the multiple-bump steady-state solutions. Much more detail on these solutions can be found in [Laing et al., 2002].

The boundary conditions are prescribed to be Diriclet conditions at the boundaries of the solution area,  $(-R,R)$ , and have a value of zero. This corresponds to the physical reality that signals do not travel an infinite distance.

### 10.1.2 Problem Solution

To solve the problem (10.5) with the functions and initial and boundary conditions described above a preliminary evaluation of the solution domain must be completed. The diffusive portion has the eigenvalue spectrum as discussed in Section 7.1.3 for the standard second order discretization. There is no advective portion. The integral portion shares characteristics with those of earlier sections. Specifically the eigenvalues are relatively small compared to the other sections. For the initial conditions specified above the eigenvalues have the following configuration:

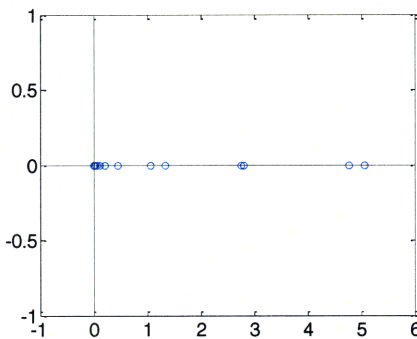


Figure 10.5: Eigenvalues of integral portion

Even when the diffusion coefficient is zero, these eigenvalues are not significant in their effect on the stability of the time integration.

Positivity preservation is not necessary in this example since the synaptic drive can be positive or negative. Regardless, the primary source of spurious oscillations, the advective portion, is not present in this problem so the need for filtration can be eliminated.

The solution methods undertaken in [Elvin & Laing, 2005] consisted of explicit Euler, implicit Euler, and Crank-Nicolson. These numerical experiments described in the paper can be replicated for the purposes of comparison. To evaluate accuracy the steady state solution for simple solutions can be compared with the time-integrated numerical approximation over long enough time periods.

For the first experiments, the simplest case with  $d$  and  $r$  equal to zero will be solved so that the result can be compared with the known analytical solution. For this and all following experiments  $b$  is set to 0.25 and the range is from  $-12.5\pi$  to  $12.5\pi$ .

The time span was set to 20 units; after that time there were no significant changes so this effectively approximates the long-time solution. The initial condition is set with  $L=6$ . The results are shown below.

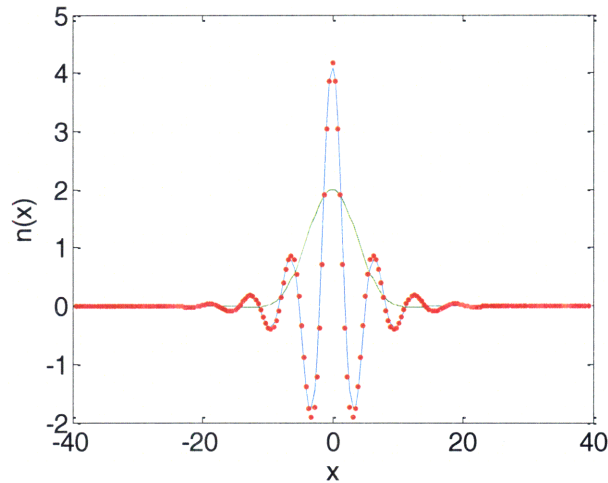


Figure 10.6: Long-time solution, 201 grid points,  $L=6$

The initial condition is the green dashed line and the analytical solution is the red dotted line. Qualitatively the results are quite close. The error measurements, as defined in Section 2.2.5, yield the following values.

Table 10.1: Error evaluation, neuron base case

Grid Points	Absolute Error	Relative Error	Peak Error
101	0.0319	0.0010	0.0754
201	0.0423	0.00068	0.0171

The errors for using the methods of the Elvin paper were effectively the same. That is not surprising given the fairly simple form of this first test. The results after 201 grid points produced negligible improvement in the accuracy.

It is important to note that changing the value of  $L$  in the initial conditions determines the final steady state, i.e. the number of bumps. A few different values of  $L$  will be tested following the results in the paper. It is worth mentioning here the sensitivity of the form of the solution to grid points and time steps. A deficiency in either of these is actually enough to cause a solution to converge to a different number of bumps. This gives a very good first level qualitative check even in the cases of the more complicated problems outlined below. With regard to this criterion the RKC- based solution performed well in all cases over 200 grid points.

Now the more complicated form will be tested. The parameter  $r$  is given a consistent value of 0.01 and  $d$  is varied to the values of the paper. Judging from the results of the simple version there is likely to be little difference in accuracy between the methods here and those of the paper. And since there is no longer an analytical solution available the

criteria of comparison will be performance. Specifically the number of time steps, the number of function evaluations, and the total solution time will be evaluated.

The results of the tests are displayed below. The RKC-based method is compared with the Explicit Euler and Crank-Nicolson methods used in the paper. The initial parameter  $L$  is tested at values of 6, 2.5, and 1.7 which results in one, two, and three bump solutions, respectively.

**Table 10.2: Solution efficiency results from neural example**

Method	L	d	Fcn Evals	Time Steps	Soln Time (s)
EE	6	0.1	50	51	15
CN			429	10	187
RKC			65	14	20
EE		0.5	200	201	47
CN			431	10	179
RKC			159	17	39
EE		1.0	400	401	99
CN			429	10	180
RKC			219	20	57
EE	2.5	0.1	50	51	15
CN			435	10	192
RKC			65	14	22
EE		0.5	200	201	47
CN			441	10	193
RKC			161	18	40
EE		1.0	400	401	99
CN			445	10	199
RKC			221	19	63
EE	1.7	0.1	50	51	15
CN			439	10	182
RKC			67	14	19
EE		0.5	200	201	47
CN			450	10	202
RKC			159	18	47
EE		1.0	400	401	99
CN			454	10	208
RKC			225	20	67

EE, CN, and RKC represent Explicit Euler, Crank-Nicolson, and Runge-Kutta Chebyshev (modified) methods, respectively. The number of grid points was 201,  $r$  was 0.10, and  $\theta$  was 1.5.

The advantages of the RKC-based method are immediately apparent for larger values of the diffusion coefficient. The Explicit Euler method eventually requires a large number of time steps to remain stable while the Crank-Nicolson method is inherently time-

consuming due to the of matrix decompositions required at each time step. Note that for RKC the number of function evaluations increased slightly with decreasing  $L$ . This is because as  $L$  decreases the number of bumps increases, and therefore solution complexity increases causing RKC's error correction to change the solution method slightly. As the two basic methods do not have error correction they showed identical performance for the different  $L$  values. In fact, the Crank-Nicolson method did not change significantly with  $d$  either since its stability domain is unbounded; the only differences in function evaluations were in the solving during the matrix decomposition.

It should be noted that neither of these methods have any sort of error correction, the number of time steps is determined either by stability (Explicit Euler) or arbitrarily (Crank-Nicolson). While the Explicit Euler shows some advantage for low diffusion it is in fact only a first-order method so accuracy is a concern. A second-order or higher method would have a similar form to that of the RKC (when  $d$  is relatively small) and similar performance.

For values of the diffusion coefficient greater than about 0.50 the long-time solution collapses to a flat line. However, it is really the intermediate dynamics that are of the most interest anyway. The actual physics of much of the system are not well known so the ability to handle a wide range of potential values is useful when using these types of models to compare with experimental data.

Overall the advantage of the RKC-based method is clear for these types of problems. The speed advantage is apparent in most situations and while the accuracy is not directly measurable for the more complicated problems the error correction methods in the RKC program is likely a significant factor. The above PIDE system is of course a very great simplification of reality. However it ultimately demonstrates the efficiency of the RKC-based approach. For use in more complicated models that better explain real systems this efficiency would become quite essential.

## **10.2 Radiative Heat Transfer**

### **10.2.1 Problem Background and Setup**

Heat transfer by electromagnetic waves, known as thermal radiation or radiative heat transfer is important in many applications and is a fundamental property of all materials. However, it was historically much less well understood than the other primary forms of heat transfer, conduction and convection. This is due in no small part to the less-intuitive underlying description of radiative heat transfer which depends on fourth-power temperature relationships and electromagnetic theory and leads to rather complicated descriptive equations. Additionally, the long range nature of radiation makes standard modeling techniques ineffective. Indeed, even relatively simple situations rely on partial integro-differential equations for models. With numerous practical uses and challenging equations, radiative heat transfer represents an active area of research.

The key feature of radiative heat transfer is that it does not need a medium for transmission; however the presence of a medium does have significant effects. These features necessitate several descriptive terms beyond temperature and some constant describing the medium. First off is the refractive index,  $n$ , of a medium which describes the relative speed of electromagnetic waves in a medium relative to a vacuum which is defined as  $n=1.0$ . The wave itself is identified by its frequency, wavelength, or wavenumber. In general many properties change with different frequencies; for the purposes of this section it is assumed that all dependent parameter have been integrated over all wavelengths.

The first energetic quantity of interest is the total emissive power,  $E$ , which has units of emitted energy / time / surface area. Emittance refers to radiative energy put off by a body. The standard reference point for a body is a blackbody, a theoretical object which has perfect absorption and therefore maximum emission. A subscript  $b$  refers to a blackbody hereafter. Using Planck's Law for a black surface bounded by a transparent medium and integrating over all wavelengths gives the total blackbody emissive power,

$$E_b(T) = n^2 \sigma T^4$$

where  $\sigma = \frac{2\pi^5 k^4}{15h^3 c_0^2}$  is the Stefan-Boltzmann constant,  $T$  is temperature,  $k$  is Boltzmann's constant,  $h$  is Planck's constant, and  $c_0$  is the speed of light in a vacuum.

The next important concept is solid angles. Solid angles are the two-dimensional analog of one-dimensional angles. Since radiation from a point on a surface can leave a surface in any direction and have different energies for each direction, it is useful to define the direction in terms of a polar angle,  $\theta$  (measured away from a normal vector to the surface) and an azimuthal angle,  $\psi$  (measured from an arbitrary axis on the surface) to give a unit direction vector,  $\underline{s}$ . With these two coordinates any point on a unit hemisphere above the point can be defined. For some other surface above the point, the projection of that surface onto a plane perpendicular to the direction vector divided by the square of the distance (denoted  $S$ ) between the new surface and the original point is called the solid angle,  $\Omega$ . The diagram on page 12 of [Modest, 2003] further elucidates the description. It should be noted that the maximum possible solid angle is  $2\pi$  as seen by integrating over all possible directions:

$$\int_{\psi=0}^{2\pi} \int_{\theta=0}^{\pi/2} \sin \theta d\theta d\psi = 2\pi$$

Solid angles are measured in dimensionless steradians, sr.

With solid angles defined, we can now consider the radiative intensity,  $I$ , which has units of radiative energy flow / time / area normal to rays / solid angle. The emissive power is the radiative intensity integrated over all possible directions pointing away from the

surface. For a blackbody the radiative intensity is independent of direction or diffuse so the relationship simplifies to

$$E_b(\underline{r}, T) = \pi I_b(\underline{r}, T).$$

where  $\underline{r}$  is a position vector describing the point of emission.

Now we have the tools in place to consider some problems involving media between surfaces that participate in the radiative heat transfer through emission, absorption, and scattering. Within a medium the concept of emissive power does not have any meaning since there is no surface to serve as a basis. Intensity, however, requires only a point and is therefore the quantity of choice.

First consider a non-participating medium and a radiative intensity traveling along some path  $\underline{s}$ . If the length scales are within the range of normal engineering problems, the speed of light is so fast relative to the length that all energy arrives “instantaneously” at each point along the path. This assumption will allow later calculations to be simplified.

Radiation is attenuated by absorption and scattering. Absorption depends on the properties and density of molecules present in a medium. It has been shown that absorption is proportional to the incident intensity and the distance of the beam. Therefore, over a path,  $\underline{s}$ , the effect on radiative intensity can be quantified with a linear coefficient,  $\kappa$ , by

$$dI_{abs} = -\kappa I ds. \quad (10.5)$$

The coefficient (and those defined subsequently) can actually have a dependence on the path. Scattering is a bit more complicated as the radiative energy now travels in another direction rather than just increasing the energy of the medium. However the equation for change along one direction has a similar form as for absorption, with a linear coefficient,  $\sigma_s$ ,

$$dI_{sca} = -\sigma_s I ds. \quad (10.6)$$

These two coefficients are often combined as the extinction coefficient,  $\beta$ . Any of the attenuation coefficients have the same form and the differential equation can be solved. Considering the extinction coefficient case we have

$$dI_{abs+sca} = -\beta I ds$$

$$I(s) = I(0) \exp - \int_0^s \beta ds = I(0) e^{-\beta \tau}$$

The radiative intensity at point  $s$  was found by integrating the extinction coefficient along the path. The result of this integral is known as the optical distance and is denoted  $\tau$ ; it is used as a proxy for distance in many of the later calculations.

Radiation is augmented by emission and scattering. Emission along the path is similar to absorption in that it is proportional to the path. However, now we are concerned with the local energy content in the medium. At thermodynamic equilibrium this can be shown to be just the blackbody intensity at all points in the medium, so the equation for the change in intensity due to emission is

$$dI_{em} = \kappa I_b ds. \quad (10.7)$$

where  $\kappa$  is as above. By itself this equation is fairly trivial but it is often combined with the absorption equation; the combined equation can be solved to give

$$\begin{aligned} dI_{abs+em} &= \kappa I_b - I ds \\ e^{\int_0^s \kappa ds'} \frac{dI}{ds} + e^{\int_0^s \kappa ds'} \kappa I &= e^{\int_0^s \kappa ds'} \kappa I_b \\ \int_{s=0, I=I(0)}^{s=s, I=I(s)} d \left( e^{\int_0^s \kappa ds'} I \right) &= I_b \int_0^s e^{\int_0^s \kappa ds'} \kappa ds \\ e^{\int_0^s \kappa ds'} I(s) - e^{\int_0^0 \kappa ds'} I(0) &= I_b \left( e^{\int_0^s \kappa ds'} - e^{\int_0^0 \kappa ds'} \right) \\ I(s) &= I(0)e^{-\tau} + I_b (1 - e^{-\tau}) \end{aligned}$$

where  $\tau$  is defined as above, with the assumption that  $\beta$  is zero.

Augmentative scattering is more complicated to describe since scattered radiative intensity from all directions must be considered. Consider the path of a cone defined by the unit vector  $\underline{s}$ . Now consider an infinitesimal segment of the line  $ds$  and a perpendicular area  $dA$  that together define a volume  $dV$ . Another cone defined by  $\underline{s}_i$  projects onto this area. Defining  $d\Omega_i$  as the solid angle of the incoming ray, the total radiative heat flux impinging on  $dA$  within this solid angle is

$$I(\underline{s}_i)(dA \underline{s}_i \cdot \underline{s}) d\Omega_i.$$

This flux travels through the volume  $dV$  for a distance  $ds/(\underline{s}_i \cdot \underline{s})$  (where the dot product is just the cosine of the angle between the two rays since the vectors are unit vectors). Now the total energy scattered away from  $\underline{s}_i$  is

$$- dI_{sca} = \sigma_s I ds = \sigma_s I(\underline{s}_i)(dA \underline{s}_i \cdot \underline{s}) d\Omega_i \left( \frac{ds}{\underline{s}_i \cdot \underline{s}} \right) = \sigma_s I(\underline{s}_i) dA d\Omega_i ds$$

Some fraction of this is scattered into the cone  $d\Omega$  defined by the original cone. This amount is defined by a scattering phase function,  $\Phi(\underline{s}_i, \underline{s})$  which defines the probability



that a ray from one direction will be scattered in another direction; it is conventionally divided by a normalizing factor of  $4\pi$ . The energy scattered from  $d\Omega_i$  to  $d\Omega$  is then

$$\sigma_s I(\underline{s}_i) dA d\Omega_i ds \frac{\Phi(\underline{s}_i, \underline{s})}{4\pi} d\Omega$$

Now the total energy flux scattered into the direction  $\underline{s}$  from all incoming directions is

$$dI(\underline{s})_{sca} dA d\Omega = \int_{4\pi} \sigma_s I(\underline{s}_i) dA d\Omega_i ds \frac{\Phi(\underline{s}_i, \underline{s})}{4\pi} d\Omega$$

or

$$dI(\underline{s})_{sca} = ds \frac{\sigma_s}{4\pi} \int_{4\pi} I(\underline{s}_i) \Phi(\underline{s}_i, \underline{s}) d\Omega_i . \quad (10.8)$$

where the integration over  $4\pi$  is to include all possible solid angles (the surface area of a unit sphere).

Now that all of the terms for attenuation and augmentation are defined the total energy balance on the cone defined by  $\underline{s}$  can be considered. Equations (10.5), (10.6), (10.7), and (10.8) can be combined but the differential terms on the left hand sides are ambiguous. Clearly the difference should encompass changes in both time and position. However, as was argued above, changes in time are generally insignificant on engineering time scales. Therefore, the quasi-steady state equation will be considered hereafter:

$$\frac{dI(s, \underline{s})}{ds} = \kappa I_b(s) - \kappa I(s, \underline{s}) - \sigma_s I(s, \underline{s}) + \frac{\sigma_s}{4\pi} \int_{4\pi} \sigma_s I(\underline{s}_i) \Phi(\underline{s}_i, \underline{s}) d\Omega_i$$

This equation is often re-written in terms of the optical distance defined above and a coefficient  $\omega$ ,

$$\omega = \frac{\sigma_s}{\kappa + \sigma_s}$$

To give the equation

$$\frac{dI}{d\tau} = -I + (1 - \omega)I_b + \frac{\omega}{4\pi} \int_{4\pi} I(\underline{s}_i) \Phi(\underline{s}_i, \underline{s}) d\Omega_i \quad (10.9)$$

Where the last two terms are combined as the source function:

$$S(\tau, \underline{s}) = (1 - \omega)I_b + \frac{\omega}{4\pi} \int_{4\pi} I(\underline{s}_i) \Phi(\underline{s}_i, \underline{s}) d\Omega_i \quad (10.10)$$

The general solution to (10.9) can be obviated by beginning from the form with the source function and applying an integrating factor:

$$\begin{aligned}\frac{dI}{d\tau} + I &= S \\ \frac{d}{d\tau} e^{\tau} I &= e^{\tau} S \\ I(\tau) &= I(0)e^{-\tau} + \int_0^{\tau} S e^{-(\tau-\tau')} d\tau'\end{aligned}$$

This form demonstrates how the entering intensity decays exponentially over the optical intensity. The integrand describes the self-extinction over the distance from a given point to the point of consideration; the integral sums all of these effects over the path. Clearly this equation can be quite difficult to solve completely; however there are a few more assumptions that can be useful before a solution is attempted.

First off, it is worth defining two more quantities that are of interest in engineering calculations. The incident radiation,  $G$ , is a useful quantity when the scattering is isotropic. It is defined as

$$G(\tau) = \int_{4\pi} I(\tau, \underline{s}_i) d\Omega_i \quad (10.11)$$

and describes the radiation impinging on a point from all sides. Another important quantity is the radiative heat flux vector,  $\underline{q}$ . In this context it is defined as

$$\underline{q}(\tau) = \int_{4\pi} I(\tau, \underline{s}) \underline{s} d\Omega \quad (10.12)$$

and describes the heat flux within the participating medium along the direction of  $\underline{s}$ .

From this point onward we will focus on a general example: a gray medium between two parallel plates. In this case we will consider radiative heat transfer in one dimension and on a system in radiative equilibrium. This relatively simple case has some examples that can be solved analytically to serve as a basis for comparison; in addition this case can be readily expanded to consider more complicated cases.

First off, a gray medium refers to one which has the same emittance and absorption characteristics across all wavelengths; this idealized condition is consistent with the above uses of total emissive power and radiative intensity which were implied to have been integrated over all wavelengths. Radiative equilibrium effectively implies that radiation is the only mode of heat transfer in the system.

To set up the system of interest, consider two parallel plates, each with some temperature profile. A point of interest,  $P$ , is somewhere within the medium; our goal is to describe the energy at this point.

Intensity leaving from a point on the lower plate can be described with a polar angle  $\theta$  measured from a vector normal to the surface ( $z$ -direction) and an azimuthal angle  $\psi$  measured from some axis in the plane of the surface ( $x$ - $y$  plane). Denote the radiative intensity leaving this point by  $I_w(\theta, \psi)$ . Now this radiation is augmented by the source term (Equation (10.10)) through emittance and scattering from other directions. This radiation also decays due to absorption and scattering exponentially according to the distance traveled.

The assumption is now made that both plates are isothermal and isotropic: temperature and radiation do not vary across the surface. This still allows for a dependence upon  $\theta$ . Now if the temperature and radiative properties of the medium vary only in the vertical direction every point with the same  $z$ -coordinate will have the same radiative intensity.

Thus if we define the optical depth as  $\tau = \int_0^z \beta dz'$  we can determine the radiative intensity in terms of only  $\tau$  and  $\theta$ . For use later we shall define the (more standard) optical depth along the path  $\underline{s}$  as  $\tau_s = \int_0^s \beta ds'$ .

These simplifications allow the source term to be expressed as

$$S(\tau, \theta) = (1 - \omega)I_b(\tau) + \frac{\omega}{4\pi} \int_0^{2\pi} \int_0^\pi I(\tau, \theta_i) \Phi(\theta, \psi, \theta_i, \psi_i) \sin \theta_i d\theta_i d\psi_i.$$

There are several simplifications to the scattering phase function,  $\Phi$ . For the first example presented below scattering will be ignored. This amounts to setting  $\omega$  to zero so that the source term simplifies to just the blackbody intensity  $I_b$ .

Let us now return to the radiative transfer equation (10.9). Recall that the standard definition for optical depth (along the path  $\underline{s}$ ) was denoted  $\tau_s$ ; this is related to the  $z$ -direction  $\tau$  by  $\tau_s = \tau / \cos \theta$ . Defining the radiative transfer equation in terms of this new  $\tau$  and also integrating over  $\psi$  (which is irrelevant due to the symmetry described earlier) gives

$$\begin{aligned} \frac{1}{\beta} \frac{dI}{ds} &= \frac{dI}{d\tau_s} = -I + (1 - \omega)I_b + \frac{\omega}{4\pi} \int_0^{2\pi} \int_0^\pi I(\tau, \theta_i) \Phi(\theta, \psi, \theta_i, \psi_i) \sin \theta_i d\theta_i d\psi_i \\ \cos \theta \frac{dI}{d\tau} &= -I + (1 - \omega)I_b + \frac{\omega}{2} \int_0^\pi I(\tau, \theta_i) \Phi(\theta, \theta_i) \sin \theta_i d\theta_i \end{aligned}$$

Now the general solution for intensity can be written as

$$I^+(\tau, \theta) = I_1(\theta)e^{-\tau/\cos\theta} + \int_0^\tau S(\tau', \theta)e^{-(\tau-\tau')/\cos\theta} \frac{d\tau'}{\cos\theta}$$

where the superscript + indicates that this expression is valid for radiation emanating from the lower wall ( $\tau=0, I_1$ ). From the top wall ( $\tau=L$ ) we have  $\tau_s = -(\tau_L - \tau) / \cos\theta$  where  $\cos\theta$  will of course be negative for  $\theta > \pi/2$ , corresponding to when the direction is opposite to that of the (upward pointing) normal vector. The intensity for the top wall (2) is then

$$\begin{aligned} I^-(\tau, \theta) &= I_2(\theta)e^{(\tau_L-\tau)/\cos\theta} + \int_{\tau_L}^\tau S(\tau', \theta)e^{(\tau'-\tau)/\cos\theta} \frac{d\tau'}{\cos\theta} \\ &= I_2(\theta)e^{(\tau_L-\tau)/\cos\theta} - \int_\tau^{\tau_L} S(\tau', \theta)e^{(\tau'-\tau)/\cos\theta} \frac{d\tau'}{\cos\theta} \end{aligned}$$

To make the notation more compact we can use  $\mu = \cos\theta$  in the previous three equations.

Now we are interested in calculating the radiative heat flux within the medium. Starting from equation (10.12) and recalling that the magnitude of the unit  $\underline{s}$  vector reduces to  $\cos\theta$  based on the conditions described above we have

$$\begin{aligned} \underline{q}(\tau) &= \int_{4\pi} I(\tau, \underline{s}) \underline{s} d\Omega \\ q(\tau) &= \int_0^{2\pi} \int_0^\pi I(\tau, \theta) \cos\theta \sin\theta d\theta d\psi \\ &= 2\pi \int_{-1}^1 I(\tau, \mu) \mu d\mu \\ &= 2\pi \left( \int_{-1}^0 I^-(\tau, \mu) \mu d\mu + \int_0^1 I^+(\tau, \mu) \mu d\mu \right) \\ &= 2\pi \left( \int_0^1 I^-(\tau, -\mu) \mu d\mu + \int_0^1 I^+(\tau, \mu) \mu d\mu \right) \\ &= 2\pi \left( \int_0^1 I_1(\mu) e^{-\tau/\mu} \mu d\mu - \int_0^1 I_2(-\mu) e^{-(\tau_L-\tau)/\mu} \mu d\mu \right) \\ &\quad + 2\pi \left( \int_0^\tau S(\tau', \mu) e^{-(\tau-\tau')/\mu} d\tau' - \int_\tau^{\tau_L} S(\tau', -\mu) e^{-(\tau'-\tau)/\mu} d\tau' \right) \end{aligned}$$

With flux defined we now have the tools to consider an example. Let us consider a non-scattering medium ( $\omega=0$ ) and black surrounding surfaces. The non-scattering simplifies the source equation to

$$S(\tau, \underline{s}) = I_b(\tau')$$

and the walls can now be described as  $I_{b1}$  and  $I_{b2}$ . Now the intensity in the upward and downward directions can be written, in terms of  $\mu$ , as

$$I^+(\tau, \mu) = I_{b1} e^{-\tau/\mu} + \frac{1}{\mu} \int_0^\tau I_b(\tau') e^{-(\tau-\tau')/\mu} d\tau'$$

$$I^-(\tau, \mu) = I_{b2} e^{(\tau_L-\tau)/\mu} - \frac{1}{\mu} \int_\tau^{\tau_L} I_b(\tau') e^{-(\tau'-\tau)/\mu} d\tau'$$

Now that the intensities and radiative sources are independent of direction these terms can be taken out of the integral and the order of integration can be reversed. This allows the heat flux to be written as

$$q(\tau) = 2\pi \left( I_{b1} \int_0^1 e^{-\tau/\mu} \mu d\mu - I_{b2} \int_0^1 e^{-(\tau_L-\tau)/\mu} \mu d\mu \right)$$

$$+ 2\pi \left( \int_0^\tau I_b(\tau') \int_0^1 e^{-(\tau-\tau')/\mu} d\mu d\tau' - \int_\tau^{\tau_L} I_b(\tau') \int_0^1 e^{-(\tau'-\tau)/\mu} d\mu d\tau' \right)$$

Now we note that in the integration over space  $\mu$  is essentially a dummy variable. The integrals over  $\mu$  can be written more compactly as exponential integrals and by changing the dummy variable by  $t = 1/\mu$ .

$$E_n(x) = \int_1^\infty e^{-xt} \frac{dt}{t^n} = \int_0^1 \mu^{n-2} e^{-x/\mu} d\mu$$

Finally, substitution into the heat flux equation gives

$$q(\tau) = 2\pi \left( I_{b1} E_3(\tau) - I_{b2} E_3(\tau_L - \tau) + \int_0^\tau I_b(\tau') E_2(\tau - \tau') d\tau' - \int_\tau^{\tau_L} I_b(\tau') E_2(\tau' - \tau) d\tau' \right)$$

And if the medium is gray we have  $I_b = n^2 \sigma T^4 / \pi$  so that the heat flux can be written in terms of temperature and simplified:

$$q(\tau) = 2n^2 \sigma \left( T_1^4 E_3(\tau) - T_2^4 E_3(\tau_L - \tau) + \int_0^{\tau_L} T^4(\tau') E_2(|\tau - \tau'|) \text{sign}(\tau - \tau') d\tau' \right)$$

In this reduced form we can return to considering transient cases. Consider the energy equation for the medium

$$\rho c_v \frac{\partial T}{\partial t} = -\frac{\partial q}{\partial z}$$

where  $\rho$  and  $c_v$  are the density and heat capacity of the medium, respectively. Differentiating the heat flux equation then gives us an equation describing the temperature profiles in the gray medium between two black parallel plates:

$$\begin{aligned} \rho c_v \frac{\partial T}{\partial t} &= -2n^2 \sigma \beta \frac{\partial}{\partial \tau} \left( T_1^4 E_3(\tau) - T_2^4 E_3(\tau_L - \tau) + \int_0^{\tau_L} T^4(\tau') E_2(|\tau - \tau'|) \text{sign}(\tau - \tau') d\tau' \right) \\ &= -2n^2 \sigma \beta \left( T_1^4 E_2(\tau) - T_2^4 E_2(\tau_L - \tau) \right. \\ &\quad \left. - 2n^2 \sigma \beta \frac{\partial}{\partial \tau} \left( \int_0^{\tau} T^4(\tau') E_2(\tau - \tau') d\tau' - \int_{\tau}^{\tau_L} T^4(\tau') E_2(\tau' - \tau) d\tau' \right) \right) \\ &= 2n^2 \sigma \beta \left( T_1^4 E_2(\tau) + T_2^4 E_2(\tau_L - \tau) \right) - 2n^2 \sigma \beta \\ &\quad \cdot \left( T^4(\tau) E_2(\tau - \tau) + \int_0^{\tau} \frac{\partial}{\partial \tau} T^4(\tau') E_2(\tau - \tau') d\tau' - \left( -T^4(\tau) E_2(\tau - \tau) + \int_{\tau}^{\tau_L} \frac{\partial}{\partial \tau} T^4(\tau') E_2(\tau' - \tau) d\tau' \right) \right) \\ &= 2n^2 \sigma \beta \left( T_1^4 E_2(\tau) + T_2^4 E_2(\tau_L - \tau) \right) \\ &\quad - 2n^2 \sigma \beta \left( T^4(\tau) + \int_0^{\tau} T^4(\tau') E_1(\tau - \tau') d\tau' - \left( -T^4(\tau) + \int_{\tau}^{\tau_L} T^4(\tau') E_1(\tau' - \tau) d\tau' \right) \right) \\ &= 2n^2 \sigma \beta \left( T_1^4 E_2(\tau) + T_2^4 E_2(\tau_L - \tau) + \int_0^{\tau} T^4(\tau') E_1(\tau - \tau') d\tau' + \int_{\tau}^{\tau_L} T^4(\tau') E_1(\tau' - \tau) d\tau' - 2T^4(\tau) \right) \\ &= 2n^2 \sigma \beta \left( T_1^4 E_2(\tau) + T_2^4 E_2(\tau_L - \tau) + \int_0^{\tau_L} T^4(\tau') E_1(|\tau - \tau'|) d\tau' - 2T^4(\tau) \right) \end{aligned}$$

This equation forms the basis for the next example. As is often done in the literature (e.g. [Prasad & Hering, 1969]), this equation can be non-dimensionalized as follows:

$$\frac{\partial \Theta}{\partial \bar{t}} = \frac{1}{2} \left( E_2(\bar{\tau}) + \Theta_2^4 E_3(\tau_L - \bar{\tau}) + \int_0^{\tau_L} \Theta^4(\tau', \bar{t}) E_1(|\tau - \tau'|) d\tau' - 2\Theta^4(\tau, \bar{t}) \right) \quad (10.13)$$

where the new variables are

$$\Theta = \frac{T}{T_1}, \quad \Theta_2 = \frac{T_2}{T_1}, \quad \bar{t} = \frac{4n^2 \beta \sigma T_1^3}{\rho c_v} t$$

The greatest challenge in solving this equation numerically is the fact that the function  $E_1(|\tau - \tau'|)$  has a singularity at the origin. This will be addressed in the next section.

### 10.2.2 Problem Solution

For the problem evaluation the non-dimensional radiative transfer equation (10.13) will be solved. The results in the literature lack sufficient detail to fully compare the solution methods. Despite this, some aspects of the solution techniques are still comparable.

The initial condition is that the entire medium and sides are at some uniform temperature,  $T_2$  after which the lower side goes through a step increase to a new temperature,  $T_1$ . The situation is depicted below.

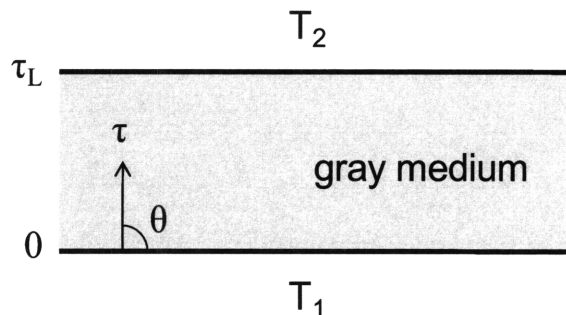


Figure 10.7: Radiative example

There is technically no need for boundary conditions since the only derivative is in time. In dimensionless form the initial conditions are then

$$\Theta(\tau, 0) = \Theta_2 \quad \Theta_1(0^+) = 1$$

The first step is to evaluate the eigenvalues of the problem. In this case it is particularly easy since there is neither an advective nor a diffusive term. Similarly to previous examples the integral terms here do not pose any significant challenges in terms of stability of the time integration. In this case, though, there is some challenge in terms of the integration due to the singularity.

The singularity will potentially be an issue at every grid point since the value  $|\tau_i - \tau'|$  will achieve a value of zero at some point along the grid. The most practical solution is to create a finer grid at the points near the singularity and also modify the function approximating the exponential integral to return a large but finite value for an evaluation at zero. With these modifications a standard trapezoidal integration scheme over a

modified grid can be employed. As will be shown later these approximations seem to have little effect on the overall accuracy of the solution.

There is no known analytical solution for any of the versions of the solution. Thus, to compare the accuracy of various methods, we solve the problem at a very a high level of accuracy (many grid points and time steps) and determine at what number of grid points the solution begins to diverge significantly from the accurate solution.

The descriptions of the solution methods in the various papers mentioned are not precise enough for exact solution. However, from a performance standpoint the time integration can be compared with the built-in MATLAB solvers, ode45 and ode 15s.

The numerical experiments are run over several optical depths and time ranges. To compare with the literature the final results are normalized according to the coordinates

$$\bar{\Theta} = \frac{\Theta^4 - \Theta_2^4}{1 - \Theta_2^4} \quad \bar{\tau} = \frac{\tau}{\tau_L}$$

so that the variables are comparable for any initial values and depths.

In all cases  $\Theta_2 = 0.5$  and the solution has 100 spatial grid points. The spatial integration at each time step uses a ten-fold increase in grid points in the range of  $\pm 5\%$  of the singularity. So for example if the range of  $\tau$  was from 0 to 10 and the singularity occurred at  $\tau=3$  then there would be 25 grid points from 0 to 2.5, 100 grid points from 2.5 to 3.5, and 65 grid points from 3.5 to 10. Since this variable grid is only used for the space integration it does not affect the stability in the manner mentioned in Section 2.2.1.

A few solution temperature distributions are presented below for various times and depths.



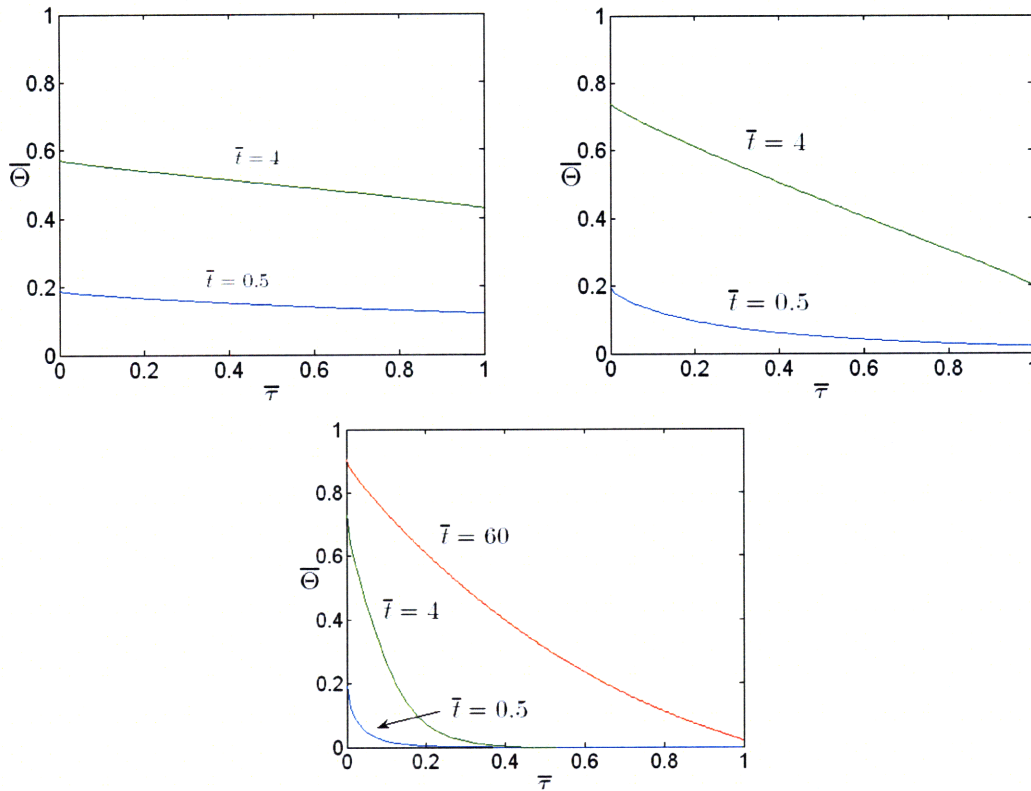


Figure 10.8: Temperature profile for  $\tau_L=0.1$  (left), and  $\tau_L=1.0$ , and  $\tau_L=10$  (bottom)

For these coordinates the initial condition is uniformly zero. After dimensionless time,  $\bar{\tau}$ , exceeds about 4 the solution doesn't change substantially for the first two examples. In the third example it takes closer to 60 for the long-term solution. Physically, we see that for case with optical thickness equal to 0.1 the temperature is nearly uniform throughout the surface and simply increases to an equilibrium value. As explained in [Prasad & Hering, 1969] this is approaching the optically thin limit where flux is nearly constant. As the optical thickness increases, the gradient becomes steeper. Eventually the optically thick limit is approached where only the short range interactions between gas molecules are present and the radiative transport resembles diffusion.

These results agree qualitatively (and quantitatively to the extent that values can be estimated from a graph) with those in the literature. Results for other parameters showed similar agreement.

The next evaluation is performance. Several test runs are compared among the modified Runge-Kutta Chebyshev, ode45, and ode15s. The parameters are as above.

**Table 10.3: Solution efficiency results from radiative heat transfer example**

Method	$\tau_L$	$\bar{t}$	Fcn Evals	Time Steps	Soln Time (s)
ode45	0.1	0.5	61	10	7
ode15s			115	11	35
RKC			55	11	7
ode45		4.0	67	11	8
ode15s			130	13	41
RKC			55	11	7
ode45	10	0.5	63	11	8
ode15s			115	11	35
RKC			57	11	8
ode45		4.0	69	11	9
ode15s			130	13	42
RKC			57	11	8
ode45		60	331	63	104
ode15s			461	28	176
RKC			207	45	86

The solutions obtained in each case were effectively identical which is not surprising due to the error correcting methods employed among the three methods. The results show that RKC has an advantage over ode45 and a large one over ode15s in terms of solution time. These results are what we would expect. Since there are no diffusive or advective portions these do not substantially affect the stability domain requirements. As mentioned above the integral portion has a comparatively small impact on the stability domain. However, there is clearly some impact so as to give RKC's adaptive approach an advantage over the similarly structured ode45. ode15s is noticeably slower due to the need to decompose and solve the dense, non-linear matrix due to the integral portion at each time step.

Overall this example has demonstrated another area in which the approach using problem characteristics in solution design is advantageous. Following the derivations and simplifications in the previous sections obviates the fact that problems involving radiative heat transfer can be quite complex. Despite its simplicity, this example demonstrates the importance of efficient solution techniques that would be necessary in attempting more substantial physical models.

## 11.0 Conclusions and Directions for Future Research

### 11.1 Conclusions

Modeling natural phenomena as systems is one of the primary aspects of chemical engineering and differential equations serve as the backbone for many of these models. Many characteristics of such models make them challenging to solve, such as the number of variables, non-local phenomena, the range of physical constants, and number of repeated solutions needed. All of these challenges represent areas of research that are quite active.

This thesis has developed several important techniques that can be applied to a wide range of systems of both partial differential and partial integro-differential equations. Numerical techniques for both the spatial discretization and time integration from several sources have been combined and enhanced for the design of solution methods that actively exploit problem knowledge. The resulting numerical methods achieve solutions that are accurate, robust, and preserve positivity in a computationally efficient manner.

The most important spatial discretization technique developed was the positivity preserving method. This technique allows for a consistency of order three while avoiding any spurious oscillations and minimizing artificial diffusion. And the design has allowed for the inclusion of systems with multiple dimensions and forms beyond the basic advection equation.

The time integration techniques are designed to exploit knowledge about the problem domain. Based on the Runge-Kutta Chebyshev design, they change the shape of the stability domain to accommodate the eigenvalues resulting from the spatial discretization of the right-hand side terms. The modification of the original design in this thesis allows for the inclusion of diffusive, advective, and integral terms. This allows for many classes of problems to be handled explicitly. In the case of extremely stiff problems, an implicit-explicit method was also developed to reap as many of the benefits of the explicit techniques as possible in problems where implicit techniques become necessary.

Population balance systems evaluated exemplified many of the challenges present in integro-differential equations. Multiple species, long-range interactions, and large characteristic domains are all present. Application of the solution techniques showed gains in both accuracy and efficiency over standard techniques.

The examples of both the neural and radiative heat transfer examples demonstrated the wide range of systems that can be solved with the solution tools developed in this thesis. Specifically, the neural example allowed for the efficient handling of an additional diffusional term that more accurately models the real physics. The radiative example demonstrated that a challenging integral can be handled in an efficient manner using explicit methods.

Overall the solution methods developed in this thesis have addressed the challenges mentioned above. Their development helps in the analysis of the increasingly complicated systems that underlie many interesting problems in chemical engineering and beyond.

### **11.2 Directions for Future Research**

While the tools developed in this thesis have been successful there are many pathways for further improvement of the methods and opportunities in the analysis of new and different systems.

At the basic level there is still some theoretical development for some of the novel techniques that can be developed. While their general efficacy has been well explored a deeper exploration of aspects such as convergence might lead to even better models.

The examples employed in this thesis have the potential for further expansion. For example, the neural problem is technically only valid over one layer of cells. It could be expanded to multiple layers since the solution method is efficient and relatively robust to more complicated systems. The radiative example could be expanded across multiple dimensions and also incorporate other modes of heat transfer as well.

One major opportunity is in parameter estimation. In such problems, solutions with similar structures are solved repeatedly and change only slightly at each update of the parameters. This is especially attractive for the methods of this thesis for two reasons. First, the similarity in problem structure could be exploited, for example, by examining the eigenvalue structure initially and using this to design a method that remains stable around small perturbations to this solution domain. Second, the need to solve the problem many times can benefit greatly from any improvements in efficiency of the time integration.

Another class of problems is those where the non-local effects extend over time. This is present in the Fisher-Kolmogorov equation for traveling wave solutions in diffusive systems and materials equations where a substance deforms with some “memory” of its previous form. The general form of such equations is

$$\frac{\partial b(x,t)}{\partial t} = \int_0^t f(s) \frac{\partial^2 b(x,s)}{\partial x^2} ds + g b(x,t) .$$

The need to include all past values for the quantity  $b(x,t)$  at each time step requires a re-design of the time integration technique. However it is clearly more advantageous to have an explicit type of method to facilitate this.

A different aspect to consider is the actual language and low-level implementation of the program. Throughout this thesis MATLAB was used for all of the programs but is limited in many aspects. For a more robust implementation in actual application

development a more efficient programming language should be used. This would allow one major advantage of explicit methods to be exploited through parallel processing. Very briefly, parallel processing could enable the evaluation of each line of the matrix to be split among processors at each time step. This is possible for explicit methods since the only dependency is upon the previous time steps, which allows all tasks at each time step to potentially be approached simultaneously.

Ultimately the work of this thesis serves as a solid base but there are many opportunities to be explored further from this point. It is the hope of the author that future investigators will be able to take the basic work in many new and exciting directions.

## **Appendix A: Capstone Paper**

### ***A.1 Executive Summary***

The methodology for the valuation of financial options has grown substantially since its beginnings with the work of Black, Scholes, and Merton. It remains an active area of research today as financial products become increasingly complex and the demand for determining prices quickly becomes ever more urgent.

One of the primary difficulties with the Black-Scholes model is that it assumes that prices move continuously. However as the data from financial markets consistently shows jumps of various sizes are the norm, not the exception. Recent work [Cont 2004; Cont & Voltchkova 2005] has demonstrated that models incorporating jumps into the underlying asset price movements give a superior representation of observed prices.

But the challenge still remains of solving the valuation equations efficiently. One configuration of the above models results in a partial integro-differential equation (PIDE). There are many similarities between that equation and those found in the main part of the thesis. Thus it is the work of this paper to apply the numerical methods to the jump process models for option valuation.

Results are compared for European options among several different numerical techniques. It is demonstrated that the Runge-Kutta Chebyshev-based approach of this paper offers significant advantages over the standard methods provided in the literature. Additionally the problems in which the underlying parameters vary with time can be solved as well.

## **A.2 Introduction**

In this thesis several techniques have been developed for solving systems of partial integro-differential equations. Thus far these systems of equations have been used to model physical phenomena. However, there are many other situations that can be described by equations of similar structure.

Financial derivatives are one such type of situation. Many sources have detailed the mechanics of these products as well as the math that describes them (e.g. [Baxter and Rennie 1996] and [Wilmott et al. 1995]); a brief overview is provided here. Preeminent among solution techniques for financial options is the Black-Scholes model. This analytical equation determines the value of an option on an observable underlying given several other parameters. While it is well established, it has many limitations. Indeed almost all cases of interest are based on modified versions of the equation that must be solved numerically.

One of the more interesting extensions is the jump process models. Simply put these model assume asset prices move in discrete jumps rather than with simple continuous diffusion. Obviously this captures reality more accurately and easily allows for the possibility of rare but significant price movements to be incorporated. Naturally these models are more difficult to solve.

As such there is always an interest in better methods for solving these problems. Accuracy is always important but in the fast-paced world of finance solution speed becomes extremely important. Overall this makes the solution techniques developed in this thesis potentially interesting.

This paper will cover a substantial background on the development of the mathematical framework, discuss the implementation in MATLAB, and provide results of several different scenarios.

## **A.3 Background**

The standard Black-Scholes option pricing model was explained and derived in Chapter 4.0. Section 4.3 gave a brief overview of jump processes. Here a more in depth description is given as a basis for the model developed in later sections. The development in this section primarily follows that of [Cont 2004].

First some of the basic theoretical concepts will be developed for the well-known methods. These concepts will then be used to demonstrate the Black Scholes result via a different approach than that taken in section 4.2. Next the mechanics of jump processes will be delved into more thoroughly. Finally all of the pieces will be integrated into the various jump models for option valuation. With these in place the numerical techniques for their solution will be developed.

The point of all these analyses is to determine the correct price of a given financial derivative. The premise of using expectations to calculate these values is attractive. Determining the expected value of some future payoff should give us the price we would pay for it. However we must be very careful with this idea.

Consider a stock with a log-normal distribution,

$$S_T = S_0 \exp(X)$$

where  $X$  is a normal random variable with mean  $\mu$  and standard deviation  $\sigma$ . Also consider a forward contract on this stock wherein one party agrees to purchase it for some strike price  $K$  at some future time  $T$ . Finally assume a risk-free interest rate of  $r$ . Normally forward contracts are set up so that the expected value of the future exchange of assets is zero, or

$$\mathbb{E}\left[e^{-rT}(S_T - K)\right] = 0.$$

Considering the contract we must determine  $K = \mathbb{E} S_T$ , the expected value of the stock at time  $T$ . Considering the model for the stock a logical way to proceed is to estimate the expected value of the stock at time  $T$ . By definition this is

$$\mathbb{E} S_T(X) = \int_{-\infty}^{\infty} S_T(x) f(x) dx$$

where  $f(x)$  is the standard normal pdf. This evaluates to  $S_0 e^{\mu + \frac{1}{2}\sigma^2}$  which seems to be a good estimate for the forward price. However this is completely wrong.

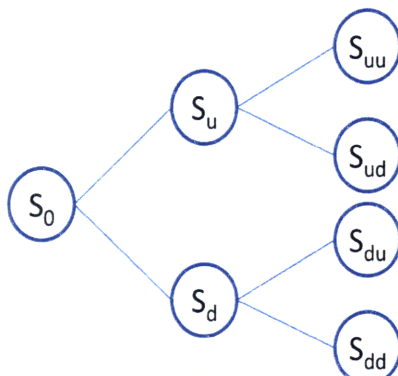
To see why consider a different strategy. Rather than estimating the value of the asset, let us attempt to replicate the contract. If we are selling (short) the contract, we can borrow the cost of the stock today,  $S_0$ , and purchase one unit of stock. This is held until time  $T$  and then exchanged for the amount  $K$ . The cost of this would be the cost of the loan at time  $T$ ,  $S_0 e^{rT}$ . Similarly, if we are buying (long) the contract we could short one unit of the stock and invest the proceeds, giving us  $S_0 e^{rT}$  at time  $T$ . Thus we see that the only possible value for  $K$  that both sides would agree on is  $S_0 e^{rT}$ . This price is the arbitrage price since any other price would allow one party to make a riskless profit.

So the strike price of  $S_0 e^{\mu + \frac{1}{2}\sigma^2}$  is incorrect by virtue of the fact that it would allow another party to enter into the contract and then replicate other side of the contract to fulfill its obligation. This party could make a riskless profit with certainty. To sum up the situation, arbitrage dominates over all methods of determining prices in financial markets. As was mentioned in [Baxter & Rennie, 1996], “if there is an arbitrage price, any other price is too dangerous to quote”.



So just using expectation blindly does not work but expectation does form an important part of the portfolio replication methods. To gain a better understanding of the expectation approach it is advantageous to consider first the discrete case.

Discrete models for asset prices are visualized as trees. Generally these are binary models where each node of a process branches into two different nodes at each time step. This is depicted below.



**Figure A.1: Discrete price tree**

The basic concept is well known, most notably as the Cox-Rubenstein option pricing model. However, we need to be more mathematically precise to put the concepts to use in more complicated models.

The possible asset prices and the relationships between them are collectively a process; this is denoted  $S$ . The set probabilities associated with each branch is the measure of the tree and is denoted  $\mathbb{P}$ . An important point to emphasize is that the process and measure are separate concepts and both are needed to describe the tree.

A filtration,  $\mathcal{F}_i$ , represents the history of asset prices up to time  $i$ . If the tree is non-recombinant then each node has its own filtration. At each time step there are many nodes so a particular  $\mathcal{F}_i$  can have several values. A claim,  $H$ , on the tree is a random variable which is a function of the history up to a given time point (i.e. the filtration). A claim could be, for example, the largest value the asset achieved on a path, the final value on the path, etc. Note that a claim is defined only at a particular time point while a process is defined for the entire tree.

Now we return to the expectation. In the context of the tree the expectation depends upon a specified measure and can be conditioned on a filtration. For example, the notation  $\mathbb{E}_{\mathbb{P}} H | \mathcal{F}_i$  indicates the expectation of the claim  $H$  using the probabilities defined by  $\mathbb{P}$  for the continuation of the paths that have initial segment  $\mathcal{F}_i$ . Effectively this is like finding the expectation as if we had started at a specific node. Also note that the expectation for a general  $i$  is in fact a process in  $i$ . This allows a claim to be converted into a process.

The next concept is a previsible process,  $\varphi$ . It is defined on the same tree as above but its value at any node is determined by one time step earlier,  $\mathcal{F}_{i-1}$ . An example of this could be a bond whose value at the next time step is determined by the interest rate set at the current time step. When considering a tree note that the size of the moves from node to node are previsible but exactly which direction is not.

The final definition at this stage is the martingale. Symbolically a process  $S$  is a martingale with respect to a measure  $\mathbb{P}$  and a filtration  $\mathcal{F}_i$  if

$$\mathbb{E}_{\mathbb{P}}[S_j | \mathcal{F}_i] = S_i \quad \forall i \leq j.$$

Conceptually, this expression means that the future expected value at time  $j$  of the process  $S$  under measure  $\mathbb{P}$  conditional on its history up to time  $i$  is the value of the process at time  $i$ . This is illustrated below where we see how two different probability measures affect the expected value.

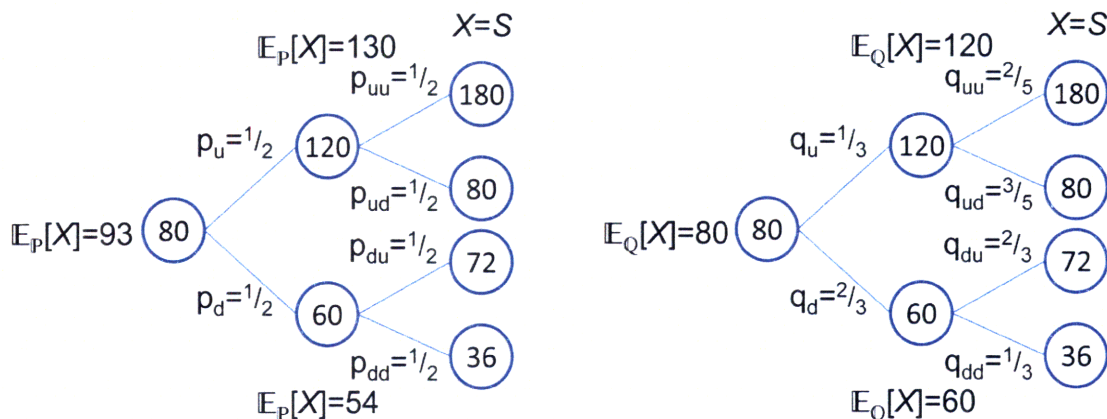


Figure A.2: Different probability measures, discrete case

In this case, the process is a martingale with respect to measure  $\mathbb{Q}$ . Another way to think about this idea is that the process has no drift upward or downward under the measure  $\mathbb{Q}$ .

From these definitions a useful tool that follows is the binomial representation theorem. Given a process  $S$  that is a  $\mathbb{Q}$ -martingale and another  $\mathbb{Q}$ -martingale  $N$ , there exists a previsible process  $\varphi$  such that

$$N_i = N_0 + \sum_{k=1}^i \varphi_k \Delta S_k$$

where  $\Delta S_i = S_i - S_{i-1}$  (note that this increment is itself previsible, though the direction a process takes is not).

Additionally the tower law,

$$\mathbb{E}_P \left[ \mathbb{E}_P \left[ X | \mathcal{F}_j \right] | \mathcal{F}_i \right] = \mathbb{E}_P \left[ X | \mathcal{F}_i \right] \quad i \leq j$$

allows us to confirm that for any claim  $H$ , the process  $\mathbb{E}_P \left[ H | \mathcal{F}_i \right]$  is always a P-martingale.

Now let us attempt to replicate a general option. Define a previsible, positive process  $B_i$  as a riskless bond and set  $B_0 = 1$ . The inverse process,  $B_i^{-1}$  is referred to as the discount process.  $Z_i \equiv B_i^{-1} S_i$  is the discounted stock process. And the discounted claim at time  $T$  is  $B_T^{-1} H$ . Now let us assume a Q such that  $Z_i$  is a Q-martingale and define the process  $E_i = \mathbb{E}_Q[B_T^{-1} H | \mathcal{F}_i]$  which is itself a Q-martingale. From the binomial representation theorem there exists a previsible process  $\varphi$  such that

$$E_i = E_0 + \sum_{k=1}^i \varphi_k \Delta Z_k \quad (A.1)$$

Now we construct a portfolio at time step  $i$ ,  $\Pi_i$ , which consists of  $\varphi_{i+1}$  units of the stock  $S$ , and  $\psi_{i+1} = (E_i - \varphi_{i+1} B_i^{-1} S_i)$  units of the bond.

Let us check if this makes sense. At time zero  $\Pi_0$  is worth

$$\varphi_1 S_0 + \psi_1 B_0 = E_0 = \mathbb{E}_Q \left[ B_T^{-1} H \right],$$

the cost of creation. At time 1 the original portfolio is now worth

$$\varphi_1 S_1 + \psi_1 B_1 = B_1 \left( E_0 + \varphi_1 \left( B_1^{-1} S_1 - B_0^{-1} S_0 \right) \right)$$

Since  $B_i^{-1} S_i - B_{i-1}^{-1} S_{i-1} = \Delta Z_i$  the binomial equation representation from (A.1) simplifies the expression to give  $B_i E_i$ . At time 1 a new portfolio,  $\Pi_1$ , must be purchased to maintain the replication. But this new portfolio costs exactly  $B_1 E_1$  to purchase regardless of which path  $S$  took (i.e. regardless of the value obtained by  $\mathcal{F}_1$ ).

The process is continued until the final value of portfolio  $\Pi_{T-1}$  is obtained, which is  $B_{T-1}^{-1} B_T H = H$ , as required. Thus the portfolio is perfectly replicated and remained self-financing.

So the general expression for the value of a given claim at any time point is

$$B_i \mathbb{E}_Q \left[ B_T^{-1} X | \mathcal{F}_i \right].$$

Now let us move through the continuous version. There are many mathematical subtleties in the move from the discrete version. But at a high level there are many analogs.

The primary change is that the binary process is replaced with Brownian motion in the form of a Wiener process. A Wiener process  $W_t$  is distributed under some measure  $\mathbb{P}$  as a normal random variable  $N(0,t)$ . The increment  $W_{s+t} - W_s$  is distributed as  $N(0,t)$ , under  $\mathbb{P}$  and is independent of the history up to  $s$ ,  $\mathcal{F}_s$ .

In subsequent sections it will be argued that Wiener processes are inadequate to fully describe the movements of financial assets. However, it is still instructive to develop some of the methods for working with such processes and gain more insight.

At any rate, the conventional model for stock prices is exponential Brownian motion

$$S_t = e^{\sigma W_t + \mu t}$$

where  $\sigma$  is some volatility and  $\mu$  is some drift. This is one type of stochastic process; stochastic processes have the general form

$$X_t = X_0 + \int_0^t \sigma_s dW_s + \int_0^t \mu_s ds$$

which can be written in differential form as

$$dX_t = \sigma_t dW_t + \mu_t dt$$

which is a stochastic differential equation (SDE).

Due to the nature of these processes one must resort to stochastic calculus to perform any sort of analysis. The key to stochastic analysis is Ito's lemma, discussed in chapter 4.0. It is restated here for convenience.

If  $X$  is a stochastic process, satisfying  $dX_t = \sigma_t dW_t + \mu_t dt$ , and  $f$  is a deterministic twice continuously differentiable function, then  $Y_t \equiv f(X_t)$  is also a stochastic process and is given by

$$dY_t = \sigma_t f'(X_t) dW_t + \mu_t f'(X_t) + \frac{1}{2} \sigma_t^2 f''(X_t) dt$$

Let us attempt to find the differential for the exponential Brownian motion equation. We can set  $X_t$  to be the process  $\sigma W_t + \mu t$  which is simple enough that the differential is obviously  $dX_t = \sigma_t dW_t + \mu_t dt$ . The desired  $S_t$  can then be written as  $f(X_t)$  if  $f$  is the exponential function,  $f(x) = e^x$ . Now Ito's lemma is applied to give

$$dS_t = \sigma f'(X_t) dW_t + \mu f'(X_t) + \frac{1}{2} \sigma^2 f''(X_t) dt$$

$$dS_t = S_t \sigma dW_t + (\mu + \frac{1}{2} \sigma^2) dt$$

Converting processes to SDE is relatively straight forward with Ito's lemma, but converting SDE's to processes (i.e. solving them) is where the challenge lies. This generally requires numerical methods which are covered in later sections. However, there are still a few simple systems that are solvable. It is instructive to go over them as it highlights several tools and methods that will be useful in subsequent sections.

We now return to the idea of measures. The Wiener process described above was contingent on a given measure. And based on the previous discussions it is clear that changing measures is useful for valuing claims on assets. But exactly how to change measures is not obvious. For this we will need the Radon-Nikodym derivative and the Cameron-Martin-Girsanov theorem.

A brief return to the discrete world will help explain these concepts. Recall the tree depicted earlier with the probabilities marked.

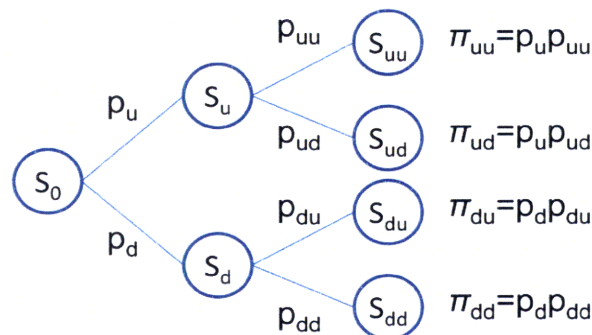


Figure A.3: Discrete tree with cumulative probabilities

The  $\pi_i$ 's above represent the cumulative probabilities of each path. Now if there were some different measure  $Q$  with cumulative probabilities denoted  $\rho_i$ , the ratios could be expressed at the end of each branch. This mapping,  $dQ/dP$ , is the Radon-Nikodym derivative of  $Q$  with respect to  $P$ . Note that this ratio is itself a random variable since it depends on the path taken.

For this derivative to exist at all points the two measures must be equivalent. That is, any event that is possible under  $P$  must be possible under  $Q$  and *vice versa*. With this last caveat we can see how knowing the Radon-Nikodym derivative and one of the measures is sufficient to completely specify the other.

The Radon-Nikodym derivative can be used to change between expectations under different measures. If  $x_i$  is each possible value of the claim  $H$ , we have

$$\mathbb{E}_Q H = \sum_i \rho_i x_i = \sum_i x_i \left( \frac{\rho_i}{\pi_i} x \right)_i = \mathbb{E}_Q \left[ \frac{dQ}{dP} H \right]$$

Additionally, we can turn the Radon-Nikodym derivative into a process,

$$\zeta_t = \mathbb{E}_P \left[ \frac{dQ}{dP} \middle| \mathcal{F}_t \right] \quad \forall t \leq T$$

which allows us to define the change in measure over a given filtration as

$$\mathbb{E}_Q [H_t | \mathcal{F}_s] = \zeta_s^{-1} \mathbb{E}_P [\zeta_t H_t | \mathcal{F}_s].$$

Now we can return to the continuous world. The most important change is that we now must use density functions to determine the probabilities of events. For example, an  $N(0,1)$  variable can be represented as

$$f_P(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \quad \mathbb{P}(X \in A) = \int_A \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx$$

where  $\mathbb{P}$  is the probability measure used and  $A$  is some event. The definition of the Radon-Nikodym derivative for continuous processes is as follows:

$$\frac{dQ}{dP}(\omega) = \lim_{n \rightarrow \infty} \frac{f_Q^n(x_1, \dots, x_n)}{f_P^n(x_1, \dots, x_n)}$$

where  $\omega$  is a given path and  $x_i$  is  $W_{it}(\omega)$ .

With this definition we can consider what a given distribution “looks like” under a different measure. To aid with this, first consider moment-generating functions. For a random variable  $X$  with distribution  $N(\mu, \sigma)$  we have, by definition,

$$\mathbb{E}_P \exp(\theta X) = \exp \left( \theta \mu + \frac{1}{2} \theta^2 \sigma^2 \right) \quad \forall \theta \in \mathfrak{R}.$$

Using the Radon-Nikodym derivative we can find what happens when the expectation is under  $Q$ . First, we make the *ansatz* that for Brownian motion we have the relationship between  $\mathbb{P}$  and  $Q$  given by

$$\frac{dQ}{dP} = \exp \left( -\gamma W_T - \frac{1}{2} \gamma^2 T \right)$$

Using the process  $W_T$  we have

$$\begin{aligned}\mathbb{E}_Q \exp(\theta W_T) &= \mathbb{E}_P \left[ \frac{dQ}{dP} \exp(\theta W_T) \right] = \mathbb{E}_P \left[ \exp \left[ -\gamma W_T - \frac{1}{2} \gamma^2 T + \theta W_T \right] \right] \\ &= \exp \left[ -\frac{1}{2} \gamma^2 \mu + \frac{1}{2} \theta - \gamma^2 T^2 \right] = \exp \left[ -\theta \gamma T + \frac{1}{2} \theta^2 T \right]\end{aligned}$$

because  $W_T$  is  $N(0, T)$  with respect to  $\mathbb{P}$ . Note that this is the moment-generating function of an  $N(\gamma T, T)$  variable. Thus we see that under  $\mathbb{Q}$  the variable is still normal and the variance remains  $T$ , but the mean has changed to  $-\gamma T$ . It can be formally proven that the only effect of changing the measure from  $\mathbb{P}$  to  $\mathbb{Q}$  using the above Radon-Nikodym derivative is to add a constant drift,  $-\gamma$ . For convenience we can define  $\mathbb{Q}$ -Brownian motion as  $\tilde{W}_t = W_t + \gamma t$ .

Putting all of these ideas together leads to the Cameron-Martin-Girsanov theorem, which states that if  $W_t$  is a  $\mathbb{P}$ -Brownian motion and  $\gamma_t$  is an  $\mathcal{F}$ -previsible process satisfying

$$\mathbb{E}_P \left[ \exp \left[ \frac{1}{2} \int_0^T \gamma_t^2 dt \right] \right] < \infty, \text{ then there exists a measure } \mathbb{Q} \text{ such that:}$$

$\mathbb{Q}$  is equivalent to  $\mathbb{P}$

$$\frac{dQ}{dP} = \exp \left( - \int_0^T \gamma_t dW_t - \frac{1}{2} \int_0^T \gamma_t^2 dt \right)$$

$$\tilde{W}_t = W_t + \int_0^t \gamma_s ds \text{ is a } \mathbb{Q}\text{-Brownian motion.}$$

The converse of this theorem is also true.

Now we move to the last major component in the continuous version: martingales. A stochastic process  $M_t$  is a martingale with respect to a measure  $\mathbb{P}$  if and only if

$$\begin{aligned}\mathbb{E}_P [M_t] &< \infty \quad \forall t \\ \mathbb{E}_P [M_t | \mathcal{F}_s] &= M_s \quad \forall s \leq t\end{aligned}$$

As in the discrete case, a martingale measure is one which makes the expected present value based on its past history exactly its current value. Put another way there is no drift expected to occur upwards or downwards.

With this definition we can now define the tower law and the martingale representation theorem for the continuous version. Specifically we have for the continuous random variable  $X$ ,

$$\mathbb{E}_P \left[ \mathbb{E}_P [X | \mathcal{F}_t | \mathcal{F}_s] \right] = \mathbb{E}_P [X | \mathcal{F}_s] \quad s \leq t$$

which allow us to state that for  $N_t$  to be a  $\mathbb{P}$ -martingale we need only  $\mathbb{E}_P [N_t | \mathcal{F}_s] = N_s$ .

And for the martingale representation theorem, given a process  $M_t$  that is a  $\mathbb{Q}$ -martingale and another  $\mathbb{Q}$ -martingale process  $N_t$ , there exists a previsible process  $\varphi$  such that

$$N_t = N_0 + \int_0^t \varphi_s dM_s.$$

Note that technically the Q-martingale's volatility must be positive with probability 1. The process  $\varphi$  just ends up being the ratio of the two Q-martingale's respective volatilities.

One other useful property to determine if a driftless process is a martingale is as follows. If  $dX_t = \sigma_t X_t dW_t$  for some  $\mathcal{F}$ -previsible process  $\sigma_t$  then

$$\mathbb{E} \left[ \exp \frac{1}{2} \int_0^T \sigma_s^2 ds \right] < \infty \quad \Rightarrow \quad X \text{ is a martingale} \quad (\text{A.2})$$

Now we can move to continuous portfolio representation and the Black-Scholes model. Consider the stock process  $S_t = S_0 \exp(\sigma W_t + \mu t)$  and the bond process  $B_t = \exp(rt)$  where  $r$  is the riskless interest rate,  $\sigma$  is the stock volatility, and  $\mu$  is the stock drift; all of these are constants.

The essential steps to portfolio replication can be used for any claim:

1. Find a measure Q under which the discounted stock process,  $Z_t$ , is a martingale.
2. Form the process  $E_t = \mathbb{E}_Q [B_t^{-1} H | \mathcal{F}_t]$ .
3. Find a previsible process  $\varphi_t$  such that  $dE_t = \varphi_t dZ_t$ .

Recall from the discrete version that using the discounted stock process allows us to remove the effect of cash growth. Inverting the bond process lets us define the discounted stock process as  $Z_t = B_t^{-1} S_t$  as well as the discounted claim  $B_t^{-1} H$ . Using the discounted stock price the SDE is

$$dZ_t = Z_t \left( \sigma dW_t + \left( \mu - r + \frac{1}{2} \sigma^2 \right) dt \right)$$

which can be readily verified with Ito's lemma.

Step 1: To make  $Z_t$  into a martingale we need the Cameron-Martin-Girsanov theorem. In the above SDE we want the drift term to be neutralized. This can be accomplished by defining a process  $\gamma_t$  which has a constant value

$$\gamma = \frac{\mu - r + \frac{1}{2} \sigma^2}{\sigma}.$$

The Cameron-Martin-Girsanov theorem then allows for a measure Q such that  $\tilde{W}_t = W_t + \gamma t$  is a Q-Brownian process. Now the SDE is



$$dZ_t = \sigma Z_t d\tilde{W}_t$$

This process appears to be driftless under  $\mathbb{Q}$ . Since  $\sigma$  is constant, by (A.2) this process is a martingale. Thus we have a  $\mathbb{Q}$ -martingale process and  $\mathbb{Q}$  must be the martingale measure for  $Z_t$ .

Step 2: Now we form  $E_t = \mathbb{E}_{\mathbb{Q}}[B_t^{-1}H | \mathcal{F}_t]$  which is itself a  $\mathbb{Q}$ -martingale.

Step 3: Next we want to construct  $E_t$  out of  $Z_t$  and the previsible process  $\varphi_t$ . From the martingale representation theory we have

$$\begin{aligned} E_t &= \mathbb{E}_{\mathbb{Q}}[B_t^{-1}X | \mathcal{F}_t] = E_0 + \int_0^t \varphi_s dZ_s = \mathbb{E}_{\mathbb{Q}}[B_t^{-1}X] + \int_0^t \varphi_s dZ_s \\ dE_t &= \varphi_t dZ_t \end{aligned}$$

as desired.

Now we have a replicating strategy: Hold  $\varphi_t$  units of stock at time  $t$  and hold  $\psi_t = E_t - \varphi_t Z_t$  units of the bond. To check if this is correct consider the value of the portfolio,

$$\begin{aligned} V_t &= \varphi_t S_t + \psi_t B_t = B_t E_t \\ d(B_t E_t) &= \frac{1}{2} d(B_t + E_t)^2 - B_t^2 - E_t^2 \\ &\text{using Ito's lemma...} \\ d(B_t E_t) &= (B_t + E_t)(dB_t + dE_t) + \frac{1}{2} \sigma_t^2 dt - B_t dB_t - E_t dE_t - \frac{1}{2} \sigma_t^2 dt \\ dV_t &= B_t dE_t + E_t dB_t \end{aligned}$$

Since we have  $dE_t = \varphi_t dZ_t$ , then  $dV_t = \varphi_t B_t dZ_t + E_t dB_t$ . Also, we have  $E_t = \psi_t + \varphi_t Z_t$ , so

$$\begin{aligned} dV_t &= \varphi_t B_t dZ_t + (\varphi_t Z_t + \psi_t) dB_t \\ &= \varphi_t (B_t dZ_t + Z_t dB_t) + \psi_t dB_t \end{aligned}$$

and since  $S_t = B_t Z_t$  and  $d(B_t Z_t) = B_t dZ_t + Z_t dB_t$ , as above, we have

$$dV_t = \varphi_t dS_t + \psi_t dB_t$$

so the portfolio  $(\varphi_t, \psi_t)$  is self-financing.

Now let us use this strategy to price a European call option with exercise date  $T$ . The payoff,  $H$ , is defined as  $\max(S_T - k, 0)$  or  $(S_T - k)^+$ . To find the value of this option at time zero we need

$$V_0 = e^{-rT} \mathbb{E}_Q \left[ (S_T - k)^+ \right]$$

where  $Q$  is the martingale measure for  $Z_t = B_t^{-1} S_t$ . Note that the discounted bond is constant and was moved out of the expectation; note also that this claim only depends upon the value of the stock at time  $T$ . So to solve this we only need the marginal distribution of  $S_T$  under  $Q$ .

To evaluate this we need to consider the process for  $S_t$  written in terms of the  $Q$ -Brownian motion  $\tilde{W}_t$  as defined above. Since  $\tilde{W}_t$  is defined for the discounted stock price and extra term appears when we apply it to the undiscounted  $S_t$ :

$$\begin{aligned} dS_t &= \sigma S_t dW_t + (\mu + \frac{1}{2} \sigma^2) S_t dt \quad \text{and} \quad d\tilde{W}_t = dW_t + \frac{(\mu - r + \frac{1}{2} \sigma^2)}{\sigma} dt \\ \Rightarrow dS_t &= \sigma S_t d\tilde{W}_t + r S_t dt \end{aligned}$$

Now we have the following, which can be confirmed by Ito's lemma

$$\begin{aligned} d \log(S_t) &= \sigma d\tilde{W}_t + (r - \frac{1}{2} \sigma^2) dt \\ \log(S_t) &= \log(S_0) + \sigma \tilde{W}_t + (r - \frac{1}{2} \sigma^2) t \\ S_t &= S_0 \exp \left( \sigma \tilde{W}_t + (r - \frac{1}{2} \sigma^2) t \right) \end{aligned}$$

Since  $\tilde{W}_t$  is distributed as an  $N(0,t)$  variable under  $Q$ , we can see that  $S_T$  has the distribution of  $S_0$  times the exponential of a normal variable with mean  $(r - \frac{1}{2} \sigma^2)T$  and variance  $\sigma^2 T$ . Defining  $G$  as a random variable with distribution  $N(-\frac{1}{2} \sigma^2 T, \sigma^2 T)$  we can write  $S_T$  as  $S_0 e^{G+rT}$ . If we write the claim out and apply the definition of the expectation operator we have

$$\begin{aligned} V_0(S_0, T) &= e^{-rT} \mathbb{E}_Q \left[ (S_0 e^{G+rT} - k)^+ \right] \\ &= \frac{1}{\sqrt{2\pi\sigma^2 T}} \int_{\log(k/S_0) - rT}^{\infty} S_0 e^x - k e^{-rT} \exp \left( -\frac{(x + \frac{1}{2} \sigma^2 T)^2}{2\sigma^2 T} \right) dx \end{aligned}$$

Now this is exactly the Black-Scholes expression from section 4.2. Defining  $\Phi(\cdot)$  as the cumulative normal distribution function, this expression can be written as

$$V_0(S_0, T) = S_0 \Phi \left( \frac{\log(S_0 / k) + (r + \sigma^2)T}{\sigma \sqrt{T}} \right) - k e^{-rT} \Phi \left( \frac{\log(S_0 / k) + (r - \sigma^2)T}{\sigma \sqrt{T}} \right)$$

which is of course the well-known Black-Scholes price of a European call option.

Thus we have seen how we can use expectations and Wiener processes explicitly to arrive at the Black-Scholes result. However as discussed in chapter 4.0 standard Black-Scholes models are inadequate for describing real financial markets. Thus we turn to jump processes. Fortunately the same framework developed above applies to different types of processes.

First off it is worthwhile to provide a brief argument for the use of jump process. The most obvious issue with the Black-Scholes model is the underlying assumption of normality of returns implied by the use of Wiener processes. As many years of data have shown and recent events have obviated, so-called “rare events” are relatively common: Asset prices can and do move by very large jumps in response to any number of factors. And as any financial practitioner knows the volatility of a given option is not constant over different asset prices, contrary to the assumptions of the Black-Scholes model. Indeed this leads to the well-known implied volatility and evocatively named graphical representations of this (smiles, smirks, etc.).

These issues alone do not condemn the model. It is possible to enhance the basic Black-Scholes model by allowing for volatility of volatility via non-linear diffusion coefficients which will result in distributions with “fat tails”. However, this still does not address the fundamental issue that asset prices move with discrete jumps.

On a more technical level Brownian motion is self-similar over scale, that is, as one decreases the time scale the process retains the same shape and is indistinguishable from another process if the axes were removed. Asset prices may be similar to Brownian motion over long time scales but as we observe the data on the daily or intraday scale one cannot ignore the discrete jumps in price. Since these time scales are often of interest it seems prudent to use a model that captures these features most easily and effectively.

When jump processes are employed most of the desired features of asset prices come out naturally. This means that the physical interpretation is richer and more intuitive which in turn makes the adjustment of parameters more natural.

The main challenge of jump-based models, as is generally the case, lies in their solution. Any models beyond the simplest Black-Scholes variants require some degree of numerical solution. But before we delve too deeply into that, let us discuss the development of the equations.

It is helpful to add to and expand upon some of the definitions of the previous section.

The probability measures  $\mathbb{P}$  and  $\mathbb{Q}$  defined above represent only one type of measures. In general measures can be defined for any type of set. A few types are important here. The Lebesgue measure for a set  $A$  in  $\mathbb{R}$  is

$$\lambda(A) = \int_A dx.$$

This corresponds to the concepts of volume, area, etc. The Dirac measure is defined as

$$\mu_X(A) = \# \{i, x_i \in A\} = \sum_{i \geq 1} \mathbb{1}_{x_i \in A}$$

where  $\#\{\cdot\}$  indicates the number of elements. Measures can operate on functions as well. A simple case of this is when  $\mu$  is the Lebesgue measure,  $\mu(f)$  is the Lebesgue integral of  $f$ .

$$\mu(f) = \int_{x \in A} f(x) \mu(dx).$$

We now come across the concept of probability space. This is written as  $(\Omega, \mathcal{F}, \mathbb{P})$ . The last two terms are as they were defined previously. The term  $\Omega$  represents the universe of all possible events and a random variable  $X(\omega)$  just maps specific events into some  $\mathbb{P}$ -measurable space. This definition is somewhat abstract in many cases but it is technically a necessary starting point for many theorems.

Another important concept is the cadlag function. This is an acronym for *continu à droite, limite à gauche* (right continuous with left limits). This is represented with the equations

$$f(t-) = \lim_{s \rightarrow t, s < t} f(s) \quad f(t+) = \lim_{s \rightarrow t, s > t} f(s)$$

and  $f(t) = f(t+)$

It is helpful to visualize this with the figure below.

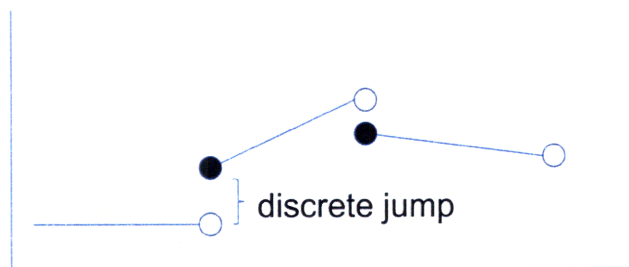


Figure A.4: Cadlag function

What this means in application to processes used later on is that any jumps in value are “unexpected”. If we are moving along in time from left to right, the value at the (random) time where the jump occurs cannot be extrapolated based on following the path immediately preceding it.

This leads to the concept of non-anticipating random times. Random times are just positive random variables that represent the time at which some event is going to take place. If based upon the filtration  $\mathcal{F}_t$  it can be determined whether or not the event has occurred the random time is called non-anticipating (or a stopping time). An example of

a stopping time is the exit time from an interval. If a process  $X$  starts at  $t = 0$  and  $a > 0$  the exit time from the interval  $(-\infty, a)$  is

$$T_a = \inf_{t > 0, X_t > a} .$$

An example of random time that is not a stopping time is the time when  $X$  reaches its maximum:

$$T_{\max} = \inf_{t \in [0, T], X_t = \sup_{s \in [0, T]} X_s} .$$

The global maximum of the entire process needs to be known before the time at which  $T_{\max}$  occurred is known.

We also need one more type of process. To construct the jump processes used herein Levy processes are used. The Wiener processes used in the above development represent one example of a Levy process. The other primary type is Poisson processes.

The first part to constructing Poisson processes are exponential random variables. They have the pdf

$$p(x) = \lambda e^{-\lambda x} \mathbf{1}_{x \geq 0} .$$

An important property of exponential random variables is that they are memoryless. That is, if  $T$  is a random time, the distribution of  $T - t$  with  $T > t$  is the same as knowing the distribution of  $T$  itself. Mathematically,

$$\mathbb{P}(T > t + s | T > t) = \frac{\int_{t+s}^{\infty} \lambda e^{-\lambda x} dx}{\int_t^{\infty} \lambda e^{-\lambda x} dx} = \mathbb{P}(T > s), \quad \forall t, s > 0$$

The distribution of the sum  $T_n = \tau_1 + \dots + \tau_n$ , where the  $\tau_i$ 's are iid exponential variables with parameter  $\lambda$ , has the pdf

$$p_n(x) = \lambda e^{-\lambda x} \frac{(\lambda x)^{n-1}}{(n-1)!} \mathbf{1}_{x \geq 0}$$

Closely related to the exponential distribution is the (discrete) Poisson distribution with pmf

$$\mathbb{P}(N = n) = e^{-\lambda} \frac{\lambda^n}{n!} .$$

For the series of iid exponential random variables with parameter  $\lambda$ ,  $(\tau_i)_{i>1}$ , the random variable

$$N_t = \inf \left\{ n \geq 1, \sum_{i=1}^n \tau_i > t \right\}$$

follows the Poisson distribution with parameter  $\lambda t$ .

Now the Poisson process can be formally defined: With  $(\tau_i)_{i>1}$  and  $T_n$  as defined above the process  $N_t$  is a Poisson process defined by

$$N_t = \sum_{n \geq 1} \mathbb{1}_{t \leq T_n}$$

Thus we see that the Poisson process is a counting process. It counts the number of random times,  $T_n$ , that occur between 0 and  $t$ .  $(T_n - T_{n-1})_{n \geq 1}$  is an iid sequence of exponential variables.

The Poisson process is not a martingale, unlike a Wiener process. However, the compensated Poisson process,

$$\tilde{N}_t = N_t - \lambda t$$

does have this property.

Now the tools are in place to begin constructing Levy processes. All Levy process can be represented as a combination of a Wiener process and a (possibly infinite) number of independent Poisson processes.

A compound Poisson process is defined as

$$X_t = \sum_{i=1}^{N_t} Y_i$$

where  $N_t$  is a Poisson process with parameter  $\lambda$  and the  $Y_i$ 's are iid random variables with pdf  $f(\cdot)$ . Such processes effectively provide the jumps in jump-diffusion models: the jumps arrive with intensity  $\lambda$  and the jumps size is drawn from the distribution of  $Y$ .

A couple useful tools when considering compound Levy processes are the jump measure and the Levy measure. The jump measure is associated with the number of jumps of some compound Poisson process  $X$ :

$$J_X(B) = \# (t, \Delta X_t) \in B, \quad B \subset \mathbb{R}^d \times [0, \infty)$$

so that for some set of jump sizes  $A$ ,  $J_X([t_1, t_2] \times A)$  counts the number of jumps over a given time interval. It can be shown that the intensity measure of  $J_X$  is  $\mu(dx \times dt) = \lambda f(dx)dt$ . The Levy measure is then defined as  $\nu(\cdot) = \lambda f(\cdot)$ . It measure determines the expected number of jumps over a certain time interval that belong to some size  $A$ . Formally,

$$\nu(A) = \mathbb{E} \left[ \# \{ t \in [0, 1] : \Delta X_t \neq 0, \Delta X_t \in A \} \right].$$

Note that the Levy measure is not a probability measure since it integrates to  $\lambda$ .

One issue that comes up is the fact that there is no restriction on  $\nu$  being finite so the jumps size could go to zero and the measure would still be valid. However, jumps of zero size would technically violate the cadlag property of Levy processes so some care must be taken in various definitions. This becomes important, for example, in the distinction between the two primary types of Levy processes used for financial models in this paper: jump-diffusion and infinite activity models.

The Levy-Ito decomposition provides a convenient representation of Levy processes. The basic form decomposes the process into a Brownian component and the compensated Poisson component. The Brownian component is characterized with a drift vector  $\gamma$  and a covariance matrix  $\Sigma$ . Then the entire process is represented as

$$\begin{aligned} X_t &= \gamma t + \Sigma_t W_t + X_t^I + \lim_{\epsilon \rightarrow 0} \tilde{X}_t^\epsilon \\ X_t^I &= \int_{|x| \geq 1, s \in [0, t]} x J_X(ds \times dx) \\ \tilde{X}_t^\epsilon &= \int_{\epsilon \leq |x| < 1, s \in [0, t]} x J_X(ds \times dx) - \nu(dx)ds = \int_{\epsilon \leq |x| < 1, s \in [0, t]} x \tilde{J}_X(ds \times dx) \end{aligned} \tag{A.3}$$

The last term is necessary to ensure that a singularity at zero for  $\nu$  does not lead to non-convergence. The shifted process  $\tilde{J}_X$  is a martingale for which convergence can be shown. The choice of one for the cut-off point is arbitrary but does not result in any loss of generality.

This then leads to the Levy-Khinchin representation:

$$\begin{aligned} \mathbb{E}[e^{izX_t}] &= e^{t\psi(z)} \\ \text{with } \psi(z) &= -\frac{1}{2} \Sigma z^2 + i\gamma z + \int_{-\infty}^{\infty} e^{izx} - 1 - izx \mathbf{1}_{|x| \leq 1} \nu(dx) \end{aligned} \tag{A.4}$$

Note that the function  $g(x) = \mathbb{1}_{|x| \leq 1}$  is the truncation function;  $\gamma$  does depend upon the choice of truncation function but the other parameters do not.  $\mathbb{1}_{|x| \leq 1}$  is the standard choice in the literature.

This representation applies to any Levy process that is defined by what is known as a characteristic triplet,  $(\Sigma, \nu, \gamma)$ . Now we see that with a characteristic triplet and either the Levy-Ito decomposition or the Levy-Khinchin representation we can specify and Levy process.

Also of great importance is the infinitesimal generator. It is defined as

$$Lf = \lim_{t \rightarrow 0} \frac{1}{t} \mathbb{E} [f(x + X_t) - f(x)] .$$

The reason why this is interesting becomes apparent when we see the infinitesimal generator of a Levy process with characteristic triplet  $(\Sigma, \nu, \gamma)$ :

$$Lf(x) = \sum_{j,k=1}^d \Sigma_{jk} \frac{\partial^2 f}{\partial x_j \partial x_k}(x) + \sum_{j=1}^d \gamma_j \frac{\partial f}{\partial x_j}(x) + \int_{\mathbb{R}^d} \left( f(x+x') - f(x) - \sum_{j=1}^d x'_j \frac{\partial f}{\partial x_j} \mathbb{1}_{|x'| \leq 1} \right) \nu(dx')$$

This partial integro-differential equation is similar in form to the other types described in the main body of this thesis.

When building Levy processes for financial models there are a few useful manipulations. First is the linear transformation. The Levy process  $X_t$  can be multiplied by a matrix  $M$  to get a new probability density. Another transformation is the tilting of the Levy measure. This consists of defining a new Levy measure which has the form  $\tilde{\nu} = e^{\theta x} \nu(dx)$  and results in a new characteristic triplet with this new measure. Next is subordination. We begin with a Brownian motion and subordinate it with a non-decreasing another independent, non-decreasing Levy process; this can be thought of as changing the underlying time to one that moves in discrete intervals.

Now we can at last begin to consider the Levy processes that are used in financial models. For the purposes of this paper the models considered can be divided into jump-diffusion and infinite activity models.

For jump-diffusion models the Levy process takes the form

$$X_t = \gamma t + \sigma W_t + \sum_{i=1}^{N_t} Y_i$$

These are very easy to interpret. The standard Brownian motion is supplemented with a compound Poisson process. The key parameter to specify is the distribution of jump sizes,  $\nu_0$ . In the model of [Merton 1976] the jumps are assumed to have a normal



distribution,  $Y_t \sim N(\mu, \sigma^2)$ . This actually allows for an analytically tractable model as the probability function of  $X_t$  can be represented with an infinite series:

$$V(S, t) = e^{-\lambda e^{\mu + \delta^2/2} T} \sum_{i=0}^{\infty} \frac{(\lambda_p T)^i}{i!} V_{BS}(S, t; i) \quad (\text{A.5})$$

where  $V_{BS}$  is the analytical Black Scholes solution for a call or put option; for the dependence on  $i$ , consider the put option example

$$\begin{aligned} V_{BS, put}(S, t; i) &= -S\Phi(-d_1) + Ke^{-rT}\Phi(-d_2) \\ d_1 &= \frac{1}{\varsigma_i \sqrt{t}} \log S/K + r + \frac{1}{2}\varsigma_i^2 t \\ d_2 &= \frac{1}{\varsigma_i \sqrt{t}} \log S/K - r + \frac{1}{2}\varsigma_i^2 t \\ \varsigma_i &= r + \frac{i}{T}(\mu + \frac{1}{2}\delta^2) - \lambda(e^{\mu + \delta^2/2} - 1) \end{aligned}$$

Of course much of the interest in this formulation of the probability lies in the flexibility to specify the distribution of jumps. And more interesting distributions generally require numerical methods for the solution. However, the Merton formulation serves as a convenient way to test for the accuracy of numerical models for simple cases.

Infinite activity models represent a larger family. As with jump-diffusion models the point is to specify the Levy measure. The basic idea that underlies these models is subordination, outlined above. Consider a  $C_t$  to be a subordinator and an independent Brownian motion  $W_t$  with drift  $\mu$ . Then via subordination we obtain the new Levy process  $X_t = \sigma W(C_t) + \mu C_t$ . One way to understand this concept is as information arrival being described by the subordinated process. The jump structure of processes that can be represented in the following form:

$$\nu(x) = \int_0^{\infty} e^{-\frac{(x-\mu t)^2}{2t}} \frac{\rho(dt)}{\sqrt{2\pi t}}$$

where  $\rho$  is the Levy measure of the subordinator.

The next consideration is the form of the subordinator. A common choice for the measure of the subordinator is of the form  $c/x^{\alpha+1}$  with  $0 \leq \alpha < 1$  and  $c$  some constant. This basic form can be exponentially tilted as well. If we have

$$\rho(x) = \frac{ce^{-\eta x}}{x^{\alpha+1}} \mathbb{1}_{x>0}$$

then  $c$  alters the jump intensity for all sizes, i.e. changes the scale,  $\eta$  determines the decay rate of large jumps and  $\alpha$  determines the relative importance of small jumps. If  $\alpha$  equals 0 or  $\frac{1}{2}$  the probability density can be determined analytically; thus these two forms are the most popular choices. For  $\alpha=0$  it is called the Gamma process and for  $\alpha=\frac{1}{2}$  it is called the Inverse Gaussian process.

Now that we have the measure for the subordinated process we can determine the measure for the Levy measure for the resulting process. Below  $\theta$  represents the drift and  $\sigma$  the volatility of the independent Brownian motion;  $1/\kappa$  is the variance of the subordinator at time 1. Using a conversion formula the result is obtained as a complicated expression involving Bessel functions. For the Gamma and Inverse Gaussian cases mentioned above the expressions for the Levy measure are

$$\nu(x) = \frac{\kappa}{|x|} e^{Ax - B|x|}$$

$$\text{with } A = \frac{\theta}{\sigma^2} \text{ and } B = \frac{\sqrt{\theta^2 + 2\sigma^2\kappa}}{\sigma^2}$$

for the Gamma case and

$$\nu(x) = \frac{C}{|x|} e^{Ax} K_1(B|x|)$$

$$\text{with } A \text{ and } B \text{ as above and } C = \frac{\sqrt{\theta^2 + \sigma^2\kappa}}{2\pi\sigma / \sqrt{\kappa}}$$

for the Inverse Gaussian case, with  $K_a(\cdot)$  representing a modified Bessel function of the second kind. The Gamma case will be used in the later sections so it is helpful to consider the shape of the distribution now. The main features are that the distribution goes to infinity at zero and the  $A$  and  $B$  parameters control the size of the tails. The potential for asymmetry is a very useful feature as we will see later. Note the truncation of the graphs needed due to the infinity at zero.

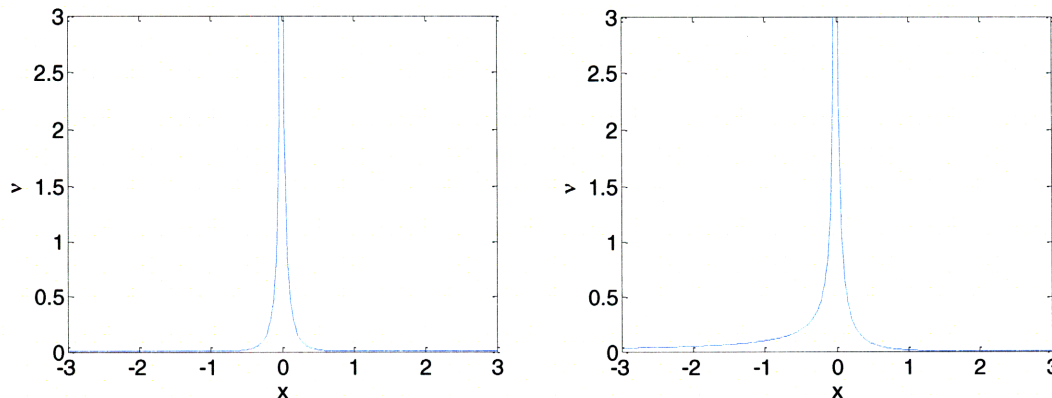


Figure A.5: Levy distribution for Gamma case

$\kappa = 0.1$ ,  $A = B = 5$  (left),  $\kappa = 0.1$   $A = 0.002$ ,  $B = 2$

A couple more variants exist. The tempered stable process multiplies the Levy measure of a stable process by a decreasing exponent. As the name implies, this can make large jumps less substantial. The form of the measure is similar to those above.

Generalized hyperbolic models start from specifying the probability density directly. These models are not closed under convolution generally result in the most complicated expressions for the measure but can be very useful when unusual pdf's are known.

All of the pieces are now in place to begin building actual financial models. As in the case when we had only Brownian motion the basic procedure consists of specifying the functional form of the asset price process,  $S_t$  (and a discounted version,  $Z_t$ ), finding a measure  $Q$  under which the discounted asset process is a martingale, forming the process  $E_t = \mathbb{E}_Q [B_t^{-1} X | \mathcal{F}_t^-]$  and then forming a portfolio with the previsible (bond) process such that  $dE_t = \varphi_t dZ_t$ .

The stock process has the form

$$S_T = S_0 \exp(rt + X_t) \quad (\text{A.6})$$

where  $X_t$  is a Levy process given by equation (A.3). Note that we have added the riskless drift  $rt$  to the exponent; this is done to give the process slightly more convenient notation later on. The change has no effect on the final result since there is already some arbitrary drift in the process. It merely results in a slightly different martingale process for the discounted price. The discounted process is then

$$Z_T = e^{-rT} S_T = S_0 \exp(X_T).$$

Now we need the measure  $Q$  under which this process is a martingale. As in the previous section, the conceptual notion that for the process to be a martingale the drift term must vanish is still valid. We want  $\mathbb{E}[e^{X_t}] = 1$  so, using the Levy-Khinchin representation (A.4), we see that we must have  $\mathbb{E}[e^{izX_t}] = e^{t\psi(z)}$  equal to one so we set  $z$  to  $-i$  and get

$$\frac{1}{2} \sigma^2 + \gamma + \int_{-\infty}^{\infty} e^x - 1 - x \mathbb{1}_{|x| \leq 1} \nu(dx) = 0$$

Thus the condition for the exponential Levy process to be a martingale is

$$\gamma = -\frac{1}{2} \sigma^2 - \int_{-\infty}^{\infty} e^x - 1 - x \mathbb{1}_{|x| \leq 1} \nu(dx).$$

Now if we consider both terms in the exponent and this restriction on  $\gamma$  we have an infinitesimal generator of

$$Lf(x) = \frac{1}{2}\sigma^2 \left( \frac{\partial^2 f}{\partial x^2} - \frac{\partial f}{\partial x} \right) + \int_{-\infty}^{\infty} \left( f(x+x') - f(x) - (e^{x'} - 1) \frac{\partial f}{\partial x}(x) \right) \nu(dx').$$

To put this equation to work, first recall the expression for option value that we want:

$$V_t = B_t \mathbb{E}_Q [B_T^{-1} H | \mathcal{F}_t].$$

For the rest of this work we will only consider payoffs that depend upon the terminal value,  $S_T$ . Since expression (A.6) already has the  $rt$  factored and we are using the bond process  $B_T = e^{r(T-t)}$  the expression we should use is

$$V_t(t, S) = \mathbb{E}_Q [e^{-r(T-t)} H_T | S_t = S].$$

To make this function more tractable we can make the change of variables suggested by [Cont & Voltchkova 2005]:

$$\begin{aligned} \tau &= T - t & x &= \ln(S / S_0) \\ h(x) &= H(S_0 e^x) & u(\tau, x) &= e^{r\tau} V(S_0 e^x, T - \tau) \end{aligned}$$

to give the desired expression

$$u(x, \tau) = \mathbb{E}_Q [h(x + r\tau + X_\tau)]$$

and thus

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}\sigma^2 \left( \frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial x} \right) + \int_{-\infty}^{\infty} \left( u(x+x', \tau) - u(x, \tau) - (e^{x'} - 1) \frac{\partial u}{\partial x}(x, \tau) \right) \nu(dx') + r \frac{\partial u}{\partial x}. \quad (\text{A.7})$$

Now we have a partial integro-differential equation that will allow us to determine option value.

#### **A.4 Implementation**

With the PIDE defined by (A.7) the similarity to other problems defined in the main body of this thesis becomes apparent. The discretization and solution procedures have many similarities but there are several new issues unique to this situation.

### A.4.1 Numerical Set-Up

The integration takes place over the entire real line. To make this problem tractable the domain is restricted to some interval,  $[A_L, A_U]$ . In addition the jumps need to be limited to some reasonable size (larger than the working interval),  $[B_L, B_U]$ . The measure  $\nu(x)$  can be defined by some given function.

Let us set up the numerical system. We set up a vector  $x^{full}$  of  $M_T$  points with grid spacing  $h$ . The “universe” is restricted to the interval  $[B_L, B_U]$  so the value of  $\lambda$  should be recalculated as

$$\hat{\lambda} = \sum_{i=M_{B_L}}^{M_{B_U}} \nu_j \approx \lambda = \int_{-\infty}^{\infty} \nu(dx)$$

where the  $M$ 's indicate the indices of the interval points. Similarly we can define  $\hat{\alpha}$

$$\hat{\alpha} = \sum_{i=M_{B_L}}^{M_{B_U}} (e^{x_j} - 1)\nu_j \approx \alpha = \int_{-\infty}^{\infty} (e^x - 1)\nu(dx)$$

as a convenient constant.

Now the system is

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}\sigma^2 \frac{\partial^2 u}{\partial x^2} - \frac{1}{2}\sigma^2 - r + \alpha \frac{\partial u}{\partial x} + \lambda u + \int_{B_L}^{B_U} u(x+x', \tau)\nu(dx').$$

The first two terms on the RHS are exactly the same as the advection diffusion problem with a diffusion coefficient of  $\frac{1}{2}\sigma^2$  and a velocity of  $(\frac{1}{2}\sigma^2 - r - \alpha)$ . The integral is less complicated than the coagulation term of the population balance systems but it does have the feature that it must be evaluated over the entire range of  $x$ .

In the infinite intensity case the value of the measure  $\nu$  can be infinite, *i.e.*  $\int_{-\infty}^{\infty} \nu(dx) = \infty$ .

This makes the above method impractical to apply in cases where certain regions (typically around  $x=0$ ) blow up. To handle this situation we approximate the process  $X$  with a finite activity process with a modified diffusion. We define a small number  $\epsilon$  that encompasses the region so that we have

$$\sigma_\epsilon^2 = \int_{-\epsilon}^{\epsilon} x^2 \nu(dx) < \infty$$

which is added to the variance defined for the finite intensity case. The condition for this new process to be a martingale requires modifying  $\gamma$  and allows us to define the following new constants:

$$\lambda_\epsilon = \int_{|x| \geq \epsilon} \nu(dx) \quad \alpha_\epsilon = \int_{|x| \geq \epsilon} (e^x - 1)\nu(dx)$$

and their corresponding numerical approximations. Thus the PIDE for this case is

$$\frac{\partial u}{\partial \tau} = \frac{1}{2} \sigma^2 + \sigma_\epsilon^2 \frac{\partial^2 u}{\partial x^2} - \frac{1}{2} \sigma^2 + \sigma_\epsilon^2 - r + \alpha_\epsilon \frac{\partial u}{\partial x} + \lambda_\epsilon u + \int_{B_L}^{B_U} u(x+x', \tau) \mathbb{1}_{|x'| \geq \epsilon} \nu(dx'). \quad (\text{A.8})$$

In terms of numerics, the derivatives with respect to  $x$  can be defined in the same manner as in sections 2.2.1 and 6.1. Note that we define the discrete approximation to  $u$  as  $w$ . For the second derivative we have

$$\frac{\partial^2 w_j}{\partial x^2} \approx \frac{1}{h^2} w(x_{j-1}, \tau) - 2w(x_j, \tau) + w(x_{j+1}, \tau)$$

where  $j$  is the index along the grid of the  $x$  vector.

For the first derivative we can use the positivity preserving methods described in chapter 6.0. Due to the nature of the problem positivity is not as great a concern as in the advection equation for the constant velocity case. However, when the volatility and interest rates are allowed to vary the situation becomes more interesting. In most cases the eigenvalues exhibit similar arrangements to those found in section 7.1.3.

The integral term is approximated as

$$w^{int} = Gw$$

where the  $G$  matrix contains a numerical approximations of the measure expression  $\nu$  shifted for each line.

At this point it is worth expanding upon the description of the solution space, the vector  $xfull$ , and the boundaries defined by the  $A$ 's and  $B$ 's since this aspect makes the integral approximation a bit different than those of earlier sections.

The portion of the vector  $xfull$  for which we actually report the solution is only over the span  $[S_L, S_U]$ , but the calculations are carried out over  $[A_L, A_U]$  to mitigate end effects and the span  $[B_L, B_U]$  is used for the integration to better approximate the fact that the integrals extend to infinity. The total length of  $xfull$  must actually be even greater than the span  $[B_L, B_U]$  since  $w(\tau, x+x')$  requires a shift due to the  $x$  term that is added. This is tacitly assuming that the distribution goes to zero for values greater outside of the cutoff points. This situation can best be observed in a sample of the  $G$  matrix below.

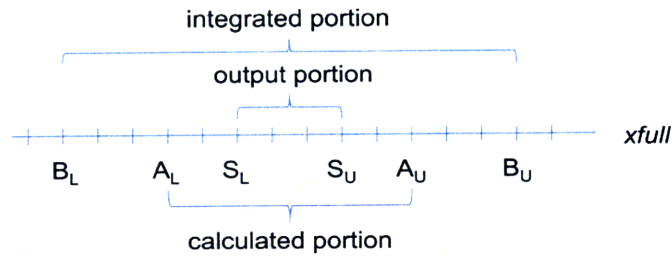


Figure A.6:  $x_{full}$  vector

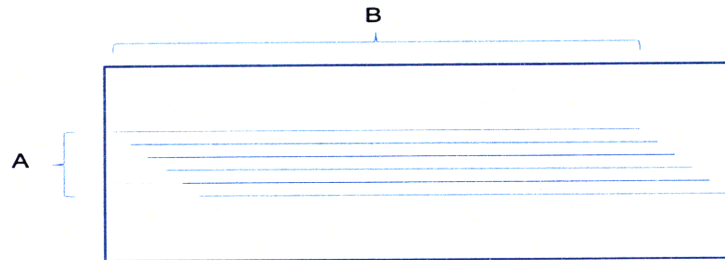


Figure A.7:  $G$  matrix

For a given line of the matrix multiplied by  $w$  we essentially want to have the sum

$$w_j^{int,1} = \sum_{i=N_{B_L}}^{N_{B_U}} w_{i+j-o} v_i.$$

The addition of  $j$  to the index in the sum represents the  $x$  that is added to  $x'$  and the  $o$  represents the offset that is required since the domain of  $x_{full}$  is greater than the integral range  $[B_L, B_U]$ . The need for this extra range is apparent when we observe figure Figure A.7. The vector representing the  $v$  portion of the integral must “shift” as it moves down the matrix so that the corresponding values for  $w$  can be multiplied. To accomplish this there must be some “extra” zeros at the end of that vector which can be thought of as the tail ends of the distribution for  $v$  which should go to zero as infinity is approached.

The part of the vector  $w$  (or function  $u$ ) that we care about for the final output of option values only spans  $[A_L, A_U]$  which is why the matrix  $G$  is only non-sparse in the middle portion (vertically). The other portions of the matrix are just the identity matrix. We only keep the other sections of  $w$  so that we get a better representation of the integral.

The next issue to consider is boundary conditions. As in many of the other examples in this thesis, the system is second order in space ( $x$ ) and first order in time ( $\tau$ ) indicating that we need two spatial boundary conditions and one initial condition. As was discussed in chapter 4.0, for option-pricing systems typical choices for the boundary conditions are as follows.

European Call Option		
	$V(S,T)$	$u(x,\tau)$
IC	$V(S,T) = (S - K)^+$	$u(x, 0) = (S_0 e^x - K)^+$
Diriclet BC	$V(0,t) = 0$	$u(1, \tau) = 0$
	$V(S_{max}, t) = S_{max}$	$u(x_{max}, \tau) = e^{r\tau} S_0 e^{x_{max}}$
Neumann BC	$\left. \frac{\partial V(S,t)}{\partial S} \right _{S=0} = 0$	$\left. \frac{\partial u(x,\tau)}{\partial x} \right _{x=1} = 0$
	$\left. \frac{\partial V(S,t)}{\partial S} \right _{S=S_{max}} = 1$	$\left. \frac{\partial u(x,\tau)}{\partial x} \right _{x=x_{max}} = e^{r\tau} S_0 e^x$

Here we give the conditions for a European call option; other cases can be derived similarly.

Application of the boundary conditions becomes a bit more involved than just setting the value at the terminal one or two points of the  $w$ -vector. Since we are only interested in values for  $u$  over the range  $[A_L, A_U]$  we need to assign values to the entire portion of  $w$  that is outside of this range. For a call option, for example, we see that the extreme portions of the option value as a function of underlying price remain the same shape regardless of what happens; they only translate horizontally by some amount that scales with the time  $\tau$ . As would be expected this scaling factor is just the velocity term  $(\frac{1}{2}\sigma^2 - r - \hat{\alpha})$  which has the same function as the velocity in the advection problem. Thus we apply this known solution to the values of the  $w$ -vector at each time step. This can be thought of as applying the Diriclet boundary conditions to the entire portion of the vector that is outside the solution region rather than just the two extreme points.

Finally we consider the eigenvalues of the various portions of the problem. Judging be the similarity of equation (A.7) to the advection diffusion equation it should not be surprising that the eigenvalues have similar arrangements to that case. Indeed the eigenvalues due to the second derivative portion have the characteristic pattern as the diffusion case: along the negative real line with a maximum value of  $-4(\frac{1}{2}\sigma^2)/h^2$ . The first derivative portion is handled via the positivity preserving filter as mentioned above. And as was the case for the advection problem, considering the upper bound on the spread of eigenvalues of the non-filtered case is sufficient to ensure stability. A couple of example eigenvalue configurations are displayed below for reference.



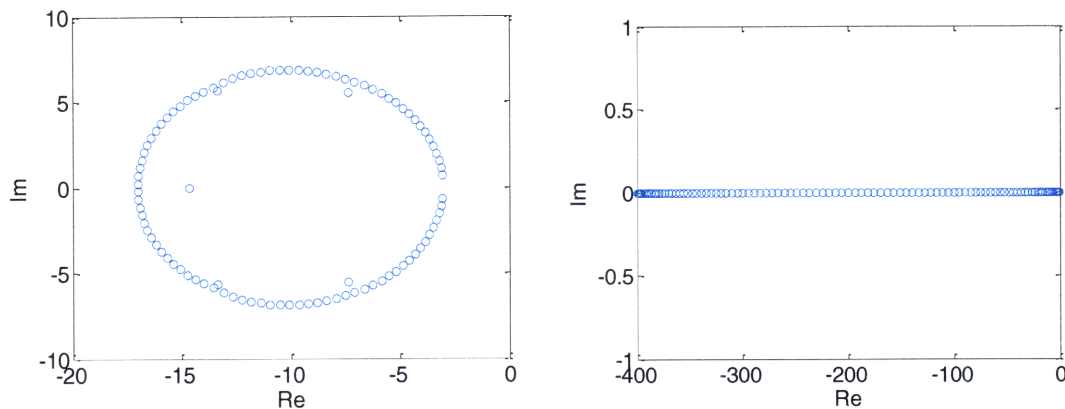


Figure A.8: Eigenvalues for the first (left) and second derivative components

The portion of the equation due to the linear dependence on  $\lambda$  simply has the eigenvalue of  $\lambda$  and poses no stability challenges. The integral portion is relatively straightforward in terms of eigenvalue analysis. The eigenvalues spread along the real axis and are smaller in magnitude than those due to the other terms. This behavior was also observed in the integral portions of PIDEs of other examples. Thus the main concerns for stability are the discretizations of the first and second derivative portions.

As was the case with several examples from the main thesis the various parameters ( $\sigma$ ,  $r$ , etc.) can change with time. This of course changes the scaling of the eigenvalue spread but this can be handled easily since the stability domain only needs to be scaled by a different amount; the eigenvalues need not be recalculated entirely.

#### A.4.2 Actual Program

Now everything is in place to discuss the actual MATLAB implementation. The main function is entitled `finOpt_jumpPIDE8.m`. In addition there is a program `probTempFinJump1.m` that contains all of the parameters that define the immutable parameters for different scenarios. The inputs for the function that can be specified by the user are as follows.

Table A.1: Inputs for `finOpt_jumpPIDE8.m`

Input	Description
$T$	time to expiration
$K$	strike price
$r$	risk-free interest rate, may be a vector
$\sigma$	variance, may be a vector
$M$	number of grid points

With the user inputs given the program converts the variables and parameters to their transformed counterparts. It then sets up the payoff conditions (which become the initial

conditions after the transformation of variables). The two main cases considered here are European call options and European put options.

The program then assembles the  $x_{full}$  vector. Overall the  $x_{full}$  vector consists of four regions. The central region  $[S_U, S_L]$  is the final reported solution; the region immediately surrounding (and including) that region  $[A_U, A_L]$  is where all of the sections of the PIDE are evaluated; the region immediately surrounding that region  $[B_U, B_L]$  is where the integral portion is evaluated (this comprises the tails of the distribution); and finally the outermost section is effectively a placeholder that allows the integral to be evaluated where it is centered at different points.

The region outside of  $[A_U, A_L]$  for the initial conditions is saved as a parameter which will be used as a boundary condition, shifted by the product of interest rate and time, as discussed in the boundary condition section above.

Everything is now in place to begin the time integration. The method employed in the paper [Cont & Voltchkova 2005] is the default choice. Their method uses a simple implicit-explicit splitting scheme. The integral portion is handled with an explicit Euler method and the remaining terms are handled with an implicit Euler method. This is the most basic IMEX scheme discussed in section 5.3. It is first order overall in convergence and so the accuracy is not very high. The next two time integration options are the built-in MATLAB functions `ode45` and `ode15s`. The last option is the modified two-step Runge-Kuta Chebyshev method developed in this thesis.

At each time step of the specified integration method, the function `odefcn` is evaluated. This function first calculates the various approximated parameters of the jump size distribution,  $\hat{\lambda}$  and  $\hat{\alpha}$ . It then assembles the matrix for the discretization of the second derivative of  $u$  and multiplies it by the coefficient for the current time step. Only the section of the matrix corresponding to the range  $[A_U, A_L]$  is populated with the approximations; the remaining portion is imply ones along the diagonal. The positivity-preserving filter is then applied based on the sign and magnitude of the velocity term in the manner described in section 6.1. The term  $\hat{\lambda}$  is multiplied an identity matrix to represent the  $\lambda u$  term, again over the  $[A_U, A_L]$  section of the matrix. For the integral terms an identity matrix of size  $M_T$  is once again assembled. Then for each point in the range  $[A_U, A_L]$  the vector  $nu$  (which has a range  $[B_U, B_L]$ ) is inserted beginning at the left most point of the matrix and shifted one point right for each subsequent row. Finally the boundary conditions are applied by inserting the initial condition shifted by the current discount factor into the updated  $w$  vector outside of the range  $[A_U, A_L]$ . Note that the discount factor is  $e^{-\Sigma_{vel} \cdot dt}$  where  $\Sigma_{vel} \cdot dt$  is the sum of the spot velocity terms multiplied by their corresponding time interval, up to the time of the current time step.

The output of the time integration consists of a matrix  $W$  of the  $w$  vectors at each time step and a vector of the transformed times,  $tau$ . With these data several figures can be displayed and errors and be calculated in cases where analytical solutions are available.

The graphical output will be discussed in the next section. The error comparison can be completed for the Merton Gaussian jump model since there is a series solution for the case of constant interest rates and volatilities given by equation (A.5). The error can then be calculated by using the norms as defined in section 2.2.5 for the absolute and relative error.

## A.5 Results

To demonstrate the effectiveness of the solution method several examples are considered and the results of various techniques are compared both for accuracy and efficiency. Two types of options are considered: European calls and puts. First off we will consider the Merton Gaussian jump model. Since an analytical solution exists we can consider the properties of these models before we consider the output of the numerical approximations. Consider a European call option and a European put option, both with unit strike price and one year to expiration. Now let the standard deviation of the underlying price be 0.15 and the interest rate be 0.05. The Black Scholes solution yields the expected result and is depicted in Figure A.9 below with the green solid line. But now consider the Merton Gaussian jump model. Recall the basic idea that we are adding jumps to the standard diffusion model. For this example consider the jumps are drawn from a normal distribution with standard deviation  $\delta = 1.0$  and arrival rate  $\lambda = 0.1$  (per year). This solution is represented by the red dashed line. At the extreme prices the Merton jump model collapses to the standard Black Scholes model. But near the strike price the option price from the Merton jump model is higher than the Black Scholes price; essentially this indicates that the volatility is higher in that region. This behavior makes sense since by adding the jumps there is more volatility than in the diffusion underlying the standard Black Scholes model.

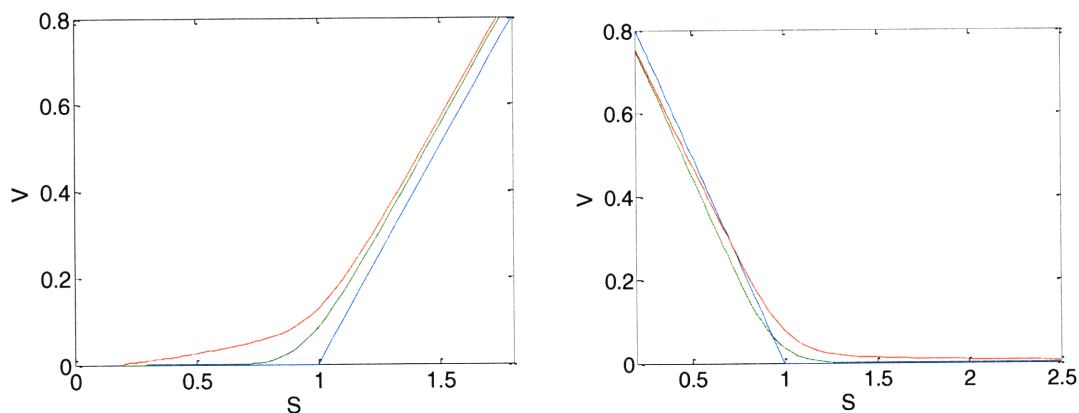


Figure A.9: Merton Gaussian jump model for European Call (left) and Put options

To further exemplify the properties of the non-Black Scholes models one can consider the volatility surface. The volatility surface displays volatility of an option implied by the model output over different asset prices and expiration dates. The implied volatility is the value one obtains for  $\sigma$  when the Black Scholes equation is solved for a given value when all of the other parameters are set. These values can be plotted over several

different coordinate systems. For this work the coordinates will be time to expiration,  $T$ , and moneyness,  $Ke^{-rT}/S$ . An example is plotted below for the Merton Gaussian jump model. The time ranges from 0 to 1 year and the stock price range is similar to the above plots.

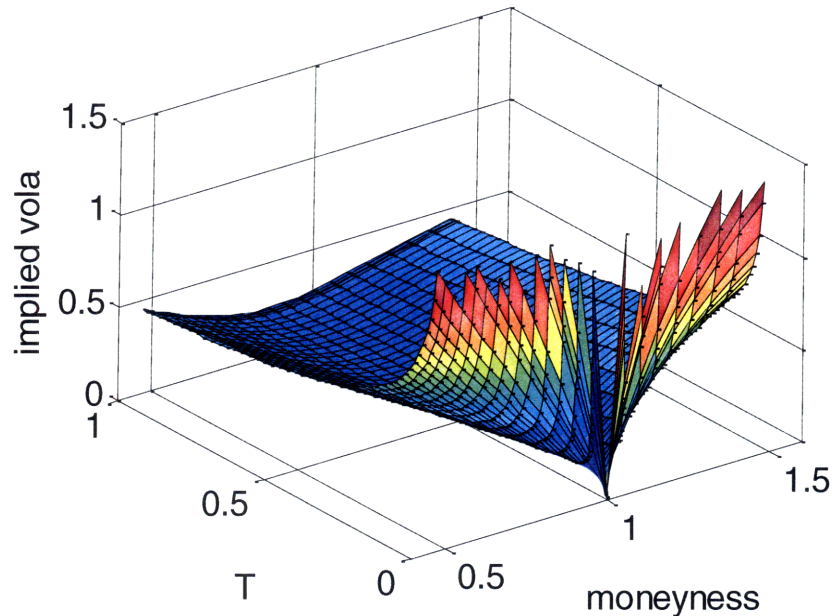


Figure A.10: Volatility surface, Put option with Merton Gaussian jumps

Several times are noteworthy about this figure. First of all the fact that there is any variation at all is significant. Black Scholes assumes a constant volatility but here different times and prices affect it. Note that when the option is at the money the volatility most closely approaches the Black Scholes volatility (0.15 in this case). Also as the option approaches expiration the volatility blows up: very small changes in asset price can result in large changes in option value. Farther out in time the volatility flattens out as sensitivity to change increases. Overall these features more accurately represent reality than the standard Black Scholes. The shape, often referred to as volatility smile, is observed in many traded options. There are several possible explanations for the shape but the most probable one is the fact that the market knows that significant jumps in price can occur. While this does not prove the validity of any model by itself it indicates that we are likely on the right track.

Now we can consider the numerical approximations of the Merton Gaussian jump model. We will vary the number of grid points and some of the parameters to create different scenarios. For each scenario an acceptable amount of error relative to the analytical solution will be chosen and the number of grid points will be fixed. Each time integration method will then be adjusted so as to obtain the desired accuracy and the statistics will be compared. For the methods with error tolerance built in, the relative tolerance is set to 0.01 and the absolute tolerance to 0.001.

The following parameters are fixed for all cases:

$$\begin{array}{llll}
 T = 1.0 & K = 1.0 & S_0 = 1.0 & \delta = 1.0 \\
 [S_L, S_U] = [2/3, 2.0] & [A_L, A_U] = [-5, 5] & [B_L, B_U] = [-7, 7] & 
 \end{array}$$

The results of the model runs are presented below. IMEX refers to the simple Euler IMEX method used in the literature; ode45 and ode15s are the built in MATLAB functions; and RKC is the modified 2-step Runge-Kutta Chebyshev method developed in this thesis.

**Table A.2: Performance of Methods, Marton Gaussian jump case**

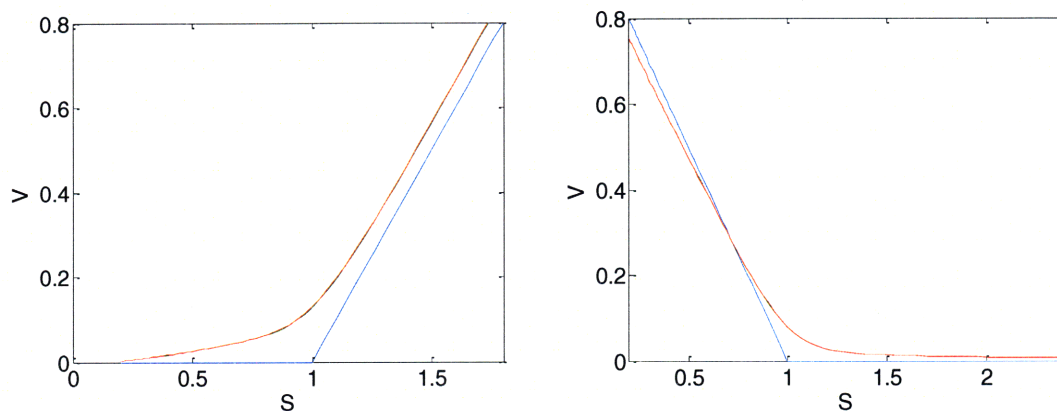
method	grid	$\sigma$	$r$	$\lambda$	t steps	fcn evals	time (s)
IMEX	301	0.15	0.05	0.1	30	29	14.2
ode45	301	0.15	0.05	0.1	19	127	2.5
ode15s	301	0.15	0.05	0.1	36	770	19.6
RKC	301	0.15	0.05	0.1	12	60	1.1
IMEX	601	0.15	0.05	0.1	30	29	98.1
ode45	601	0.15	0.05	0.1	55	337	25.6
ode15s	601	0.15	0.05	0.1	45	1501	147
RKC	601	0.15	0.05	0.1	29	145	11.8
IMEX	301	0.50	0.05	0.1	35	34	18.1
ode45	301	0.50	0.05	0.1	144	913	18.7
ode15s	301	0.50	0.05	0.1	62	801	24.8
RKC	301	0.50	0.05	0.1	77	385	8.3
IMEX	301	0.15	0.20	0.1	35	34	17.9
ode45	301	0.15	0.20	0.1	20	127	2.5
ode15s	301	0.15	0.20	0.1	37	771	21.8
RKC	301	0.15	0.20	0.1	21	105	2.2
IMEX	301	0.15	0.05	1.0	35	34	17.9
ode45	301	0.15	0.05	1.0	29	193	3.8
ode15s	301	0.15	0.05	1.0	42	777	22.3
RKC	301	0.15	0.05	1.0	22	108	2.4

There are several conclusions that we can draw from these data. Broadly, we see that the best choice is the RKC-based method across all of the scenarios. It is roughly a factor of two faster than ode45 and an order of magnitude or more faster than the two implicit methods. It should be noted that the IMEX method has no error correction so it is run several times to hone in on the number of time steps required to obtain the same accuracy as the other methods. The relative advantages do change in the different scenarios, however.

In the second scenario the grid density is doubled and there is roughly a ten-fold increase in time. The advantage in number of time steps has a relative decrease compared to the two implicit methods since their time scales vary with the efficiency of the matrix decomposition. None of the other scenarios result in such a dramatic increase. Next we see that increasing the volatility, the analog of increasing diffusion, demonstrates an

increase in the relative advantage of the RKC-based method since it can increase the internal stages to limit the increase in the number of time steps. However in the fourth scenario the relative advantage of the RKC-based method decreases. This is because the high interest rate, which corresponds to strong advection, increases the magnitude of the imaginary eigenvalues and even the two-stage method can only do so much to increase the stability domain in that direction. In the last case the main effect of the increased jump arrival rate is that the solution changes more rapidly so the error correction requires more time steps to maintain accuracy. Overall the IMEX method remains fairly constant since it is not adaptive; there was a slight increase in the number of time steps need to attain the desired accuracy when the solution evolved more substantially in the latter scenarios. But the advantage of the RKC-based approach is apparent. And in cases with variable parameters a non-adaptive method becomes even more difficult to manage.

Graphically, we see for the call and put options that there is close agreement between the analytical solution (green line) and the approximation (red dashed line). The results are similar for all of the above cases. These plots are from the conditions of the first scenario for a European call and put option.



**Figure A.11: Approximate and Exact call (left) and put option value solutions**

The red dashed line indicates the approximate solution and the green line is the exact solution.

The results for the volatility surface are very similar to the analytical solution as well. For the sake of variety consider the same conditions but with a  $\sigma$  value of 0.40 and a  $\lambda$  value of 0.5. The surface is as follows.

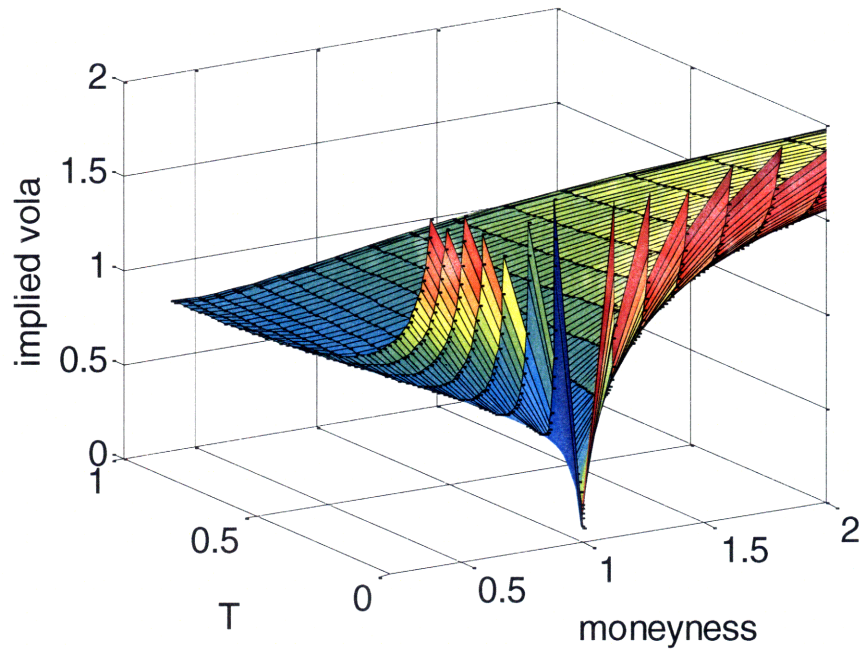


Figure A.12: Volatility surface, Put option with Merton Gaussian jumps

The parameter values are  $\sigma = 0.4$  and  $\lambda = 0.5$ .

Note that the shape is similar to the case with smaller  $\sigma$  and  $\lambda$ . However, the shift upward is more pronounced for strongly in the money options. This makes sense since the relative increase in volatility is greater for extreme asset values. This type of behavior is convenient for many situations but it is often useful to have more control over the tail behavior. This leads to the Gamma case.

The Gamma case is very similar from a numerics standpoint. The main challenge is the truncation and approximation due to the infinite value in the Levy density as discussed above. Once the matrix representing the integral is formed performance is much the same as above. The issue with the Gamma case is that there is no analytical solution against which to compare the numerical approximation. A very fine grid (1000 points) with very high error correction ( $1E-6$ ) is used as the reference solution for various cases; the other solutions are required to attain an acceptable closeness to this solution (as given by the relative and absolute errors). The constant parameters are the same as above. The parameters specific to the Gamma case are as follows:

$$\kappa = 0.5 \qquad \eta_- = 2.7 \qquad \eta_+ = 5.9$$

From here we obtain the following results for the first scenario; other results showed the same trends as above.

Table A.3: Performance of methods, Gamma case

method	grid	$\sigma$	$r$	$\lambda$	t steps	fcn evals	time (s)
IMEX	301	0.15	0.05	0.1	30	29	14.1

ode45	301	0.15	0.05	0.1	21	133	2.7
ode15s	301	0.15	0.05	0.1	39	774	19.2
RKC	301	0.15	0.05	0.1	12	64	1.1

The plots for a Gamma case with the above parameters are presented below. To gain a bit more insight, consider two different situations for comparison. In both cases we have  $\kappa = 0.5$  but in Case 1 (dashed green line) we have  $\eta_- = 0.75$  and  $\eta_+ = 3$  and for Case 2 (dashed red line) we have  $\eta_- = 3$  and  $\eta_+ = 0.75$ . For Case 1 in both the call and put options the value is relatively similar to the Black Scholes solution (not pictured) but the volatility is much greater in Case 2. Overall this demonstrates the flexibility of the Gamma model.

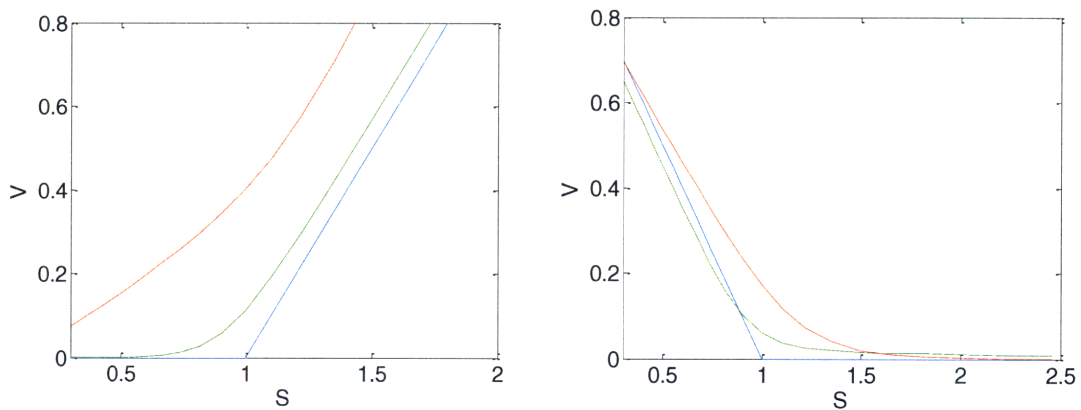
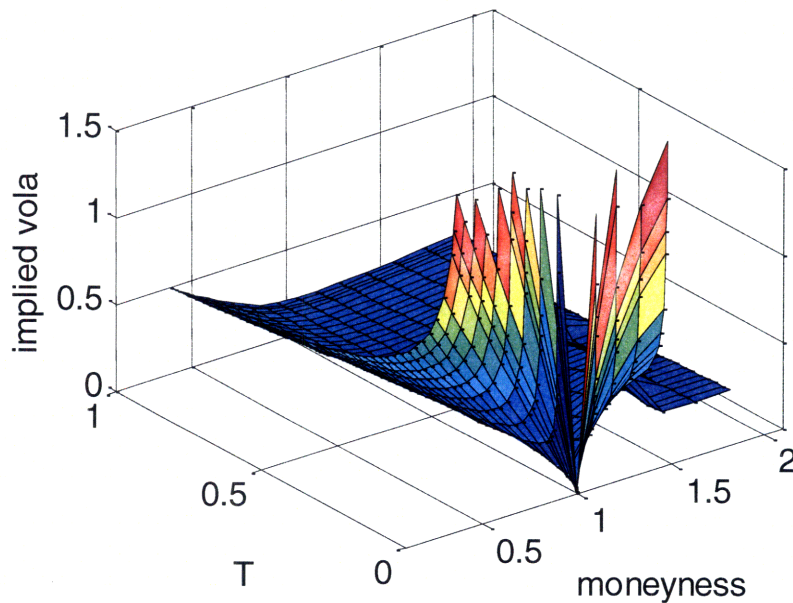


Figure A.13: Gamma model value of call (left) and put options; 2 cases

Further insight can be gained by considering the volatility surface. Consider Cases 1 and 2 from above on a put option.





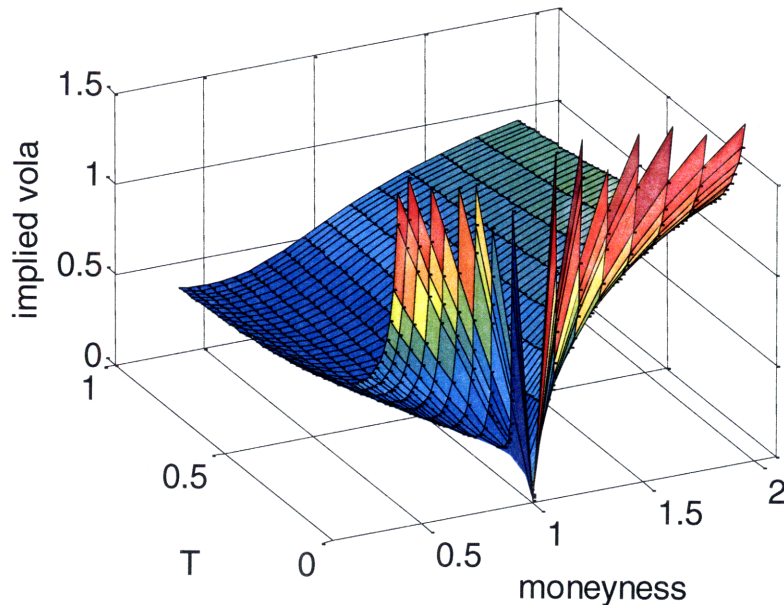


Figure A.14: Volatility surface for Put option, Gamma model, Cases 1 (top) and 2

Note that just interchanging the two parameters results in significantly different surfaces. Besides being greater in magnitude the curve in Case 2 is slightly concave down while Case 1 has a distinct upward trend at low moneyness. This once again demonstrates the flexibility of the Gamma model.

Finally let us consider what happens when we allow for parameters that can change with time. This is a very important feature since all of the parameters used in the model are likely to change in any real situation. Of course with any of the variable parameter cases there is no analytical solution against which to compare. Thus we must be content with merely assessing if the solution gives a reasonable result and presume that the error correction within the time integration methods is performing as designed.

The numerical setup is much the same as in the constant parameter cases. One substantial difference is that some of the intermediate parameters (e.g.  $\nu$ ) must be calculated at each evaluation of the main differential equation function (odefcn). Due to the nature in which MATLAB handles functions this results in a noticeably larger solution time under identical conditions to the constant parameter case. This is only an issue with the design of the programming language rather than the algorithm itself so changing to a different environment would eliminate the inefficiency.

Now we can consider a couple of different cases. First let us try a Merton Gaussian jump example with the following parameters fixed:

$$\begin{array}{llll}
 T = 1.0 & K = 1.0 & S_0 = 1.0 & \delta = 1.0 \\
 [S_L, S_U] = [2/3, 2.0] & [A_L, A_U] = [-5, 5] & [B_L, B_U] = [-7, 7]
 \end{array}$$

The other parameters can vary and are described by the following equations:

$$\sigma^2 = 0.4^2 \tau$$

$$r = 0.2 \tau$$

$$\lambda = 0.1 \tau$$

These are relatively simple functions but the evaluation of functions with more complicated and/or realistic behavior could easily be input given the proper data. The results are given in the table below. Note that the simple IMEX method is not included. This is because it is not adaptive so the changing parameters would require the specification of a variable time step vector after the fact. This is a large part of the reason that more advanced solution techniques were considered; in most practical applications it is not feasible to rely on manual adjustment of the time steps.

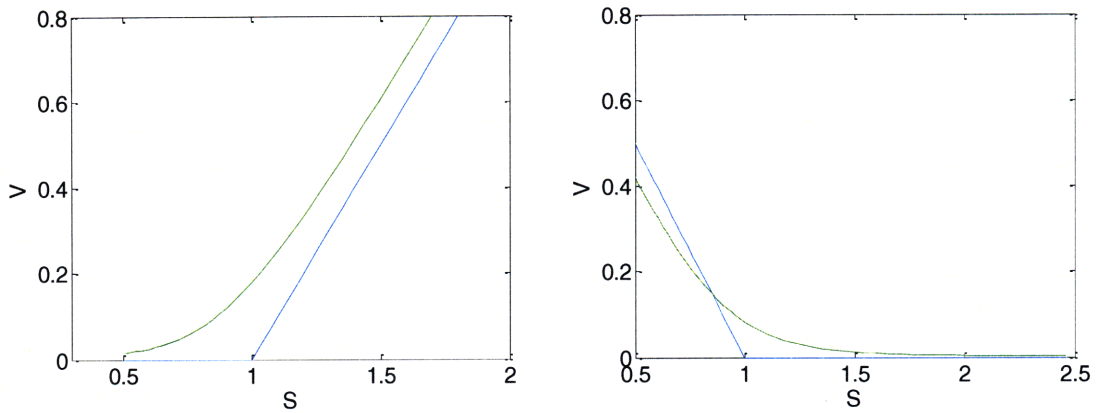
**Table A.4: Performance of Methods, Merton case**

method	grid	$\sigma^2$	$r$	$\lambda$	t steps	fcn evals	time (s)
ode45	301	$0.4^2 \tau$	$0.2 \tau$	$0.1 \tau$	23	163	24.5
ode15s	301	$0.4^2 \tau$	$0.2 \tau$	$0.1 \tau$	42	2018	301
RKC	301	$0.4^2 \tau$	$0.2 \tau$	$0.1 \tau$	18	90	16.3

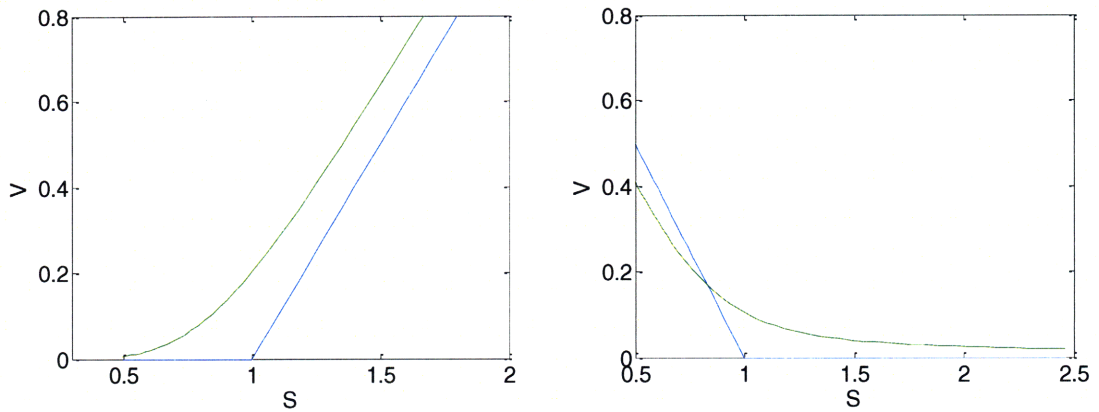
The solution times are substantially longer for all cases as noted above. However the most important factors to consider are the time steps and function evaluations. The issue that stands out is the fact that the implicit ode15s method is substantially slower; proportionally even more so than in the constant parameter cases. This is likely due to the fact that it must calculate (approximately) the Jacobian matrix more frequently since it changes significantly with time. The ode45 and RKC method performed similarly to the constant parameter case in relative terms. The number of time steps and function evaluations was toward the lower end of the ranges for the constant parameter cases. This demonstrates that the methods are adapting to the parameters as they step through time despite the high values at large  $\tau$ . The modified Runge-Kutta Chebyshev method is the best performer as expected but its relative advantage over ode45 has decreased somewhat. Most likely this is due to the relatively high  $r$  and  $\sigma$  values (both of which affect the “velocity” term and consequently the stability in the imaginary direction) for some of the time range.

The performance results for the Gamma case are similar. Most of the challenges have to do with the interest rate and volatility parameters as was just discussed above. The same tests were run with the additional parameters  $\kappa = 0.5$ ,  $\eta_- = 0.75$ , and  $\eta_+ = 3$ .

As expected the graphical results for both the Merton and Gamma cases are similar to their constant parameter counterparts. Qualitatively it is impossible to point to any one feature that demonstrates the parameters have changed over time. Nevertheless the results are plausible. Overall the best way to check the correctness of the results would be to compare them against real data collected for under some known values for the parameters. It is still worthwhile to observe results based on the above conditions.

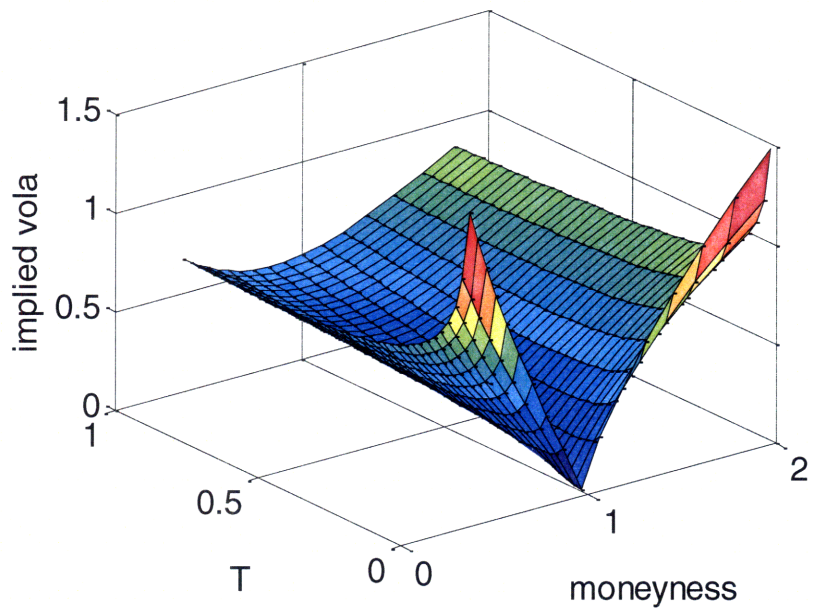
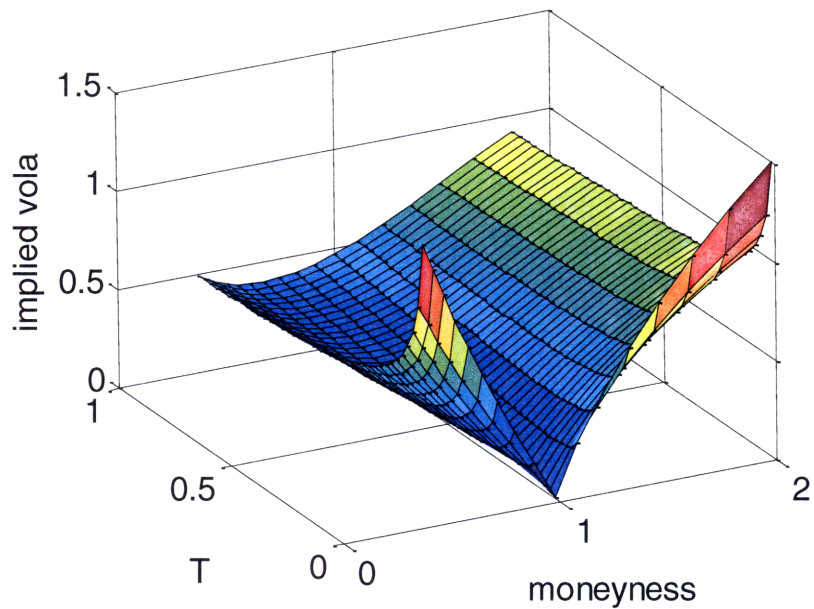


**Figure A.15: Variable parameter case, Merton model for call (left) and put options**



**Figure A.16: Variable parameter case, Gamma model for call (left) and put options**

As was mentioned above it is difficult to conclude anything about these figures. We see some of the characteristic features such as the fatter tail on the Gamma model. From the volatility surfaces it is also difficult to identify any definite feature due to the the parameters.



**Figure A.17: Volatility surfaces, variable cases for Merton (top) and Gamma models**

The two results look more similar to each other than in the constant parameter case. This is most likely due to the large value for  $\sigma$  at high  $\tau$  values which overshadows the effect of increased volatility due to the jumps.

## **A.6 Conclusions**

This paper has demonstrated the numerical solution methods developed in the main thesis can be effectively applied to option valuation problems in finance. Specifically this paper developed the background for the standard Black-Scholes based models and the extension into models incorporating jumps in asset price. This new model was written in the form of a partial integro-differential equation similar in form to those discussed in the main thesis. Next the paper discussed the implantation in MATLAB of a solution method. This method incorporated positivity-preserving discretizations and employed a modified Runge-Kutta Chebyshev integration algorithm.

The types of problems solved included European put and call options and the description of the asset price movements were based on the Merton Gaussian jump diffusion and Gamma infinite activity models. The RKC based model developed in the thesis performed significantly better than the simple implicit-explicit integration method suggested in the literature. It even had an advantage over the built-in methods of MATLAB. Also of note is fact that it could handle situations with variable parameters, which is impossible in standard models. The ability to handle variable parameter models is necessary to model in realistic situation.

Overall this paper has shown another application for the numerical techniques developed in the main thesis. While the results show significant promise there are still many opportunities to extend the work. Option types beyond European calls and puts could be considered with relatively minor changes; the PIDE approach makes it relatively straightforward to add boundary conditions. Along this track one very useful application is multi-asset options. There are several interesting challenges in modeling such options, such as the correlation matrix between assets but the greatest challenge is the large number of grid points necessary for describing such systems as the dimensionality increases. The more efficient solutions algorithms of this work provide a starting point but they must be implemented in a more efficient manner to be truly useful on large-scale problems. One of the ways to do this is through the use of parallel processing within the program. Since the method is explicit in nature each row of the matrix could actually be evaluated simultaneously at each time step, dramatically decreasing the solution time. In general a more advanced programming language should be used for any practical implementation.

The other main extension is the inclusion of real data in some form. The most obvious use for this could be to check if the results of the models agree with the data. But it would be more interesting to calibrate the model parameters against the data. This goes back to the parameter estimation concepts discussed in the main thesis; the concepts regarding problem domain information and the advantages of using explicit methods in such systems would apply in these option pricing examples as well.

As with the main thesis there are many opportunities for new exploration based on this work. It is hoped that future researchers will be able to use this work as a starting point.

### **A.7 Capstone Paper Works Cited**

- Baxter, M.; Rennie, R. (1996) *Financial Calculus: An Introduction to Derivative Pricing*. Cambridge University Press.
- Cont, R.; Voltchkova, E. (2005) "A Finite Difference Scheme for Option Pricing in Jump Diffusion and Exponential Lévy Models". *SIAM J. Numerical Analysis*, v. 43 No. 4, 1596-1626
- Cont, R.; Tankov, P. (2004) *Financial Modelling with Jump Processes*. Chapman & Hall/CRC.
- Merton, R. (1976) "Option Pricing When Underlying Stock Returns Are Discontinuous". Working Paper, MIT.
- Wilmott, P.; Howison, S.; Dewynne, J. (1995) *The Mathematics of Financial Derivatives*. Cambridge University Press.

## Appendix B: References

- Araujo, A.; Ferreira, J.A.; Oliveira, P. (2004) "Qualitative Behavior of Numerical Traveling Solutions for Reaction-Diffusion Equations with Memory". Preprint, Departamento de Matematica, Universidade de Coimbra.
- Ascher, U.M.; Ruuth, S.J.; Spiteri, R.J. (1997) "Implicit-Explicit Runge-Kutta Methods for Time-Dependent Partial Differential Equations". *Applied Numerical Mathematics*, v. 25, 151-167.
- Beers, K. (2004) *Advanced Engineering Mathematics and Numerical Methods*. Unpublished Class Notes, MIT.
- Berridge, S.J.; Schumacher, J.M. (2004) "Pricing High-Dimensional American Options Using Local Consistency Conditions". CentER Discussion Paper.
- Borrelli, R.; Coleman, C. (1998) *Differential Equations: A Modeling Perspective*. John Wiley & Sons.
- Bott, A. (1989) "A Positive Definite Advection Scheme Obtained by Nonlinear Renormalization of the Advective Fluxes". *Monthly Weather Review*, v. 117, 1006-1015.
- Briani, M.; Natalini, R.; Russo, G. "Implicit-Explicit Numerical Schemes for Jump Diffusion Processes". Working Paper, <http://www.iac.rm.cnr.it/~natalini/ps/BNR.pdf>.
- Butcher, J.C. (2003) *Numerical Analysis of Ordinary Differential Equations*. John Wiley & Sons.
- Carrayrou, J.; Mose, R.; Behra, P. (2004) "Operator-Splitting Procedures for Reactive Transport and Comparison of Mass Balance Errors". *J. Computational Physics*, v. 68, 239-268.
- Cont, R.; Voltchkova, E. (2005) "A Finite Difference Scheme for Option Pricing in Jump Diffusion and Exponential Lévy Models". *SIAM J. Numerical Analysis*, v. 43 No. 4, 1596-1626
- Cont, R.; Tankov, P. (2004) *Financial Modelling with Jump Processes*. Chapman & Hall/CRC.
- Deen, W. (1998) *Analysis of Transport Phenomena*. Oxford Univ. Press.
- Deutsch, H-P. (2004) *Derivatives and Internal Models*, 3 ed. Palgrave Macmillan.
- Diaz, J.M.F.; Braña, M.A.R.; García, B.A.; Muñoz, C.G-P.; Nieto, P.J.G. (1999) "Difficulties Inherent to the Use of Analytic Solution of the Condensation-

- Evaporation Equation for Multicomponent Aerosols”. *Atmospheric Environment*, v. 33, 1245-1259.
- Elvin, A.; Laing, C.R. (2005) “Evaluation of Numerical Integration Schemes for a Partial Integro-Differential Equation”. *Res. Lett. Inf. Math. Sci.*, v. 7, 171-186.
- Frankel, J.I.; Osborne, G.E. (1999) “A New Time Treatment for Solving Partial Integro-Differential Equations of Radiative Transport”. *IMA J. Numerical Analysis*, v. 19, 91-103.
- Hairer, E.; Nørsett, S.P.; Wanner, G. (1993) *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2 ed. Springer-Verlag.
- Hairer, E.; Wanner, G. (1993) *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2 ed. Springer-Verlag.
- Hairer, E.; Lubich, C.; Wanner, G. (2002) *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer-Verlag.
- Houwen, P.J.v.d.; Sommeijer, B.P. (1982) “A Special Class of Multistep Runge-Kutta Methods with Extended Real Stability Interval”. *IMA J. Numerical Analysis*, v. 2, 183-209.
- Hull, J.C. (2006) *Options, Futures, and Other Derivatives*. Prentice Hall.
- Hundsdoerfer, W.; Koren, B.; van Loon, M.; Verwer, J.G. (1995) “A Positive Finite Difference Advection Scheme”. *J. Computational Physics*, v. 117, 35-46.
- Hundsdoerfer, W.; Verwer, J.G. (2003) *Numerical Solution of Time Dependent Advection-Diffusion-Reaction Equations*. Springer-Verlag.
- Jerri, A.J. (1999) *Introduction to Integral Equations with Applications*, 2 ed. John Wiley & Sons.
- Iserles, A. (1996) *A First Course in the Numerical Analysis of Differential Equations*. Cambridge Univ. Press.
- Karlsen, K.H.; Lie, K-A.; Natvig, J.R.; Nordhaug, H.F.; Dahle, H.K. (2001) “Operator Splitting Techniques for Systems of Convection-Diffusion Equations: Nonlinear Error Mechanisms and Correction Strategies”. *J. Computational Physics*, v. 173, 636-661.
- Kennedy, C.A.; Carpenter, M.H. (2003) “Additive Runge-Kutta Schemes for Convection-Diffusion-Reaction Equations”. *Applied Numerical Mathematics*, v. 44, 139-181.



- Laing, C.R.; Troy, W.C.; Gutkin, B.; Ermentrout, G.B. (2002) "Multiple Bumps in a Neuronal Model of Working Memory". *SIAM J. Applied Mathematics*, v. 63, 62-97.
- Lambert, J.D. (1991) *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. John Wiley & Sons.
- Lapidus, L.; Pinder, G. (1982) *Numerical Solutions of Partial Differential Equations in Science and Engineering*. John Wiley & Sons.
- Leer, B.v. (1985) *Large Scale Computations in Fluid Mechanics*, Engquist, B.E., editor. Am. Math. Soc.
- Marchuk, G.I. (1994) *Methods of Numerical Mathematics*. Springer-Verlag.
- Marchuk, G.I., ed. (1994) *Numerical Methods and Applications*. CRC Press.
- McRae, G.J.; Goodin, W.R.; Seinfeld, J.J. (1982) "Numerical Solution of the Atmospheric Diffusion Equation for Chemically Reacting Flows". *J. Computational Physics*, v. 45, 1-42.
- Mitchell, A.R.; Griffiths, D.F. (1980) *The Finite Difference Method in Partial Differential Equations*. John Wiley & Sons.
- Modest, M. (2003) *Radiative Heat Transfer*, 2 ed. Academic Press.
- Mogilner, A.; Gueron, S. (2000) "Integro-Differential Model for Pattern Formation in Bacterial Swarm". Working Paper, <http://citeseer.ist.psu.edu/183470.html>.
- Murray, J.D. (1989) *Mathematical Biology*. Springer-Verlag.
- Obrigkeit, D. (2001) *Numerical Solution of Multicomponent Population Balance Systems with Applications to Particulate Processes*. MIT PhD Thesis.
- Øksendal, B. (2003) *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag.
- Pacelli, G.; Recchioni, M.C.; Zirilli, F. (1999) "A Hybrid Model for Pricing European Options Based on Multiple Assets with Transaction Costs". *Applied Mathematical Finance*, v. 6, 61-85.
- Prasad, K.K.; Hering, R.G. (1969) "Transient Radiative Heat Transfer in a Plane Layer". *Int. J. Heat Mass Transfer*, v. 12, 1331-1337.
- Sandu, A. (2001) "Positive Numerical Integration Methods for Chemical Kinetic Systems". *J. Computational Physics*, v. 170, 589-602.

- Seinfeld, J.H.; Pandis, S.N. (2006) *Atmospheric Chemistry: From Air Pollution to Climate Change*. John Wiley & Sons.
- Shampine, L.F.; Baca, L.S. (1984) "Error Estimators for Stiff Differential Equations". *J. of Computational and Applied Mathematics*, v. 11, 197-207.
- Shampine, L.F.; Sommeijer, B.P.; Verwer, J.G. (2005) "IRKC: an IMEX Solver for Stiff Diffusion-Reaction PDEs". Working Paper, <http://www.cwi.nl/ftp/CWIreports/MAS/MAS-E0513.pdf>.
- Sommeijer, B.P.; Verwer, J.G. (2006) "On Stabilized Integration for Time-Dependent PDEs". Working Paper, <http://www.cwi.nl/ftp/CWIreports/MAS/MAS-E0616.pdf>.
- Vasudeva Murthy, A.S.; Verwer, J.G. (1991) "Solving Parabolic Integro-Differential Equations by an explicit integration method". Stichting Mathematisch Centrum.
- Verwer, J.G.; Sommeijer, B.P.; Hundsdorfer, W. (2004), RKC Time-Stepping for Advection-Diffusion-Reaction Problems". *J. Computational Physics*, v. 201, 61-79.
- Wesseling, P. (2001) *Principles of Computational Fluid Dynamics*. Springer.
- Wilmott, P.; Howison, S.; Dewynne, J. (1995) *The Mathematics of Financial Derivatives*. Cambridge Univ. Press.
- Zhang, L. (2004) *Runge-Kutta-Chebyshev Methods for Advection-Diffusion-Reaction Problems*, Masters Project.

