Adaptive Time Stepping for the Neutron Transport Solution with the Alpha Eigenvalue

by

Andrew R. Gerlach

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in The University of Michigan
2020

Doctoral Committee:

Professor Emeritus John C. Lee, Chair
Professor Thomas Downar
Assistant Professor Brendan Kochunas
Associate Professor Shravan Veerapaneni

Andrew R. Gerlach

agerlach@umich.edu

ORCID iD: 0000-0002-4022-1356

Dedication

I'd like to dedicate this thesis to my mom. I wish she could be here to see me through this, yet somehow I know she is.

Acknowledgements

While this thesis is the culmination of my work as a doctoral student at the University of Michigan, it would not have been possible without the guidance, assistance, and support of an entire community.

First and foremost, I'd like to thank Prof. John Lee for offering me the chance to pursue my doctorate. His guidance, through thick and thin, has taught me profound lessons well beyond the vagaries of nuclear physics that I will carry with me forever. I'd like to thank Prof. Downar as well for his support throughout this process.

I owe a huge debt of gratitude to Prof. Brendan Kochunas for countless hours helping me through the MPACT learning curve. Also in that vein, I'd like to thank Dr. Benjamin Collins at Oak Ridge National Laboratory and Dan Jabaay, as well as Dr. Bob Salko and Dr. Aaron Wysocki at Oak Ridge for their assistance with my CTF coupling work.

Another round of thanks is owed to my compatriots Dr. Aaron Graham, Dr. Aaron Bevill, Dr. Tim Burke, Dr. Joel Kulesza, Dr. Ben Yee, and Dr. Brody Bassett for their help, both technical and in maintaining sanity.

I would also like to acknowledge the CASL program and NRC for providing funding.

On a more personal level, I wouldn't be where I am today without the support of my family. My parents and my wife's parents have been simply amazing. Most importantly, I owe a huge debt of gratitude to my incredible wife, Dr. Angela Ianni. And while they may have actually set me back more than they have helped me toward a PhD, my daughters Ilaria and Nina have provided the absolute best hugs when I've needed them most.

Table of Contents

Dedication ii
Acknowledgements iii
List of Figures vi
List of Tables x
Abstract xiii
Chapter 1: Introduction 1
1.1 Neutron Transport and Approximations . . . 2
1.1.1 The Independent Variables and Discretization . . . 4
1.1.2 Multigroup Approximation . . . 6
1.1.3 Angular Approximations . . . 7
1.1.4 Scattering Approximations . . . 8
1.1.5 Diffusion Approximation . . . 9
1.2 Eigenvalue Formulations of the Transport Equation . . . 10
1.2.1 k-eigenvalue . . . 10
1.2.2 α-eigenvalue . . . 11
1.3 Time Dependent Neutron Transport and Reactor Kinetics . . . 14
1.3.1 Point Kinetics . . . 14
Chapter 2: Calculation Methodology 16
2.1 Basic Solution Algorithm . . . 16
2.1.1 Source Iteration . . . 16
2.1.2 Jacobian-Free Newton-Krylov Method . . . 17
2.2 MPACT Methodology . . . 18
2.2.1 2D/1D Formulation . . . 19
2.2.2 Method of Characteristics . . . 20
2.2.3 Coarse Mesh Finite Difference Acceleration . . . 22
2.2.4 Transients . . . 23
2.3 Modification of MPACT for Solution by JFNK . . . 25
Chapter 3: Adaptive Time Stepping 27
3.1 Error Analysis of Time Discretization . . . 28
3.1.1 Finite Difference Approach . . . 30
3.1.2 α-Eigenvalue Eigenvalue Approach . . . 30

Chapter 4: α-eigenvalue Calculations with MPACT **32**
4.1 Basic $k-\alpha$ Iteration in MPACT . . . 35
4.2 Eigenvector Treatment for $k-\alpha$ Iteration . . . 38
4.3 Frequency of α Update . . . 42
4.4 α-Eigenvalue Calculation with JFNK . . . 42
4.5 Summary of Steady State $k-\alpha$ Calculation Investigations . . . 43
Chapter 5: ATS Calculations with Finite Differencing in MPACT **44**
5.1 Finite Difference Results without Feedback . . . 45
5.1.1 ATS Criteria for Finite Difference Method without Feedback . . . 46
5.1.2 Finite Difference Second Derivative Estimation . . . 49
5.1.3 Finite Difference Flux Choice . . . 51
5.2 Finite Difference Results with Feedback . . . 55
5.2.1 ATS Criteria for Finite Difference Method with Feedback . . . 55
5.2.2 Finite Difference Second Derivative Estimation and Growth Limits . . . 59
5.2.3 Finite Difference Flux Choice . . . 63
5.3 Summary of Finite Difference Results . . . 67
Chapter 6: Adaptive Time Stepping Calculations with the α-Eigenvalue in MPACT **68**
6.1 α-Eigenvalue Calculation Method . . . 68
6.2 α-Eigenvalue Results without Feedback . . . 70
6.2.1 Adaptive Time Stepping Criteria with the α-Eigenvalue without Feedback . . . 70
6.2.2 α-Eigenvalue Flux Choice . . . 73
6.2.3 Calculation of the α-Eigenvalue by Point Kinetics for ATS without Feedback . . . 74
6.3 α-Eigenvalue Results with Feedback . . . 76
6.3.1 Adaptive Time Stepping Criteria with the α-Eigenvalue . . . 76
6.3.2 α-Eigenvalue Calculation by Point Kinetics with Feedback . . . 79
6.3.3 Examination of the Role of Feedback . . . 83
6.4 Summary . . . 86
Chapter 7: Benchmarks Calculations **87**
7.1 SPERT Mini-Core . . . 87
7.1.1 SPERT Mini-Core Results . . . 88
7.1.2 SPERT Summary . . . 93
7.2 C5G7-TD Benchmark . . . 94
7.2.1 C5G7-TD Simulations . . . 97
7.2.2 C5G7-TD Summary . . . 100
Chapter 8: Summary and Conclusions **101**
8.1 Future Work . . . 101
Appendices **104**
Bibliography **121**

List of Figures

Figure 1.1 Energy Dependence of the Fission and Capture Cross-section for U-238 6
Figure 1.2 Multigroup Approximation of the Energy Spectrum 7
Figure 1.3 Example Spectrum for the α-eigenvalue Showing Points, Lines, and Continuum [4] . . 13

Figure 3.1 Demonstration of how ATS can redistribute the time steps more efficiently 27

Figure 4.1 MPACT Single Pin Model 34
Figure 4.2 Evolution of k and α_p eigenvalues for $k-\alpha$ Iteration with Algorithm 3 for pin04 . . . 36
Figure 4.3 Radial Comparison of k and α_p Eigenvectors for$k-\alpha$ Iteration with Algorithm 3 for pin04 37
Figure 4.4 Evolution of k and α_p eigenvalues for $k-\alpha$ Iteration with Algorithm 4 for pin04 . . . 39
Figure 4.5 Radial Comparison of k and α_p Eigenvectors for $k-\alpha$ Iteration with Algorithm 4 for pin04 40
Figure 4.6 Radial comparison of α_p eigenvectors for fully converged $k-\alpha$ Iteration with Algorithm 3 for pin04 41

Figure 5.1 Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type, Time Steps Shown in Parentheses . 47
Figure 5.2 Single Pin Subprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type 48
Figure 5.3 Single Pin Subprompt Transient Power and Second Derivative of the Angular Flux . . 48
Figure 5.4 Single Pin Subprompt Power Curves for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type, Time Steps Shown in Parentheses 49
Figure 5.5 Single Pin Subprompt Time Step Size for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type 50
Figure 5.6 Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type, Time Steps Shown in Parentheses . . 51
Figure 5.7 Single Pin Subprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type 52
Figure 5.8 Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type, Time Steps Shown in Parentheses 53

Figure 5.9 Single Pin Subprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type 54
Figure 5.10 Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type, Time Steps Shown in Parentheses . 57
Figure 5.11 Single Pin Superprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type 58
Figure 5.12 Single Pin Superprompt Transient Power and Second Derivative of the Angular Flux . 58
Figure 5.13 Single Pin Superprompt Power Curves for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type, Time Steps Shown in Parentheses 59
Figure 5.14 Single Pin Superprompt Time Step Size for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type 60
Figure 5.15 Single Pin Superprompt Time Step Size as a Function of Growth Limit for ATS with Nested Finite Difference Method Using the Angular Flux for Absolute Direct Criteria 61
Figure 5.16 Single Pin Superprompt Time Step Size as a Function of Growth Limit for ATS with Nested Finite Difference Method Using the Angular Flux for Relative Direct Criteria 61
Figure 5.17 Single Pin Superprompt Time Step Size as a Function of Growth Limit for ATS with Nested Finite Difference Method Using the Angular Flux for Absolute Integrated Criteria . . 62
Figure 5.18 Single Pin Superprompt Time Step Size as a Function of Growth Limit for ATS with Nested Finite Difference Method Using the Angular Flux for Relative Integrated Criteria . . 62
Figure 5.19 Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type, Time Steps Shown in Parentheses . . 63
Figure 5.20 Single Pin Superprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type 64
Figure 5.21 A Comparison of Logarithmic Second Derivative of Power, Scalar Flux, and Angular Flux for Single Pin Superprompt Transient 65
Figure 5.22 Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type, Time Steps Shown in Parentheses 65
Figure 5.23 Single Pin Superprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type 66

Figure 6.1 A Comparison of the Different α-Eigenvalues and the Logrithmic Derivative of Power for Single Pin Superprompt Transient 69
Figure 6.2 Single Pin Subprompt Power Curves for ATS with α Method Using the Angular Flux for each Criteria Type, Time Steps Shown in Parentheses 71
Figure 6.3 Single Pin Subprompt Time Step Size for ATS with α Method Using the Angular Flux for each Criteria Type 72
Figure 6.4 A Comparison of Time Constants for the α-Eigenvalue Calculation Methods without Feedback 75
Figure 6.5 A Comparison of Second Derivatives for the α-Eigenvalue Calculation Methods without Feedback 75
Figure 6.6 Single Pin Superprompt Power Curves for ATS with α Method Using the Angular Flux for each Criteria Type, Time Steps Shown in Parentheses 77

Figure 6.7 Single Pin Superprompt Time Step Size for ATS with α Method Using the Angular Flux for each Criteria Type 78
Figure 6.8 Single Pin Superprompt Power Curves for ATS with α Method Using the Scalar Flux for each Criteria Type, Time Steps Shown in Parentheses 80
Figure 6.9 Single Pin Superprompt Time Step Size for ATS with α Method Using the Scalar Flux for each Criteria Type 81
Figure 6.10 A Comparison of Time Constants for the α-Eigenvalue Calculation Methods with Feedback 81
Figure 6.11 A Comparison of Second Derivatives for the α-Eigenvalue Calculation Methods with Feedback 82
Figure 6.12 Single Pin Superprompt α-Eigenvalue and Transient Time Constant for ATS with Relative Integrated Criteria 83
Figure 6.13 Single Pin Superprompt Logarithmic Second Derivative Estimated from α for ATS with Relative Integrated Criteria 84
Figure 6.14 Single Pin Superprompt Reference Fuel Temperature Rate of Change 85
Figure 6.15 Single Pin Superprompt Error in Fuel Temperature Rate of Change 85

Figure 7.1 SPERT Mini-core Overall Configuration 88
Figure 7.2 SPERT Power Curves with $k - \alpha$ Alpha ATS 89
Figure 7.3 Time Step Sizes for SPERT with $k - \alpha$ Alpha ATS 90
Figure 7.4 Time Constants for SPERT with $k - \alpha$ Alpha ATS 90
Figure 7.5 SPERT Power Curves with PKE Alpha ATS 91
Figure 7.6 Time Step Sizes for SPERT with PKE Alpha ATS 92
Figure 7.7 Time Constants for SPERT with PKE Alpha ATS 92
Figure 7.8 C5G7-TD 2D Overall Configuration 94
Figure 7.9 C5G7-TD Assembly Detail 95
Figure 7.10 C5G7-TD Pin Cell Layout 95
Figure 7.11 C5G7 TDW1 Specification for Control Rod Movement 96
Figure 7.12 C5G7-TD Power Curves with $k - \alpha$ Alpha ATS 97
Figure 7.13 Closeup of C5G7-TD Power Curves Peaks with $k - \alpha$ Alpha ATS 98
Figure 7.14 Time Step Sizes for C5G7-TD with $k - \alpha$ Alpha ATS 98
Figure 7.15 Time Constants for C5G7-TD with $k - \alpha$ Alpha ATS 99

Figure A.1 Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 104
Figure A.2 Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 106
Figure A.3 Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 107

Figure A.4 Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 108
Figure A.5 Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 109
Figure A.6 Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 110
Figure A.7 Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 111
Figure A.8 Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 112
Figure A.9 Single Pin Subprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 113
Figure A.10 Single Pin Subprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 114
Figure A.11 Single Pin Subprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 115
Figure A.12 Single Pin Subprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 116
Figure A.13 Single Pin Superprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 117
Figure A.14 Single Pin Superprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 118
Figure A.15 Single Pin Superprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 119
Figure A.16 Single Pin Superprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses 120

List of Tables

Table 3.1 Qualitative expectations of the ATS criteria 29

Table 4.1 Single Pin Steady State α Calculation Test Cases 34
Table 4.2 Results of $k-\alpha$ Iteration with Algorithm 3 for Steady State Pins for α_a 36
Table 4.3 Results of $k-\alpha$ Iteration with Algorithm 3 for Steady State Pins for α_p 36
Table 4.4 α_a Flux Comparison for $k-\alpha$ Iteration with Algorithm 3 for Steady State Pins 36
Table 4.5 α_p Flux Comparison for $k-\alpha$ Iteration with Algorithm 3 for Steady State Pins 37
Table 4.6 Results of $k-\alpha$ Iteration with Algorithm 4 for Steady State Pins for α_a 39
Table 4.7 Results of $k-\alpha$ Iteration with Algorithm 4 for Steady State Pins for α_p 39
Table 4.8 α_a Flux Comparison for $k-\alpha$ Iteration with Algorithm 4 for Steady State Pins 40
Table 4.9 α_p Flux Comparison for $k-\alpha$ Iteration with Algorithm 4 for Steady State Pins 40
Table 4.10 Comparison of Flux for Algorithms 3 and 4 for α_p 40
Table 4.11 Comparison of flux for Algorithms 3 and 4 for α_p 40
Table 4.12 Results of JFNK $k-\alpha$ Iteration with Algorithm 3 for Steady State Pins for α_p 42
Table 4.13 Results of JFNK $k-\alpha$ Iteration with Algorithm 4 for Steady State Pins for α_p 43

Table 5.1 Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type 47
Table 5.2 Single Pin Subprompt Error Comparison for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type 50
Table 5.3 Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type 52
Table 5.4 Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type 53
Table 5.5 Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type 57
Table 5.6 Single Pin Superprompt Error Comparison for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type 60
Table 5.7 Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type 64
Table 5.8 Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type 66

Table 6.1 Single Pin Subprompt Error Comparison for ATS with α Method Using the Angular Flux for each Criteria Type 71
Table 6.2 Single Pin Subprompt Time Comparison for ATS with α Method Using the Scalar Flux for each Criteria Type 73
Table 6.3 Single Pin Subprompt Time Comparison for ATS with PKE α Method Using the Scalar Flux for each Criteria Type 74
Table 6.4 Single Pin Superprompt Error Comparison for ATS with α Method Using the Angular Flux for each Criteria Type 77
Table 6.5 Single Pin Superprompt Error Comparison for ATS with α Method Using the Scalar Flux for each Criteria Type 80

Table 7.1 Error Comparison for ATS with $k-\alpha$ Alpha Method For Each Criteria 89
Table 7.2 Error Comparison for ATS with PKE Alpha Method For Each Criteria 91

Table A.1 Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances . . . 105
Table A.2 Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances . . . 106
Table A.3 Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances . 107
Table A.4 Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances . 108
Table A.5 Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances . . . 109
Table A.6 Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances . . . 110
Table A.7 Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances . 111
Table A.8 Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances . 112
Table A.9 Single Pin Subprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances 113
Table A.10 Single Pin Subprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances 114
Table A.11 Single Pin Subprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances 115
Table A.12 Single Pin Subprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances 116
Table A.13 Single Pin Superprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances 117
Table A.14 Single Pin Superprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances 118

Table A.15 Single Pin Superprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances 119
Table A.16 Single Pin Superprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances 120

Abstract

Reactivity insertion accidents are one of the design-basis accidents that drive nuclear reactor limits. In order to operate nuclear reactors both safely and efficiently, reactor response to such an accident scenario must be well understood through simulation. While state-of-the-art reactor codes, such as MPACT, are capable of modeling transport physics for heterogeneous geometry, the computational cost is significant. This cost is only amplified for reactor transients, where the solution in the time domain is obtained through a series of calculations at discrete time points. One way to improve the computational efficiency is to adaptively select the time points at which to perform a calculation based on the evolution of the reactor through the transient.

The objective of this work is to develop an adaptive time stepping algorithm specifically for neutron transport that is able to properly characterize the evolution of the reactor throughout a transient and provide an appropriate time step size based on that characterization. In order to accomplish this, the leading order error term of the time discretization is limited. For implicit Euler, the standard time discretization method for neutron transport codes, this error term is inversely proportional to the second derivative of the angular flux in time.

Two methods are investigated for estimating the second derivative—a traditional finite difference approach and a novel alpha (time) eigenvalue approach. The methods were implemented in MPACT and characterized on a variety of transient test cases. The finite difference method is shown to suffer from two major drawbacks: untenable storage demands and oscillatory time step selection. The former issue is shown to be resolved satisfactorily by substituting the scalar flux for the angular flux without loss of accuracy. The latter issue is partially resolved by employing an alternative formulation of the finite difference approximation, but this only serves to reduce, not eliminate, the oscillations. The alpha eigenvalue method is shown to resolve both of these issues, though at a higher computational cost than the finite difference method. Both methods present a range of choices that are explored and characterized for their performance.

The result of this work is a robust adaptive time stepping scheme for MPACT that is able to increase computational efficiency of reactor transient simulations without an adversarial effect on accuracy. The capability is demonstrated for one of the C5G7 computational benchmarks and a miniature version of the SPERT reactor experiments.

Chapter 1

Introduction

Nuclear power represents a carbon-free energy source that, given current trends in climate change due to carbon emissions, should play an increasing role in power production around the world. In order for this to happen, we must be able to reliably and efficiently model nuclear reactors to ensure their safe operation. A major driver in the safety limits of nuclear reactors is their response to a reactivity initiated accident (RIA). RIAs are design basis accidents that are legally required to be analyzed by 10CFR50. Following a postulated RIA, that may produce a severe power excursion, the reactor must be shown to maintain its pressure boundary and cooling capabilities. One regulatory form this takes is a limit on the energy density following the RIA. In 2007, there was a proposal by the Nuclear Regulatory Commission to tighten the historical acceptance criteria that is currently under review. The ability to accurately model time-dependent reactor physics is crucial for setting and ensuring compliance with these RIA limits.

With ever growing computational resources, the quality of simulation capability has advanced greatly over the last few decades. While nodal methods were considered state-of-the-art 20 years ago [52], full transport codes are now the gold standard. MPACT [38], developed jointly by the University of Michigan and Oak Ridge National Laboratory under the Consortium for Advanced Simulation of LWRs (CASL) program is one such code. MPACT is capable of providing sub-pin power distributions via the 2D/1D method of characteristics (MOC) calculations [31], [11], [12]. The MPACT transient capability was developed for modeling RIAs with an internal thermal-hydraulics model [14] or coupled to the thermal-hydraulics code COBRA-TF [55]. These time-dependent simulations can require significant computational resources with a series of calculations being solved at discrete time steps, usually on the order of milliseconds. The focus of this thesis is on developing an adaptive time stepping (ATS) methodology to automatically adjust the time step size based on the most up-to-date analysis of the system evolution.

The remainder of this chapter will lay out the principles of neutron transport essential to modeling the behavior of a nuclear reactor, including the most commonly employed approximation and eigenvalue formulations. Chapter 2 will then focus on the details of the calculation methodology, with a focus on MPACT-specific algorithms, including 2D/1D, MOC, acceleration, and transient modifications. Chapter 3 will introduce the concept of ATS and propose a general methodology based on limiting the leading order truncation error of

the time discretization through finite difference or α-eigenvalue estimations. Chapter 4 will demonstrate the ability of MPACT to calculate the α-eigenvalue for steady state simulations before turning to its use with ATS. Chapter 5 will explore ATS with the finite difference approach and characterize the performance of various options. Chapter 6 will then investigate ATS with the α-eigenvalue approach. Chapter 7 will apply the ATS methods to a SPERT mini-core and C5G7 benchmark as a final demonstration. Finally, the results and future work will be summarized in Chapter 8.

1.1 Neutron Transport and Approximations

The behavior of individual neutrons is a stochastic process by nature, but the neutron population in a nuclear reactor is large enough that it can be described by a statistical average. This average behavior of neutrons is governed by the time-dependent Boltzmann transport equation and the neutron precursor balance equations as a function of space $\mathbf{r}$, energy E, angle $\mathbf{\Omega}$, and time t:

$$\frac{1}{v(E)}\frac{\partial}{\partial t}\psi(\mathbf{r},\mathbf{\Omega},E,t)+\mathbf{\Omega}\cdot\nabla\psi(\mathbf{r},\mathbf{\Omega},E,t)+\Sigma_t(\mathbf{r},E,t)\psi(\mathbf{r},\mathbf{\Omega},E,t)= \int_0^\infty\int_{4\pi}\Sigma_s(\mathbf{r},E'\rightarrow E,\mathbf{\Omega}\cdot\mathbf{\Omega}',t)\psi(\mathbf{r},\mathbf{\Omega}',E',t)d\Omega' dE' + (1-\beta)\frac{\chi_p(\mathbf{r},E)}{4\pi}\int_0^\infty\int_{4\pi}\nu\Sigma_f(\mathbf{r},E',t)\psi(\mathbf{r},\mathbf{\Omega}',E',t)d\Omega' dE' + \frac{\chi_d(\mathbf{r},E)}{4\pi}\sum_i\lambda_i C_i(\mathbf{r},t) \quad (1.1a)$$

$$\frac{\partial}{\partial t}C_i(\mathbf{r},t)=\beta_i\int_0^\infty\int_{4\pi}\nu\Sigma_f(\mathbf{r},E',t)\psi(\mathbf{r},\mathbf{\Omega}',E',t)d\Omega' dE'-\lambda_i C_i(\mathbf{r},t), \quad (1.1b)$$

with the usual definitions:

$\psi(\mathbf{r},\mathbf{\Omega},E,t)$ – angular neutron flux
$C_i(\mathbf{r},t)$ – delayed neutron precursor density for delayed group i
$\Sigma_t(\mathbf{r},E,t)$ – total cross section (probability of interaction)
$\Sigma_s(\mathbf{r},E'\rightarrow E,\mathbf{\Omega}\cdot\mathbf{\Omega}',t)$ – differential scattering cross section from energy E' to E and angle $\mathbf{\Omega}'$ to $\mathbf{\Omega}$
$\nu(E,t)$ – neutrons produced per fission from neutron with energy E
$\Sigma_f(\mathbf{r},E,t)$ – fission cross section
$\chi(\mathbf{r},E,t)$ – emission spectrum for prompt (p) and delayed (d) neutrons
$\beta_i(\mathbf{r},t)$ – delayed group i faction ($\beta=\sum_i\beta_i$)
$\lambda_i(\mathbf{r},t)$ – decay constant for delayed group i

A rigorous derivation of this equation from first principles can be found in many sources, including [15], but fundamentally, it is a conservation equation with loss terms on the left representing time attenuation, leakage, and attenuation (or removal), and production terms on the right, representing in-scatter, prompt fission, and delayed neutron contributions[1]. Eqs. (1.1) represent a system of 7-dimensional equations. In

[1]When a nucleus undergoes fission, it can break into a number of different smaller nuclei, called fission products. Among these

Cartesian coordinates, the 3-dimensional spatial dependence is given by:

$$\mathbf{r} = (x, y, z). \tag{1.2}$$

The angular dependence is typically represented by a 2-dimensional direction of flight variable given by its azimuthal angle ω and the cosine of polar angle $\mu = \cos\theta$:

$$\mathbf{\Omega} = (\Omega_x, \Omega_y, \Omega_z) = \left(\sqrt{1-\mu^2}\cos\omega, \sqrt{1-\mu^2}\sin\omega, \mu\right), \tag{1.3}$$

where $\mathbf{\Omega}$ is taken to be a unit vector so that $\mathbf{\Omega}\cdot\mathbf{\Omega} = 1$. Energy and time are 1-dimensional variables. When there is no change over time, eqs. (1.1) reduce to the single steady-state equation:

$$\begin{aligned}\mathbf{\Omega}\cdot\nabla\psi(\mathbf{r},\mathbf{\Omega},E) + \Sigma_t(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E) = \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{r},E'\to E,\mathbf{\Omega}\cdot\mathbf{\Omega}')\psi(\mathbf{r},\mathbf{\Omega}',E')d\Omega' dE' \\ + \frac{\chi(\mathbf{r},E)}{4\pi}\int_0^\infty \int_{4\pi} \nu\Sigma_f(\mathbf{r},E')\psi(\mathbf{r},\mathbf{\Omega}',E')d\Omega' dE',\end{aligned} \tag{1.4}$$

where

$$\chi(\mathbf{r},E,t) = (1-\beta(\mathbf{r},t)\chi_p(\mathbf{r},E,t) + \beta(\mathbf{r},t)\chi_d(\mathbf{r},E,t). \tag{1.5}$$

Most derivations and approximations throughout this thesis will begin with the steady-state equation for simplicity unless the time dependence is of specific importance. Also, it is often mathematically convenient to discuss the transport equation in operator notation so that algorithms can be written generically rather than being restricted to a specific implementation. Thus, we can write eq. (1.4) as

$$\mathbf{L}\psi + \mathbf{R}\psi = \mathbf{S}\psi + \mathbf{F}\psi, \tag{1.6}$$

where $\mathbf{L}$ represents the leakage operator, $\mathbf{R}$ represents the removal operator, $\mathbf{S}$ represents the scattering operator, and $\mathbf{F}$ represents the fission operator. From eq. (1.4), these operators are defined as

$$\mathbf{L}\psi = \mathbf{\Omega}\cdot\nabla\psi(\mathbf{r},\mathbf{\Omega},E) \tag{1.7a}$$

$$\mathbf{R}\psi = \Sigma_t(\mathbf{r},E)\psi(\mathbf{r},\mathbf{\Omega},E) \tag{1.7b}$$

$$\mathbf{S}\psi = \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{r},E'\to E,\mathbf{\Omega}\cdot\mathbf{\Omega}')\psi(\mathbf{r},\mathbf{\Omega}',E')d\Omega' dE' \tag{1.7c}$$

fission products, a small number will decay (or their daughter products will decay) by neutron emission. Thus, they provide an extra source of neutrons, albeit delayed from the actual fission event itself. Hence, they are referred to as delayed neutron precursors. Delayed neutrons provide both necessary margin for safe operation of a nuclear reactor and serious challenges for modeling the time dependent behavior. Both of these are results of the rather large discrepancy between the time scales of neutron transport that occurs on the order of micro- or even nanoseconds and delayed neutron emission that occurs on the on scale of milliseconds to tens of seconds. The mathematical description of the precursor behavior, given by eq. (1.1b), is a simple balance of production through fission and decay. Rather than track each delayed neutron precursor explicitly, they are typically divided into a number of groups, most often 1, 6 or 8, with a representative time constant. The concentrations of these i groups are represented by C_i in eq. (1.1b) with decay constant λ_i and fraction of delayed neutrons β_i. The total fraction of delayed neutrons is thus given by $\beta = \sum_i \beta_i$.

$$\mathbf{F}\psi = \frac{\chi(\mathbf{r},E)}{4\pi}\int_0^\infty \int_{4\pi} \nu\Sigma_f(\mathbf{r},E')\psi(\mathbf{r},\mathbf{\Omega'},E')d\Omega' dE'. \tag{1.7d}$$

Note that the definitions in eqs. (1.7) are not strict; this simply provides the fullest expression for consistency with the steady-state transport equation. Simplifications may be made to the operators, such as an isotropic scatting approximation, or they may be modified. For instance, when discussing the α-eigenvalue, it will often be necessary to pull the χ dependency out of the fission operator $\mathbf{F}$.

While the transport equation is a function of the angular flux, $\psi(\mathbf{r},\mathbf{\Omega},E,t)$, out of mathematical necessity to correctly describe neutron leakage and scattering, this is not a quantity of physical interest. Reaction rates quantities are integrated over the angular variable and are therefore more naturally described in terms of the scalar flux, defined by:

$$\phi(\mathbf{r},E,t) = \int_{4\pi} \psi(\mathbf{r},\mathbf{\Omega},E,t)d\Omega \equiv \mathbf{\Gamma}\psi, \tag{1.8}$$

where we have introduced the operator $\mathbf{\Gamma}$ to represent integration of the angular variable over the unit sphere. Since the scalar flux represents the physical quantity of interest and requires less memory for storage than the angular flux, many neutronic calculations attempt to avoid the angular flux whenever possible. This leads to the obvious problem of trying to describe a system without two of the dependent variables, that necessarily introduces approximations. As we will see, there are times when careful approximations can allow the scalar flux to be used in places of the angular flux with limited impact on the overall accuracy.

While eqs. (1.1) provide a thorough neutronic account of a nuclear reactor, they are essentially never used in this general form. For anything but very simple cases, discretization and approximations are employed to simplify at least some of the dependenct variables. We will now explore some of the basic approximations that can be used to simplify eqs. (1.1) to a form more amenable for numerical solution.

1.1.1 The Independent Variables and Discretization

Since general solution of the transport equation is impossible to derive in most practical applications, we naturally seek to solve the problem numerically. There are two fundamentally different paths available to do so– deterministic and stochastic. Deterministic codes attempt to numerically solve some form of eqs. (1.1). This requires some discretization of the problem to reduce it to a set of linear equations that may be solved either directly or iteratively. The discretization choices impose a limit on how accurate a solution may be expected, but should tend toward the exact solution as they become more refined. Stochastic methods, often referred to as Monte Carlo methods, simulate a large number of particles (neutrons) traveling and interacting within the problem geometry. Unlike deterministic methods, this can be accomplished by treating the dependent variables continuously within the constraints of the modeling capabilities, which are adequate for the relatively simple geometric shapes present in most reactors. However, the solution obtained by stochastic methods necessarily has an associated statistical error that is inversely proportional to the square of the number of trials, or particles, simulated. The focus of this research is on deterministic methods, however, so we will leave further discussion of stochastic methods behind. We will continue with the specific considerations for each of the variables for deterministic methods.

A description of the myriad approaches for spatial treatment of the transport equation could easily fill the pages of this dissertation itself, so we will restrict our attention to the basics and details germane to this research. Historically, direct discretization of the geometry is the most commonly employed method to reduce the transport equation to a system of equations. This typically requires some sort of homogenization. In nuclear reactors, the pin cell is the basic unit of geometry. In many of the early reactor analysis codes, and indeed in many modern codes, a homogenized pin cell is therefore also the basic unit of geometry. While advanced homogenization techniques, including corrections such as disadvantage and self-shielding factors and a myriad of other approximations [15], can be employed to make such a homogenization usable, these techniques cannot accurately account for the full spatial transport effects. Despite being unable to capture detailed the flux shape, with careful construction, pin cell averaged calculations can preserve reaction rates, which is the fundamental consistency desired of most approximations and discretizations.

For any traditional spatial discretization, the transport equation is transformed into a system of equations, with one or more equations for each spatial cell. The removal, scattering, and fission terms are all described within a single cell, while the leakage terms link the equations for neighboring cells. Leakage is typically best described by face-centered unknowns, while the other terms are described by cell-centered unknowns. The exact choice of discretization and the closures to relate face- and cell-centered values (e.g. diamond difference, step characteristic, etc.) define the spatial discretization.

The method of characteristics employed by MPACT, discussed more thoroughly in Section 2.2.2, avoids traditional discretization of the spatial variable in favor of tracing rays across the system. This allows the continuous system to be reduced to a set of analytical equations without homogenization of the geometry, in a similar vein as stochastic methods. The acceleration method MPACT employs, detailed in Section 2.2.3 occurs on a coarse mesh, typically a homogenized pin cell, and employs traditional spatial discretization methods. (While a detailed description of MPACT will be developed in Section 2.2, we continue to point out important high level descriptions and features throughout this chapter.)

The energy variable is almost invariably treated with the multigroup approximation in deterministic codes. This is developed in Section 1.1.2.

The angular variable is a frequent target for approximations since the only explicit angular dependence is contained in the the leakage and scattering terms, and its inclusion is more of a mathematical requirement than a physical quantity of interest. The two most common treatments are the discrete ordinates (S_N) and expansion approximations. As the name discrete ordinate implies, the S_N method is a traditional discretization of the direction variable that employs a set of carefully chosen directions and weights to represent the neutron paths of flight. Moment expansion approximations, on the other hand, express the angular flux as a summation of expansion moments, usually chosen chosen such that a specified set of integrations are preserved. Section 1.1.3 will address these treatments in greater detail. Independent of, or in conjunction with, these approximations the angular dependence of the scattering term often receives special treatment. This is discussed more thoroughly in Section 1.1.4.

Lastly, we address the time variable. Much of reactor analysis avoids this variable entirely by focusing on solution of the steady-state equation, rightfully so since the most common method of dealing with time-dependent problems is discretization that reduces the time-dependent problem to a series of steady-state solutions coupled by the time derivative term and delayed neutron source. Unfortunately, this covers a very large range of time scales, leading to a stiff set of equations. This necessitates use of implicit, or backward, time-differencing for stability, which requires solving a linear system at each time point [35].

1.1.2 Multigroup Approximation

Energy is a particularly difficult variable to deal with in the transport equation since 1) it spans a dozen orders of magnitude and 2) the cross sections have a strong and highly irregular dependence on energy, as shown in Figure 1.1. These two facts immediately preclude a straightforward discretization. Instead, the so-called mulitgroup approximation is employed. A maximum cut off energy, typically $E_0 = 20$ MeV, is chosen, and the energy spectrum is divided into some number of groups, G, as shown in Figure 1.2.

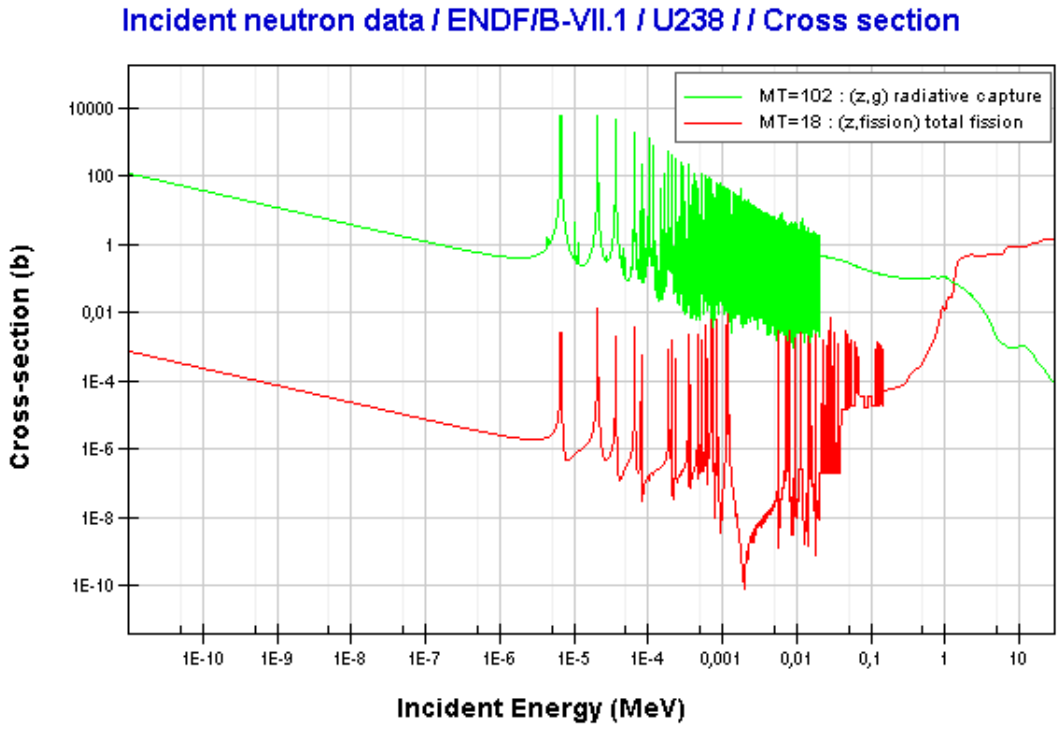

Figure 1.1: Energy Dependence of the Fission and Capture Cross-section for U-238

The steady-state transport equation is then integrated over each energy group $1 \leq g \leq G$, forming a system of G equations coupled through the scattering and fission terms:

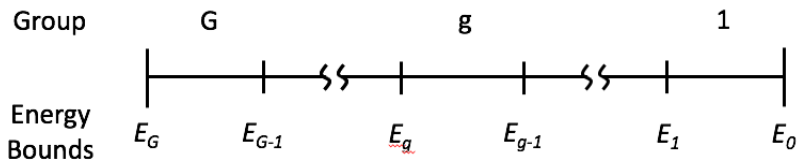

Figure 1.2: Multigroup Approximation of the Energy Spectrum

$$\boldsymbol{\Omega}\cdot\nabla\psi_g(\mathbf{r},\boldsymbol{\Omega})+\Sigma_{t,g}(\mathbf{r})\psi_g(\mathbf{r},\boldsymbol{\Omega})=\sum_{g'=1}^{G}\int_{4\pi}\Sigma_{s,g'\rightarrow g}(\mathbf{r},\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}')\psi_{g'}(\mathbf{r},\boldsymbol{\Omega}')d\Omega' + \frac{\chi_g(\mathbf{r})}{4\pi}\sum_{g'=1}^{G}\int_{4\pi}\nu\Sigma_{f,g'}(\mathbf{r},\boldsymbol{\Omega}')\psi_{g'}(\mathbf{r},\boldsymbol{\Omega}')d\Omega', \quad (1.9)$$

where the multigroup flux, cross sections, and fission spectrum are defined as:

$$\psi_g(\mathbf{r},\boldsymbol{\Omega})=\int_{E_g}^{E_{g-1}}\psi(\mathbf{r},\boldsymbol{\Omega},E)dE \quad (1.10a)$$

$$\Sigma_{x,g}(\mathbf{r})=\frac{\int_{E_g}^{E_{g-1}}\Sigma_x(\mathbf{r},E)\psi(\mathbf{r},\boldsymbol{\Omega},E)dE}{\int_{E_g}^{E_{g-1}}\psi(\mathbf{r},\boldsymbol{\Omega},E)dE} \quad (\Sigma_x=\Sigma_t,\nu\Sigma_f) \quad (1.10b)$$

$$\Sigma_{s,g'\rightarrow g}(\mathbf{r},\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}')=\frac{\int_{E_g}^{E_{g-1}}\int_{E'_g}^{E_{g'-1}}\Sigma_s(\mathbf{r},\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}',E'\rightarrow E)\psi(\mathbf{r},\boldsymbol{\Omega}',E')dE'dE}{\int_{E_g}^{E_{g-1}}\psi(\mathbf{r},\boldsymbol{\Omega},E)dE} \quad (1.10c)$$

$$\chi_g(\mathbf{r})=\int_{E_g}^{E_{g-1}}\chi(\mathbf{r},E)dE \quad (1.10d)$$

While eq. (1.9) is exact, it also requires knowledge of the solution $\psi(\mathbf{r},\boldsymbol{\Omega},E)$ to compute the coefficients in eqs. (1.10). However, an approximation to the solution that sufficiently represents the energy dependence of the actual solution can yield satisfactory coefficients. Such an approximation can be obtained via intuitive means [33] or using an approximate solution. The latter method is employed in MPACT and is known as cross section shielding [38].

1.1.3 Angular Approximations

As previously mentioned, treatment of the angular variable typically falls into two categories: discrete ordinates (S_N) and expansion methods. The method of characteristics employed by MPACT can be considered a specific instance of discrete ordinates.

The S_N equations are formed by restricting the direction of flight variable to a discrete set of directions. As discussed in Section 1.1, the angular flux is a mathematical necessity, while the scalar flux is the physical quantity of interest. This requires that the integral over the unit sphere be preserved. This can be

accomplished via quadrature sets that consist of M direction and weight pairings such that:

$$\int_{4\pi} \psi(\mathbf{\Omega}) d\Omega = \sum_{m=1}^{M} w_m \psi(\mathbf{\Omega}_m) \tag{1.11}$$

is satisfied exactly for a polynomial ψ of up to order N. A variety of quadrature sets exist to satisfy specific limits and weightings of the integral in eq. (1.11). While a fascinating topic in its own right, we shall simply accept the existence of such quadrature sets for the purposes of this thesis. Note that the angular variable $\mathbf{\Omega}$ can be broken into its component azimuthal ω and polar μ components and a separate discretization and quadrature can be applied to each so that eq. (1.11) becomes

$$\sum_{m=1}^{M} w_m \psi(\mathbf{\Omega}_m) \rightarrow \sum_{\ell=1}^{L} \sum_{m=1}^{M} w_\ell w_m \psi(\omega_\ell, \mu_m). \tag{1.12}$$

This is commonly referred to as "product quadrature" and is employed by MPACT.

Expansion methods comprise the other major approach to treatment of the angular variable. Rather than discretize the angular variable directly, it is expanded in terms of the spherical harmonic functions, that are a set of orthogonal functions defined on the unit sphere:

$$\psi(\mathbf{r}, \mathbf{\Omega}, E, t) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \psi_{\ell m}(\mathbf{r}, E, t) Y_{\ell m}(\mathbf{\Omega}). \tag{1.13}$$

The series is truncated in ℓ and substituted into the transport equation. The result is then multiplied by the various $Y_{lm}(\mathbf{\Omega})$, and integrated over angle to produce a set of coupled equations that can be solved numerically. An extremely important approximation to the transport equation, the diffusion equation, is perhaps best motivated by a special instance of spherical harmonics– the P_1 equations that arise from the 1D version of the transport equation. In this case, the complicated spherical harmonic functions of degree N reduce to the simpler Legendre polynomials, denoted by $P_\ell(\mu)$:

$$\psi(x, \mu, E, t) = \sum_{\ell=0}^{N} \frac{2\ell + 1}{2} \psi_\ell(x, E, t) P_\ell(\mu). \tag{1.14}$$

1.1.4 Scattering Approximations

The scattering term is a frequent target of approximations, particularly since the differential scattering cross section, $\Sigma_s(\mathbf{r}, E' \rightarrow E, \mathbf{\Omega} \cdot \mathbf{\Omega}', t)$, only depends on the cosine of the scattering angle, $\mu_0 = \mathbf{\Omega} \cdot \mathbf{\Omega}'$, rather than on the angular variable itself. The most basic simplification that can be made is to assume that scattering is isotropic:

$$\int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{r}, E' \rightarrow E, \mathbf{\Omega} \cdot \mathbf{\Omega}', t) \psi(\mathbf{r}, \mathbf{\Omega}', E', t) d\Omega' dE' \rightarrow$$
$$\int_0^\infty \int_{4\pi} \frac{\Sigma_s(\mathbf{r}, E' \rightarrow E, t)}{4\pi} \psi(\mathbf{r}, \mathbf{\Omega}', E', t) d\Omega' dE' = \frac{1}{4\pi} \int_0^\infty \Sigma_s(\mathbf{r}, E' \rightarrow E, t) \phi(\mathbf{r}, E', t) dE'. \tag{1.15}$$

This approximation breaks down when there is significant forward scattering, as is the case in collisions with low mass atoms like hydrogen. Unfortunately, hydrogen scattering is the major source of neutron thermalization in nuclear reactors. Hence, we should not expect an isotropic scattering approximation to provide a good description of neutron transport in a nuclear reactor. However, we can apply a transport correction to the isotropic approximation that greatly improves its accuracy.

Such a correction may be obtained from the 1D monoenergetic P_1 equations where

$$\Sigma_{s0} = \int_{4\pi} \Sigma(x, \mu_0) d\mu_0 = \Sigma_s(x), \tag{1.16a}$$

$$\Sigma_{s1} = \int_{4\pi} \mu_0 \Sigma(x, \mu_0) d\mu_0 = \bar{\mu_0} \Sigma_s(x) \tag{1.16b}$$

are the differential scattering cross sections for the the zeroth and first angular moments. Note that under assumptions of isotropic elastic scattering in the center-of-mass system with low energy neutrons, μ_0 may be approximated as $2/3A$, where A is the atomic mass number of the scattering nuclei [33]. While more advanced treatments may be employed, the important point is that Σ_{s1} is a known coefficient. This result can be generalized for the 3D multigroup problem. In MPACT, transport-corrected isotropic scattering is utilized to approximate the linearly anisotropic scattering operator, with the outscatter correction for heavy elements, the inscatter correction for light elements, and the neutron leakage conservation correction for hydrogen [38] [56].

1.1.5 Diffusion Approximation

The most commonly employed approximation to the transport equation is the diffusion equation, which eliminates the angular dependence. To derive the diffusion approximation, operate on eq. (1.4) by $\int_{4\pi}(\cdot)d\Omega$. The removal and fission terms simplify in a straightforward manner to dependence on the scalar flux, while the leakage and scattering terms require more careful treatment. The leakage term depends on the first angular moment of the flux, the current $J = \int_{4\pi} \mathbf{\Omega}\psi d\Omega$. The simplest approach is to approximate the current with Fick's Law

$$J(\mathbf{r}, E) = -D(\mathbf{r}, E)\nabla\phi(\mathbf{r}, E). \tag{1.17}$$

Of course we have now introduced a diffusion coefficient D for which we have no definition. If we obtain the first angular moment of the transport equation by operating on (1.4) with $\int_{4\pi} \mathbf{\Omega}(\cdot)d\Omega$ and employ the P_1 approximation as in Section 1.1.4, we can obtain an expression for the diffusion coefficient:

$$D(\mathbf{r}, E) = \frac{1}{3\Sigma_{tr}(\mathbf{r}, E)} = \frac{1}{3(\Sigma_t(\mathbf{r}, E) - \Sigma_{s1}(\mathbf{r}, E))}, \tag{1.18}$$

where Σ_{tr} is the so-called transport-corrected total cross section. Employing Σ_{tr} instead of Σ_t allows for the scattering source to be treated isotropically while preserving some of the linear dependence on the angular variable. While a purely isotropic approximation cannot sufficiently describe typical reactor physics, this transport-corrected isotropic approximation is often adequate. The resulting diffusion equation is

$$-\nabla\cdot D(\mathbf{r},E)\nabla\phi(\mathbf{r},E)+\Sigma_t(\mathbf{r},E)\phi(\mathbf{r},E)=\int_0^\infty \Sigma_{s0}(\mathbf{r},E'\rightarrow E)\phi(\mathbf{r},E')dE'+\chi(\mathbf{r},E)\int_0^\infty \nu\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E')dE'. \tag{1.19}$$

The multigroup formulation of the diffusion equation is then

$$-\nabla\cdot D_g(\mathbf{r})\nabla\phi_g(\mathbf{r})+\Sigma_{t,g}(\mathbf{r})\phi_g(\mathbf{r})=\sum_{g'=1}^{G}\Sigma_{s0,g'\rightarrow g}(\mathbf{r})\phi_g(\mathbf{r})+\frac{\chi_g(\mathbf{r})}{k}\sum_{g'=1}^{G}\nu\Sigma_{f,g}(\mathbf{r})\phi_g(\mathbf{r}). \tag{1.20}$$

The diffusion equation is not exact and in general should not be expected to be consistent with the transport equation. The diffusion equation is valid only when the angular flux has weak (linear) dependence on the angular variable. In general, this applies to highly scattering systems without sharp discontinuities in material properties. In other words, the diffusion equation is accurate in the moderator, questionable in the presence of strong absorbers, and inaccurate near the system boundaries. Despite these short-comings, the diffusion equation does provide a reasonable approximation to the the transport solution in many cases without requiring the angular flux. Since the scalar flux is generally the quantity of interest for determining reaction rates, this reduction in dependent variables (and hence computational complexity) is much desired.

1.2 Eigenvalue Formulations of the Transport Equation

Another class of approximations to the transport equation is obtained by converting eq. (1.1) to a steady-state eigenvalue problem. The resulting equations are similar in form to (1.4) with the introduction of an eigenvalue. While the eigenvalues for any given formulation exist in a discrete or even continuous spectrum, there is typically a single eigenvalue of interest that provides some physical insight to the system. Discussion will generally be focused on this fundamental eigenvalue rather than the complete eigenspectrum.

1.2.1 k-eigenvalue

Of most interest to nuclear reactors is the k-eigenvalue problem. In this formulation, the fission source in eq. (1.4) is scaled by $\frac{1}{k}$:

$$\begin{aligned}\boldsymbol{\Omega}\cdot\nabla\psi(\mathbf{r},\boldsymbol{\Omega},E,t)+\Sigma_t(\mathbf{r},E,t)\psi(\mathbf{r},\boldsymbol{\Omega},E,t)=&\int_0^\infty\int_{4\pi}\Sigma_s(\mathbf{r},E'\rightarrow E,\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}',t)\psi(\mathbf{r},\boldsymbol{\Omega}',E',t)d\Omega'dE'\\&+\frac{1}{k}\frac{\chi(\mathbf{r},E)}{4\pi}\int_0^\infty\int_{4\pi}\nu\Sigma_f(\mathbf{r},E',t)\psi(\mathbf{r},\boldsymbol{\Omega}',E',t)d\Omega'dE'.\end{aligned} \tag{1.21}$$

Or in operator notation

$$\mathbf{L}\psi+\mathbf{R}\psi=\mathbf{S}\psi+\frac{1}{k}\mathbf{F}\psi. \tag{1.22}$$

The k-eigenvalue is referred to as the effective multiplication factor and is often thought of as the ratio of neutrons in one generation to the the previous generation [15]. Physically, the dominant k-eigenvalue characterizes the criticality of the system.

$k < 1$ implies a subcritical system with greater neutron loss than gain
$k = 1$ implies a critical system with no change in the neutron population over time, i.e. eq. (1.4) is satisfied.
$k > 1$ implies a supercritical system with greater neutron gain than loss

Mathematically, this eigenvalue enables the steady-state problem to be satisfied for systems where $k \neq 1$. That is, we have replaced the time dependence with the k-eigenvalue that artificially reduces the fission source when $k > 1$ and amplifies the fission source when $k < 1$ in compensation. Thus, the solution of eq. (1.21) is an approximation to the instantaneous solution of eq. (1.1), achieving equality only when there is no time dependence. However, for nearly critical systems, this approximation is sufficient for preserving measurable quantities like reaction rates.

The k-eigenspectrum for this problem is well-studied and characterized, such as in [15], [10], [48], and many others. The eigenvalues are all positive and real, and a dominant eigenvalue exists. That is, there is a single eigenvalue with a magnitude greater than all the other eigenvalues. This dominant eigenvalue is the eigenvalue of interest for the k-eigenvalue problem and the corresponding eigenvector is considered the fundamental mode of the solution.

1.2.2 α-eigenvalue

The α-eigenvalue problem is derived by approximating the time dependence of the angular flux with an exponential form:

$$\psi(\mathbf{r}, \mathbf{\Omega}, E, t) \approx \psi(\mathbf{r}, \mathbf{\Omega}, E) e^{\alpha t} \quad \Rightarrow \quad \frac{1}{v(E)} \frac{\partial \psi(\mathbf{r}, \mathbf{\Omega}, E, t)}{\partial t} \approx \frac{\alpha}{v(E)} \psi(\mathbf{r}, \mathbf{\Omega}, E) \tag{1.23}$$

Inserting the approximation of eq. (1.23) into eq. (1.1a) yields

$$\begin{aligned} \frac{\alpha}{v(E)} \psi(\mathbf{r}, \mathbf{\Omega}, E) + \mathbf{\Omega} \cdot \nabla \psi(\mathbf{r}, \mathbf{\Omega}, E) + \Sigma_t(\mathbf{r}, E) \psi(\mathbf{r}, \mathbf{\Omega}, E) = \\ \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{r}, E' \to E, \mathbf{\Omega} \cdot \mathbf{\Omega}') \psi(\mathbf{r}, \mathbf{\Omega}', E') d\Omega' dE' \\ + \frac{\chi_p(\mathbf{r}, E)(1 - \beta(\mathbf{r}))}{4\pi} \int_0^\infty \int_{4\pi} \nu \Sigma_f(\mathbf{r}, E') \psi(\mathbf{r}, \mathbf{\Omega}', E') d\Omega' dE' + \frac{\chi_d(\mathbf{r}, E)}{4\pi} \sum_i \lambda_i C(\mathbf{r}) \end{aligned} \tag{1.24}$$

In operator notation, we introduce the diagonal operator $\mathbf{V}$ that consists of multiplication by velocity and the vector D to represent the delayed neutron contribution to complete the α-eigenvalue formulation.

$$\alpha \mathbf{V}^{-1} \psi + \mathbf{L}\psi + \mathbf{R}\psi = \mathbf{S}\psi + \mathbf{F}_p \psi + D, \tag{1.25}$$

where we have added the subscript p to the fission operator to denote that it is the prompt fission source only. Similar to the k-eigenvalue, the fundamental α-eigenvalue can be used to characterize the criticality of the system.

$\alpha < 0$ implies a subcritical system with greater neutron loss than gain
$\alpha = 0$ implies a critical system with no change in the solution over time, i.e. eq. (1.4) is satisfied.
$\alpha > 0$ implies a supercritical system with greater neutron gain than loss

For critical configurations, eq. (1.24) reduces to eqs. (1.4). A notable difference between the k-eigenvalue eq. (1.21) and α-eigenvalue eq. (1.24) is immediately apparent– the α-eigenvalue problem has an inhomogeneous term representing the delayed neutron contribution.

Less obvious are the significant differences in the k- and α-eigenspectra. While the spectral properties of the α-eigenvalue are not the focus of this thesis, they will be important when considering calculation methods in Chapters 2 and 4. A number of studies have been undertaken to understand the α-eigenspectrum from a theoretical standpoint. In reference [32], Larsen and Zweifel provide an excellent summary (despite their explicit insistence to the contrary) and expansion of this work. Much of the early investigations focused on specific geometries and scattering treatments, while attempts to extend to more general cases encountered difficulties. Nonetheless, many of the specific results obtained appear to be generalizable, if not by formal mathematics, then at least by experiment. We will summarize the most important points, making sure to clarify.

Generally, the α-eigenspectrum exists as points, lines, and a continuum [4], as shown in the example spectrum provided by Betzler [5] in Figure 1.3. However, with the introduction of two reasonable restrictions– that velocity cannot be zero and that there are no infinite streaming paths– Jörgens has shown that only the point spectrum is present [28]. Since neutron velocity obeys a Maxwellian distribution at low energies, the zero velocity assumption is acceptable, and the restriction to finite streaming paths is consistent with any physical system. Lehner and Wing showed that the point spectrum may not exist in the continuum, and hence there is a lower bound on the point spectrum [34]. Mika established that this lower bound is equal to the minimum value of $v\Sigma_t$. [37]

As with the k-eigenvalue, we are concerned with the fundamental (right-most) α-eigenvalue. The fundamental mode is shown to exist and be real by Nelkin for a thermal neutron distribution in a sufficiently large sphere [39]. However, the higher order α-eigenvalues may be complex. Additionally, the fundamental α-eigenvalue need not be, and in fact rarely is, the largest magnitude eigenvalue. This has important ramifications for the available solution methodology.

Moving on to more practical matters, the inhomogeneous delayed neutron contribution term in eq. (1.24) can be dealt with in a variety of ways. If the precursor densities C_i are known, then no additional assumptions are needed. If this is not the case, then additional assumptions are needed. We will note two such assumptions. The first is that steady-state precursor densities are valid. This is equivalent to setting the time derivative to zero in the precursor balance equation, eq. (1.1b). This allows for the precursor densities to be solved for and upon substitution into eq. (1.24), we obtain the equation

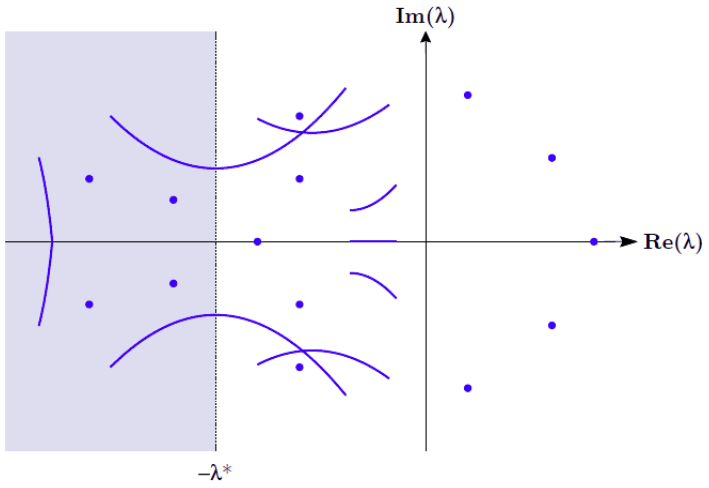

Figure 1.3: Example Spectrum for the α-eigenvalue Showing Points, Lines, and Continuum [4]

$$\frac{\alpha_p}{v(E)}\psi(\mathbf{r},\mathbf{\Omega},E,t)+\mathbf{\Omega}\cdot\nabla\psi(\mathbf{r},\mathbf{\Omega},E,t)+\Sigma_t(\mathbf{r},E,t)\psi(\mathbf{r},\mathbf{\Omega},E,t)=$$
$$\int_0^\infty\int_{4\pi}\Sigma_s(\mathbf{r},E'\to E,\mathbf{\Omega}\cdot\mathbf{\Omega}',t)\psi(\mathbf{r},\mathbf{\Omega}',E',t)d\Omega' dE'$$
$$+\frac{\chi(\mathbf{r},E)}{4\pi}\int_0^\infty\int_{4\pi}\nu\Sigma_f(\mathbf{r},E',t)\psi(\mathbf{r},\mathbf{\Omega}',E',t)d\Omega' dE'. \quad (1.26)$$

Note that eq. (1.26) is precisely the steady state eq. (1.4) with an additional α/v term. The conditions of this derivation are fulfilled immediately after a transient has been initiated from steady-state, and corresponds to the prompt jump in point kinetics parlance. Hence the subscript p on the eigenvalue in eq. (1.26) to represent the prompt α-eigenvalue. Another possibility is assuming that the precursors have the same time dependence as the angular flux:

$$C_i(\mathbf{r},t)\approx C_i(\mathbf{r})e^{\alpha t}\quad\Rightarrow\quad\frac{\partial C_i(\mathbf{r},t)}{\partial t}\approx\alpha C_i(\mathbf{r}). \quad (1.27)$$

By introducing this approximation to the precursor balance eq. (1.1b) and eliminating $C_i(\mathbf{r},t)$ from eq. (1.1a), we obtain:

$$\frac{\alpha_a}{v(E)}\psi(\mathbf{r},\mathbf{\Omega},E,t)+\mathbf{\Omega}\cdot\nabla\psi(\mathbf{r},\mathbf{\Omega},E,t)+\Sigma_t(\mathbf{r},E,t)\psi(\mathbf{r},\mathbf{\Omega},E,t)=$$
$$\int_0^\infty\int_{4\pi}\Sigma_s(\mathbf{r},E'\to E,\mathbf{\Omega}\cdot\mathbf{\Omega}',t)\psi(\mathbf{r},\mathbf{\Omega}',E',t)d\Omega' dE'$$
$$+\frac{\chi_\alpha(\mathbf{r},E)}{4\pi}\int_0^\infty\int_{4\pi}\nu\Sigma_f(\mathbf{r},E',t)\psi(\mathbf{r},\mathbf{\Omega}',E',t)d\Omega' dE' \quad (1.28)$$

where

$$\chi_\alpha(\mathbf{r},E) = (1-\beta)\chi_p(\mathbf{r},E) + \left(\sum_i \frac{\beta_i}{\alpha_a/\lambda_i + 1}\right)\chi_d(\mathbf{r},E)$$
$$= \chi(\mathbf{r},E) - \chi_d(\mathbf{r},E)\sum_i \frac{\alpha_a\beta_i}{\alpha_a+\lambda_i}. \quad (1.29)$$

Unlike the prompt α-eigenvalue, this formulation is valid in the asymptotic equilibrium limit, which is why we have introduced the subscript a for this formulation of the eigenvalue.

1.3 Time Dependent Neutron Transport and Reactor Kinetics

We now turn our attention to the specifics of time-dependent neutron transport. While steady-state calculations have the luxury of eliminating the precursor balance equations, time-dependent calculations must account for this phenomenon in addition to the extra variable.

1.3.1 Point Kinetics

Perhaps the most useful approximation for exploring the time dependent behavior of nuclear reactors are the Point Kinetics Equations (PKE). Multiple forms and derivations exist in the literature [15] [41] [33], but the general idea is to assume separability of time and the remaining dimensional space and reduce eqs. (1.1) to a system dependent on only one variable, time. We will present the form utilized in this work referred to as the Exact Point Kinetics Equations (EPKE) [21] [16] [57]. The diffusion equations, eq. (1.19) are multiplied by the adjoint of the scalar flux, ϕ^*, that need be calculated before commencing the transient, and the resulting equations are integrated over energy and space. This operation can be represented by the inner product:

$$< \phi^*(\mathbf{r},E) f(\mathbf{r},E,t) >= \int_{\mathbf{r}\in V}\int_0^\infty \phi^*(\mathbf{r},E) f(\mathbf{r},E,t) dE d\mathbf{r} \quad (1.30)$$

The resulting EPKE has the form:

$$\frac{dp}{dt} = \frac{\rho(t)-\beta_{eff}(t)}{\Lambda(t)}p(t) + \frac{1}{\Lambda(0)}\sum_i \lambda_i(t)\zeta_i(t) \quad (1.31a)$$

$$\frac{d\zeta_i}{dt} = \frac{\Lambda(0)}{\Lambda(t)}\beta_i p(t) - \lambda_i(t)\zeta_i(t) \quad (1.31b)$$

where $p(t)$ is the amplitude function and $\zeta_i(t)$ is the adjoint weighted precursor concentration. The reactivity, delayed neutron fraction, prompt generation time, and delayed neutron precursor decay constants are calculated, respectively, as:

$$\rho(t) = \frac{< \phi^*(\mathbf{r},E), (\mathbf{F}-\mathbf{L}-\mathbf{R})\phi(\mathbf{r},E,t) >}{< \phi^*(\mathbf{r},E), \mathbf{F}\phi(\mathbf{r},E,t) >}, \quad (1.32a)$$

$$\beta_{eff}(t) = \frac{< \phi^*(\mathbf{r},E), \chi_d(\mathbf{r},E)\beta(\mathbf{r})\mathbf{F}\phi(\mathbf{r},E,t) >}{< \phi^*(\mathbf{r},E), \mathbf{F}\phi(\mathbf{r},E,t) >}, \quad (1.32b)$$

$$\Lambda(t) = \frac{< \phi^*(\mathbf{r}, E), \frac{1}{v(E)}\phi(\mathbf{r}, E, t) >}{< \phi^*(\mathbf{r}, E), \mathbf{F}\phi(\mathbf{r}, E, t) >}, \tag{1.32c}$$

$$\lambda_i(t) = \frac{< \phi^*(\mathbf{r}, E), \lambda_i \chi_d(\mathbf{r}, E) C_i(\mathbf{r}, t) >}{< \phi^*(\mathbf{r}, E), \chi_d(\mathbf{r}, E) C_i(\mathbf{r}, t) >}, \tag{1.32d}$$

$$\zeta_i(t) =< \phi^*(\mathbf{r}, E), C_i(\mathbf{r}, E, t) > \tag{1.32e}$$

The EPKEs require knowledge of the flux to calculate the coefficients, but then reduce to a simple set of ordinary differential equations if the time dependence of the constants is ignored. On a short time scale, this is a reasonable constraint and allows calculation of the overall core amplitude as a function of time.

Chapter 2

Calculation Methodology

In this chapter we will introduce some of the fundamental methodology for deterministically solving the neutron transport equation. We begin with source iteration, the most commonly employed scheme. Next we will introduce a Jacobian-Free Newton-Krylov methodology based around the source iteration architecture. After these general discussions, we will move on to the specifics of the MPACT methodology that will be important for the remainder of the work in this thesis.

2.1 Basic Solution Algorithm

2.1.1 Source Iteration

The workhorse of many neutronic codes is a variation on power iteration known as source iteration (SI). Power/source iteration are a fixed-point, or Picard, iteration where the solution u^* satisfies the relation

$$u^* = f(u^*). \tag{2.1}$$

Power iteration is a two step iterative method to solve the basic eigenvalue problem $\mathbf{A}x = \lambda x$ for the dominant eigenpair; that is the largest magnitude eigenvalue λ and associated eigenvector x. In the first step, the operator $\mathbf{A}$ is applied to some guess of the eigenvector to produce an update to the eigenvector. In the second step, the Rayleigh quotient or a similar equation is used to update the guess of the eigenvalue. The two steps together form the operator f in eq. (2.1) with the solution $u^* = (x, \lambda)$. SI is a generalization of power iteration to solve the generalized eigenvalue problem $\mathbf{A}x = \lambda \mathbf{B}x$. In its most basic form, the two step process then becomes:

$$x^n = \frac{1}{\lambda^{n-1}} \mathbf{B}^{-1} \mathbf{A} x^{n-1}, \tag{2.2a}$$

$$\lambda^n = \lambda^{n-1} \frac{\|\mathbf{A}x^n\|}{\|\mathbf{A}x^{n-1}\|}, \tag{2.2b}$$

where the eigenvalue update equation is derived by requiring $\mathbf{A}x^n = \lambda^n \mathbf{B}x^n$ to be satisfied and noting that the eigenvector update can be rearranged to yield the identity $\mathbf{B}x^n = \frac{1}{\lambda^{n-1}} \mathbf{A}x^{n-1}$. For the k-eigenvalue

problem, note that $\mathbf{A} = \mathbf{F}$ and $\mathbf{B} = \mathbf{L} + \mathbf{R} - \mathbf{S}$[1] and the eigenvector is the angular flux[2] ψ (or scalar flux ϕ in the case of the diffusion equation). The immediate implications of this are that the eigenvector update requires an inversion of the form $(\mathbf{L} + \mathbf{R} - \mathbf{S})^{-1}$, and that the eigenvalue update only requires the action of the fission operator $\mathbf{F}$ on the updated flux. Since the discretization of a nuclear reactor in space, direction, and energy produces a very large system of equations, the inversion drives the computational cost of the solution method. Typically, this cannot be accomplished through a direct method, such as Guassian elimination, and requires some iteration scheme. This inner iteration does not provide an exact inversion, but it can produce a close approximation to the solution at a significantly lower computational cost. A wide variety of methods exist for iteratively computing the inversion; these are commonly referred to as transport sweeps.

There are two less obvious point that warrant discussion as well. The first is that the eigenvector update eq. (2.2a) need not adhere to this exact form. The only requirements are that it be consistent with the eigenvalue equation being solved and that it produce a suitable update to the eigenvector estimate. So for the k-eigenvalue problem, it is perfectly acceptable to substitute the standard update $\psi^n = (\mathbf{L} + \mathbf{R} - \mathbf{S})^{-1} \frac{1}{k^{n-1}} \mathbf{F}\psi^{n-1}$ with an alternative such as $\psi^n = (\mathbf{L} + \mathbf{R})^{-1} \left(\mathbf{S} + \frac{1}{k^{n-1}} \mathbf{F}\right) \psi^{n-1}$. The eigenvalue update eq. (2.2b) does not require modification to accommodate this change as it only requires a suitable estimate of the eigenvector.

The other point to consider is the convergence of SI. For a steady-state, one-group isotropically scattering, fixed source problem in planar geometry with and initial guess of $\psi^0 = 0$, the angular flux ψ^n generated by n applications of SI can easily be shown to represent the angular flux contribution of particles that have scattered up to $n - 1$ times [1]. This physical insight leads to the observation that for scattering-dominated problems, SI will converge slowly. For problems that are leakage- or absorption-dominated, however, SI will converge rapidly. Unfortunately, a typical nuclear reactor falls into the former category, necessitating some sort of acceleration.

2.1.2 Jacobian-Free Newton-Krylov Method

While SI is well-suited for calculating the k-eigenvalue, its utility and/or efficiency for calculating the α eigenvalue is questionable, owing to the spectral properties of the α eigenvalue problem. Therefore, we will also explore a fundamentally different solution methodology within MPACT. Newton's method is an alternative to fixed-point iteration that is used to solve the equation $\mathbf{G}u = 0$. This is accomplished via a two step process:

$$\mathbf{J}(u^{n-1})\delta u^n = -\mathbf{G}u^{n-1} \tag{2.3a}$$

$$u^n = u^{n-1} + \delta u^n \tag{2.3b}$$

where eq. (2.3a) is a linear equation employing the Jacobian, $\mathbf{J}$, as a linear approximation to $\mathbf{G}$ to solve for the Newton correction δu^n. The overwhelming difficulty in this approach is forming the Jacobian and solving

[1] Often with the k-eigenvalue problem λ is used to represent $1/k$. For this discussion we have eschewed this convention to marry the SI methodology with the usual mathematical eigenvalue definitions.

[2] Throughout this thesis we will often refer to the eigenvector, which the reader may think of as the angular or scalar flux.

the linear equation. For many applications, including neutron transport, explicitly forming the Jacobian is either impossible or impractical, which is perhaps one of the main reasons fixed-point iteration methods have been the primary approach for most reactor physics codes. However, interest in Newton's method has been rekindled recently by employing the Jacobian-Free Netwon-Krylov (JFNK) variation of the traditional implementation. Knoll and Keyes's seminal review of JFNK approaches and applications [29] has been a significant driver of this resurgence, and provides an excellent overview of the overall methodology. By using a Krylov method to solve the linear equation in (2.3a), such as GMRES [46], we only require *the action* of the Jacobian on an arbitrary vector rather than full knowledge of the entries of the Jacobian. By only requiring the product $\mathbf{J}v$ for arbitrary v, formation of the Jacobian about u may be avoided by employing the finite difference approximation

$$\mathbf{J}(u)v \approx \frac{1}{\epsilon}\left[\mathbf{G}(u+\epsilon v) - \mathbf{G}(u)\right] \tag{2.4}$$

where ϵ is the perturbation parameter, typically chosen as $\sqrt{\epsilon_{machine}}$ per ref. [29].

Unlike for SI, the convergence of Newton's method is expected to be quadratic, so we expect JFNK to show significant speedup over SI. This comes at the cost of additional storage in the form of the orthonormal basis formed during the Krylov iteration. With GMRES, for example, each iteration produces a new solution (fine mesh multigroup scalar flux, boundary conditions, and eigenvalue) that must be stored.

2.2 MPACT Methodology

MPACT is a method of characteristics (MOC) based 2D/1D neutron transport code developed jointly by the University of Michigan and Oak Ridge National Laboratory under the Department of Energy's Consortium for Advanced Simulation of Light Water Reactors (CASL) [38]. MPACT is capable of solving both the steady-state k-eigenvalue problem and time-dependent problems. SI is the algorithm underlying the MPACT solution methodology, though the implementation is significantly more complex than the basic SI methodology outlined in section 2.1.1.

MPACT employs the multigroup method to treat energy and the S_N approximation with a product quadrature to discretize direction, while employing the method of characteristics (MOC) to allow for an exact representation of basic geometrical structures. Thus, the discretized transport equation MPACT attempts to solve is:

$$\frac{1}{v_g}\frac{\partial \psi_{m,\ell,g}(\mathbf{r},t)}{\partial t} + \mathbf{\Omega}_{m,\ell}\cdot\nabla\psi_{m,\ell,g}(\mathbf{r},t) + \Sigma_{t,g}(\mathbf{r},t)\psi_{m,\ell,g}(\mathbf{r},t) =$$
$$\sum_{g'}^{G}\sum_{\ell'}^{L}\sum_{m'}^{M}\Sigma_{s,g'\rightarrow g}(\mathbf{r},\mathbf{\Omega}_{m',\ell'}\cdot\mathbf{\Omega}_{m,\ell},t)\psi_{m',\ell',g'}(\mathbf{r},t)w_{m'}w_{\ell'}$$
$$+\frac{\chi_g(\mathbf{r},t)}{4\pi}\sum_{g'}^{G}\sum_{\ell'}^{L}\sum_{m'}^{M}\nu\Sigma_{f,g'}(\mathbf{r},t)\psi_{m',\ell',g'}(\mathbf{r},t)w_{m'}w_{\ell'}$$
$$\mathbf{r}\in V,\quad 1\le m\le M,\quad 1\le \ell\le L,\quad 1\le g\le G,\quad (2.5)$$

with M azimuthal angles, L polar angles, and G energy groups.

2.2.1 2D/1D Formulation

One of the fundamental approximations made within MPACT is a separation of the radial (x, y) and axial (z) solution methods known as the 2D/1D formulation, that takes advantage of the fact that nuclear reactors are more homogenous in the axial direction than the radial direction. The solution in the radial and axial directions is thus solved to a level commensurate with the geometrical complexity. The idea was first introduced by two different groups in Korea [11] [12] and was used as the basis of the DeCART code, from which MPACT evolved.

The details of the 2D/1D formulation are laid out in the MPACT manual [38], but we will summarize the basic idea and implications here. A 2D version of the transport equation is created by defining an axial leakage term to represent the z-directional derivative and the resulting planar equation is integrated over an axial slice. Similarly, a 1D version of the transport equation is created by defining a radial leakage term to represent the x- and y-directional derivatives, that is then integrated over a coarse cell (typically a pin cell) in the radial direction. However, instead of solving the resulting transport equation in the z direction, a simplified form of the equation is solved, most typically the P_3 equation (though numerous other options are available in MPACT). The radial and axial equations are then coupled via the leakage terms. Thus in the radial direction eq. (2.5) becomes

$$\frac{1}{v_g}\frac{\partial \psi_{k,m,g}(x,y,t)}{\partial t} + \Omega_{m,x}\frac{\partial \psi_{k,m,g}(x,y,t)}{\partial x} + \Omega_{m,y}\frac{\partial \psi_{m,g}(x,y,t)}{\partial y} + \Sigma_{t,k,g}(x,y,t)\psi_{k.m,g}(x,y,t) =$$
$$\sum_{g'}^{G}\sum_{m'}^{M}\Sigma_{s,k,g'\rightarrow g}(x,y,\mathbf{\Omega}_{m'}\cdot\mathbf{\Omega}_m,t)\psi_{k,m',g'}(x,y,t)w_{m'}+$$
$$\frac{\chi_{k,g}(x,y,t)}{4\pi}\sum_{g'}^{G}\sum_{m'}^{M}\nu\Sigma_{f,k,g'}(x,y,t)\psi_{k,m',g'}(x,y,t)w_{m'}-$$
$$\frac{1}{4\pi\Delta z_k}\left[J_{k+1/2,g}(x,y,t)-J_{k-1/2,g}(x,y,t)\right],\quad (2.6)$$

where $J_{k\pm1/2}$ are the isotropic axial leakage through the top and bottom of plane k. While the P_3 formulation

for the axial direction is typically employed in MPACT, we present the P_1 axial equation for better clarity:

$$\frac{1}{v_g}\frac{\partial \phi_g^{XY}(z,t)}{\partial t} - \frac{\partial}{\partial z} D_g^{XY}(z,t)\frac{\partial \phi_g^{XY}(z,t)}{\partial z} + \Sigma_{t,g}^{XY}(z,t)\phi_g^{XY}(z,t) = \sum_{g'}^{G} \Sigma_{s,g'\to g}(z,t)^{XY}\phi_{g'}^{XY}(z,t) + \chi_g^{XY}(z,t)\sum_{g'}^{G}\nu\Sigma_{s,g'}(z,t)^{XY}\phi_{g'}^{XY}(z,t) - TL_g^{XY}(z,t) \quad (2.7a)$$

$$TL_g^{XY}(z,t) = \frac{1}{\Delta x}\left[J_{x+1/2,g}(z,t) - J_{x-1/2,g}(z,t)\right] + \frac{1}{\Delta y}\left[J_{y+1/2,g}(z,t) - J_{y-1/2,g}(z,t)\right], \quad (2.7b)$$

where J again represents this leakage, but this time in the transverse x and y directions. Details of the derivation can be found in Chapters 4 and 6 of the MPACT manual [38], but we will note that maintaining consistency between the solutions is the main concern for the precise formulation of the leakage terms.

2.2.2 Method of Characteristics

The power of MPACT lies in the MOC methodology used to provide the eigenvector update. As detailed in Section 2.1.1, the eigenvector update requires an inversion of operators, see eq. (2.2a). Rather than explicitly invert the matrix, however, MOC relies upon the fact that the eq. (2.6) can be solved analytically along a characteristic, or ray, if the right hand side of eq. (2.6) is known [2]. That is,

$$\Omega_{m,x}\frac{\partial \psi_{k,m,g}(x,y)}{\partial x} + \Omega_{m,y}\frac{\partial \psi_{m,g}(x,y)}{\partial y} + \Sigma_{t,k,g}(x,y)\psi_{k.m,g}(x,y) = q_{k,m,g}(x,y), \quad (2.8)$$

where

$$q_{k,m,g}(x,y) = \sum_{g'}^{G}\sum_{m'}^{M}\Sigma_{s,k,g'\to g}(x,y,\mathbf{\Omega}_{m'}\cdot\mathbf{\Omega}_m)\psi_{k,m',g'}(x,y)w_{m'} + \frac{\chi_{k,g}(x,y)}{4\pi}\sum_{g'}^{G}\sum_{m'}^{M}\nu\Sigma_{f,k,g'}(x,y)\psi_{k,m',g'}(x,y)w_{m'} - \frac{1}{4\pi\Delta z_k}\left[J_{k+1/2,g}(x,y) - J_{k-1/2,g}(x,y)\right], \quad (2.9)$$

and we have omitted the time dependence for simplicity. If a variable transformation:

$$\mathbf{r} = \mathbf{r}_0 + s\mathbf{\Omega}_m \quad (2.10)$$

is applied to (2.8), then the directional derivative can be expressed in terms of the new variable:

$$\frac{d\psi_{k,m,g}}{ds}(\mathbf{r}_0 + s\mathbf{\Omega}_m) + \frac{\Sigma_{t,k,g}(\mathbf{r}_0 + s\mathbf{\Omega}_m)}{\sqrt{1-\mu_m^2}}\psi_{k.m,g}(\mathbf{r}_0 + s\mathbf{\Omega}_m) = \frac{q_{k,m,g}(\mathbf{r}_0 + s\mathbf{\Omega}_m)}{\sqrt{1-\mu_m^2}}. \quad (2.11)$$

This equation can be solved analytically by multiplying through by the integrating factor

$$\exp\left(-\int_0^s \frac{\Sigma_{t,k,g}(\mathbf{r}_0+s'\mathbf{\Omega}_m y)}{\sqrt{1-\mu_m^2}}ds'\right), \tag{2.12}$$

which yields the solution

$$\psi_{k.m,g}(\mathbf{r}_0+s\mathbf{\Omega}_m)=\psi_{k.m,g}(\mathbf{r}_0)\exp\left(-\int_0^s \frac{\Sigma_{t,k,g}(\mathbf{r}_0+s'\mathbf{\Omega}_m y)}{\sqrt{1-\mu_m^2}}ds'\right) + \int_0^s q_{k,m,g}(\mathbf{r}_0+s'\mathbf{\Omega}_m)\exp\left(-\int_{s'}^s \frac{\Sigma_{t,k,g}(\mathbf{r}_0+s''\mathbf{\Omega}_m y)}{\sqrt{1-\mu_m^2}}ds''\right)ds' \tag{2.13}$$

with $q_{k,m,g}$ defined by eq. (2.9). If the geometry can be represented by discrete regions with constant material properties and if a flat source is also assumed within those regions, eq. (2.13) can be further simplified. For a given characteristic n passing through a discrete region i:

$$\psi^{out}_{k,i,g,m,n}=\psi^{in}_{k,i,g,m,n}\exp\left(-\frac{\Sigma_{t,k,i,g}s_{k,i,m,n}}{\sqrt{1-\mu_m^2}}\right)+\frac{q_{k,i,g,m}}{\Sigma_{t,k,i,g}}\left[1-\exp\left(-\frac{\Sigma_{t,k,i,g}s_{k,i,m,n}}{\sqrt{1-\mu_m^2}}\right)\right], \tag{2.14}$$

with

$$q_{k,i,g,m}=\sum_{g'}^{G}\sum_{m'}^{M}\Sigma_{s,k,i,g'\to g}(\mathbf{\Omega}_{m'}\cdot\mathbf{\Omega}_m)\bar{\psi}_{k,i,m',g'}w_{m'}+ \frac{\chi_{k,i,g}}{4\pi}\sum_{g'}^{G}\sum_{m'}^{M}\nu\Sigma_{f,k,i,g'}\bar{\psi}_{k,i,m',g'}w_{m'}-\frac{1}{4\pi\Delta z_k}\left[J_{k+1/2,g}-J_{k-1/2,g}\right]. \tag{2.15}$$

Here we have introduced the definitions $s_{k,i,m,n}$ as the distance ray n travels through region i in plane k along direction m, the quantities

$$\psi^{in}_{k,i,g,m,n}=\psi_{k,i,g,m,n}(\mathbf{r}_0), \tag{2.16a}$$

$$\psi^{out}_{k,i,g,m,n}=\psi_{k,i,g,m,n}(\mathbf{r}_0+s_{k,i,m,n}\mathbf{\Omega}_m), \tag{2.16b}$$

and the region-averaged angular flux

$$\bar{\psi}_{k,i,m,g}=\frac{\sum_{n\in i}\bar{\psi}_{k,i,m,g,n}s_{k,i,m,n}\delta A_{m,n}}{\sum_{n\in i}s_{k,i,m,n}\delta A_{m,n}}. \tag{2.17}$$

This in turn has introduced the cross-sectional area of the characteristic $\delta A_{m,n}$ and the segment-averaged angular flux

$$\bar{\psi}_{k,i,m,g,n}=\frac{\int_0^{s_{k,i,m,n}}\psi_{k,i,m,g,n}(s')ds'}{\int_0^{s_{k,i,m,n}}(s')ds'}=\frac{\psi^{in}_{k,i,g,m,n}-\psi^{out}_{k,i,g,m,n}}{\Sigma_{t,k,i,g}s_{k,i,m,n}}+\frac{q_{k,i,m,g}}{\Sigma_{t,k,i,g}} \tag{2.18}$$

Since the boundary conditions are defined, as they must be for a well-posed problem, ψ^{in} is known for each ray at the boundary of the system. This gives a starting point from which to trace a ray across the entire

system, with ψ^{out} from one discrete, flat-source region equal to ψ^{in} for the next region. Thus, eqs. (2.14), (2.18), and (2.17) provide an algebraic means to update the angular flux from a known source, q in eq. (2.15), without inverting a matrix. This ingenious methodology is much more amenable to parallelization than traditional matrix inversion techniques. It offers an additional benefit in that it does not require explicit storage of the angular flux. From the definition of the scalar flux in the discrete ordinance approximation, we can immediately deduce that the region-averaged scalar flux is given by

$$\bar{\phi}_{k,i,g} = \sum_{m}^{M} \bar{\psi}_{k,i,m,g} w_m, \tag{2.19}$$

so the contribution of the angular flux to the scalar flux can be accumulated as the rays are traced, and the angular flux can be immediately discarded. In fact, the segment-averaged and region-averaged angular fluxes need not be explicitly calculated at all if only the scalar flux is needed for the overall iteration scheme. The angular flux contribution to the scalar flux can be tabulated more efficiently through an algebraically optimized scheme that moves as many operations to the outer loop as possible. The content of this section is meant to provide only a basic understanding of MOC, as there are numerous considerations and subtleties beyond the scope of this thesis. Further detail can be found in references [38], [31].

In terms of the numerical implementation, the MOC method may be viewed as a collection of three primary operators. The source is computed via eq. (2.15) through individual applications of the scattering operator $\mathbf{S}$ and the fission operator $\mathbf{F}$. The MOC kernel, eq. (2.14), is an application of $(\mathbf{L}+\mathbf{R})^{-1}$ to the source. MPACT avoids explicit formation of the $\mathbf{L}$ operator by tracing rays, and as such, it should be noted that MPACT is incapable of calculating the the action of $\mathbf{L}$.

2.2.3 Coarse Mesh Finite Difference Acceleration

As noted in Section 2.1.1, the convergence for SI can be quite slow for scattering-dominated systems like nuclear reactors. This necessitates an acceleration scheme for SI to be of practical use. The standard choice is Coarse Mesh Finite Difference (CMFD) acceleration that utilizes a lower order diffusion solve to provide a coarse mesh correction to the transport solution.

The derivation of CMFD, like the diffusion equation, begins by integrating the transport equation over the angular variable. As the name implies, the acceleration method is not solved on the fine mesh, but rather on a coarse mesh, typically a pin cell. The resulting balance equation for a coarse mesh cell j with surfaces s[3] is then:

$$\sum_{s} J^{net}_{j,g,s} A_{j,s} + \Sigma_{t,j,g} \phi_{j,g} V_j = \left[\sum_{g'}^{G} \left(\Sigma_{s0,j,g'\rightarrow g} + \frac{\chi_g}{k_{eff}} \nu\Sigma_{f,j,g'} \right) \phi_{j,g'} \right] V_j, \tag{2.20}$$

where the coarse mesh-averaged quantities are defined from the fine mesh quantities as

[3]In the CMFD discussion, s is used to denote a surface, which should not be confused with the s in the MOC derivation that represents distance traveled along a characteristic

$$\Sigma_{x,j,g} = \frac{\sum_{i \in j} \Sigma_{x,i,g} \phi_{i,g} V_i}{\sum_{i \in j} \phi_{i,g} V_i}, \tag{2.21a}$$

$$\phi_{j,g} = \frac{\sum_{i \in j} \phi_{i,g} V_i}{\sum_{i \in j} V_i}, \tag{2.21b}$$

$$\chi_{j,g} = \frac{\sum_{i \in j} \chi_{i,g} \nu\Sigma_{f,i,g} \phi_{i,g} V_i}{\sum_{i \in j} \nu\Sigma_{f,i,g} \phi_{i,g} V_i}. \tag{2.21c}$$

Using the standard finite difference approximation to the diffusion equation, the net current across surface s in eq. (2.20) can be approximated as

$$J^{net}_{j,g,s} \approx -\tilde{D}_{j,g,s} \left(\phi_{j,g} - \phi_{j_s,g} \right), \tag{2.22}$$

where $\phi_{j_s,g}$ is the cell-averaged flux in the cell adjacent to cell j sharing surface s, and the diffusion coefficient $\tilde{D}_{j,g,s}$ is defined with the distance $h_{j,s}$ between the center of cells j and j_s and the traditional diffusion coefficient defined in eq. (1.18):

$$\tilde{D}_{j,g,s} = \frac{2 D_{j,g} D_{j_s,g}}{h_{j,s} \left(D_{j,g} + D_{j_s,g} \right)}. \tag{2.23}$$

CMFD improves upon this approximation by introducing a correction $\hat{D}_{j,g,s}$:

$$J^{net}_{j,g,s} \approx -\tilde{D}_{j,g,s} \left(\phi_{j,g} - \phi_{j_s,g} \right) + \hat{D}_{j,g,s} \left(\phi_{j,g} + \phi_{j_s,g} \right), \tag{2.24}$$

where $\hat{D}_{j,g,s}$ is calculated during the fine mesh transport solve:

$$\hat{D}_{j,g,s} = \frac{J^{net}_{j,g,s} + \tilde{D}_{j,g,s} \left(\phi_{j,g} - \phi_{j_s,g} \right)}{\left(\phi_{j,g} + \phi_{j_s,g} \right)}. \tag{2.25}$$

This correction factor, that can be viewed as an Eddington factor that represents the higher order angular contributions [54] [33], creates equivalence between the CMFD solution and transport (MOC) solution upon convergence, a most desirable trait for any acceleration scheme. The CMFD solution is then used to correct the fine mesh transport solution:

$$\phi^{n+1}_{i,g} = \phi^{n+1/2}_{i,g} \frac{\phi^{n+1}_{i,g}}{\phi^{n+1/2}_{i,g}}, \quad i \in j \tag{2.26}$$

where the superscript $n + 1/2$ represents the solution after the transport solve and and $n + 1$ represents the solution after the CMFD solve.

2.2.4 Transients

MPACT is capable of analyzing reactor transients (time-dependent problems), which necessitates solving a k-eigenvalue problem for the initial condition and then a series of calculations at discrete time points to characterize the evolution of the transient. Like most neutronic codes, the time dependent problem in MPACT is discretized with implicit Euler [35]. This provides necessary stability for the stiff set of equations

represented by eqs. (1.1) at the cost of solving a full linear system at each time point.

Furthermore, the time derivative term is treated with the isotropic approximation:

$$\frac{1}{v(E)}\frac{\partial}{\partial t}\psi(\mathbf{r},\mathbf{\Omega},E,t) \approx \frac{1}{4\pi v(E)}\frac{\partial}{\partial t}\phi(\mathbf{r},E,t). \tag{2.27}$$

While this approach is unable to capture high order transport effects, cancellation of error renders it an acceptable approximation [26] and allows avoidance of storing two instances of the angular flux. Combining this with implicit Euler, the time derivative term of eq. (1.1) becomes

$$\frac{1}{v(E)}\frac{\partial}{\partial t}\psi\left(\mathbf{r},\mathbf{\Omega},E,t_{n+1}\right) \approx \frac{1}{4\pi v(E)}\frac{\phi\left(\mathbf{r},E,t_{n+1}\right)-\phi\left(\mathbf{r},E,t_n\right)}{t_{n+1}-t_n}. \tag{2.28}$$

In addition to the time derivative term, solving the fully time dependent eqns. (1.1) requires solution of the neutron precursor equations to complete the modifications to the steady-state case. Rather than solve these explicitly, eq. (1.1b) is multiplied by the integrating factor $e^{-\lambda_i t}$ and a second order approximation in time is applied to the fission source ($\nu\Sigma_f\phi$) that allows for the precursor equations to be solved analytically in terms of the fission source from the current and two previous time steps. We will leave the precise mathematical formulation in reference [57] unrepeated here, as the important aspect is simply that it depends on the current fission source and other terms that can be considered a known external source at the current time point (no dependency on the current flux). This allows the overall fixed-source solution methodology from the steady-state calculation to be used for transient calculations as well by adding a transient source term, S_{tr}^n, to the right hand side:

$$\begin{aligned}\mathbf{\Omega}\cdot\nabla\psi^n(\mathbf{r},\mathbf{\Omega},E)+\Sigma_t(\mathbf{r},E)\psi^n(\mathbf{r},\mathbf{\Omega},E) = &\int_0^\infty\int_{4\pi}\Sigma_s(\mathbf{r},E'\to E,\mathbf{\Omega}\cdot\mathbf{\Omega}')\psi^n(\mathbf{r},\mathbf{\Omega}',E')ld\Omega' dE' \\ &+\frac{\chi(\mathbf{r},E)}{4\pi}\int_0^\infty\int_{4\pi}\nu\Sigma_f(\mathbf{r},E')\psi^n(\mathbf{r},\mathbf{\Omega}',E')d\Omega' dE' + S_{tr}^n(\mathbf{r},E),\end{aligned} \tag{2.29}$$

$$S_{tr}^n(\mathbf{r},E) = A(E)\phi^n(\mathbf{r},E)+B(\mathbf{r},E)\int_0^\infty\int_{4\pi}\nu\Sigma_f(\mathbf{r},E')\psi^n(\mathbf{r},\mathbf{\Omega}',E')d\Omega' dE' + C(\mathbf{r},E) \tag{2.30}$$

where

$$\begin{aligned}A(E) = -\frac{1}{v(E)\Delta t_n},\quad B(\mathbf{r},E) = \chi_d(\mathbf{r},E)\left[\omega^n(\mathbf{r})-\beta(\mathbf{r})\right], \\ C(\mathbf{r},E) = \chi_d(\mathbf{r},E)\tilde{S}_d^{n-1}(\mathbf{r}) + \frac{1}{v(E)\Delta t_n}\phi^{n-1}(\mathbf{r},E).\end{aligned} \tag{2.31}$$

The time derivative appears as the sole term in $A(E)$ and the second term in $C(\mathbf{r},E)$ while the delayed neutron precursor source appears in the first terms of $B(\mathbf{r},E)$ and $C(\mathbf{r},E)$. The second term of $B(\mathbf{r},E)$ is necessary to isolate all of the transient source modification into a single term, otherwise the fission source in eq. (2.29) would have to be modified as well.

MPACT typically employs the Transient Multilevel Method (TML) [57] for transient calculations, that consists of solves on the transport, CMFD, and EPKE levels with successively finer time steps in a Predictor-Corrector methodology. The focus of this work is on the transport step, since that provides the most difficulties, though much of what will be discussed applies to the CMFD and EPKE levels as well.

2.3 Modification of MPACT for Solution by JFNK

We now revisit the JFNK solution methodology from Section 2.1.2 in order to adapt it for use with MPACT. First let us summarize the basic MPACT SI algorithm:

Algorithm 1: MPACT Source Iteration

1 Choose ϕ^0, k^0
2 **for** $n = 1...$ **do**
3 $\quad q^n = \mathbf{S}\phi^{n-1} + \frac{1}{k^{n-1}}\mathbf{F}\phi^{n-1}$
4 $\quad \phi^n = \mathbf{\Gamma}\left(\mathbf{L}+\mathbf{R}\right)^{-1} q^n$
5 $\quad k^n = k^{n-1}\frac{\mathbf{F}\phi^n}{\mathbf{F}\phi^{n-1}}$
6 $\quad$ Exit if ϕ^n and k^n are sufficiently converged
7 **end**

Now we must recast the fixed-point iteration into the form $\mathbf{G}u = 0$. First, we define the solution vector $u = [\phi, \psi_{IN}, k]^T$ and the black box operator $\mathcal{M}$ to represent steps 3-5 from Algorithm 1 that produce the solution $[\ \phi^n, \psi_{IN}^n, k^n]^T = \mathcal{M}\left[\ \phi^{n-1}, \psi_{IN}^{n-1}, k^{n-1}\right]^T$ and rearrange the definition of the fixed-point iteration to define $\mathbf{G} = \mathbf{I} - \mathcal{M}$. Hence, Newton's method will be used to solve the problem:

$$(\mathbf{I} - \mathcal{M})u = 0, \quad u = [\phi, \psi_{IN}, k]^T \tag{2.32}$$

For each Newton (outer) iteration, the linear system

$$\mathbf{J}(u^{n-1})\delta u^n = \mathcal{M}u^{n-1} - u^{n-1} \tag{2.33}$$

is solved using a Krylov method. One MPACT sweep is thus required for each outer iteration to determine the right hand side of eq. (2.33). The Krylov (inner) iterations used to solve eq. (2.33) will require the product $\mathbf{J}(u^{n-1})v$ for arbitrary v, that can be estimated via finite difference as defined in eq. (2.4):

$$\mathbf{J}(u^{n-1})v \approx \frac{1}{\epsilon}\left[\mathbf{G}\left(u^{n-1}+\epsilon v\right) - \mathbf{G}u^{n-1}\right] = \frac{1}{\epsilon}\left[\mathcal{M}u^{n-1} - \mathcal{M}\left(u^{n-1}+\epsilon v\right)\right] + v. \tag{2.34}$$

Since $\mathcal{M}u^{n-1}$ has already been calculated to determine the right hand side of eq. (2.33), one MPACT sweep is required to calculate $\mathcal{M}\left(u^{n-1}+\epsilon v\right)$ for each Jacobain product required by the Krylov method. GMRES [46] is employed as the Krylov method for the remainder of this thesis due to its simplicity and robustness. For GMRES, only one Jacobian product, and hence one MPACT sweep, is required per Krylov iteration. The resulting algorithm to reconfigure MPACT for JFNK with GMRES is shown in Algorithm 2.

Algorithm 2: MPACT with JFNK

Choose $u_0 = \left[\phi^0, \psi^0_{IN}, k^0\right]^T$
for *n = 1...* **do**
  MPACT sweep and eigenvalue update to produce $u^* = \mathcal{M}u^{n-1}$
  Guess $\delta u^n_0 = 0 \rightarrow r_0 = -\mathbf{G}u^{n-1} = u^* - u^{n-1}$
  Set $\delta u^n_1 = \frac{r_0}{\|r_0\|}, \beta = \|r_o\|$
  for *k=1... n-1* **do**
    MPACT sweep and eigenvalue update to produce $\mathcal{M}\left(u^{n-1} + \epsilon\delta u^n_k\right)$
    $\delta u^n_{k+1} = \mathbf{J}(u^{n-1})\delta u^n_k \approx \frac{1}{\epsilon}\left[u^* - \mathcal{M}\left(u^{n-1} + \epsilon\delta u^n_k\right)\right] + \epsilon\delta u^n_k$
    Orthonormalize δu^n_{k+1}, producing $\mathbf{Q}_{k+1}$ and $\mathbf{H}_k$
    Find y_k such that $r_k = \|\mathbf{H}_k y_k - \beta e_1\|$ is minimized
    Check $\frac{r_k}{r_0}$ against tolerance
  end
  $\delta u^n = \mathbf{Q}_k y_k$
  Newton update $u^n = \left[\ \phi^n, \psi^n_{IN}, k^{(n)}\right]^T = u^{n-1} + \delta u^n$
  Check if u^n converged
end

Chapter 3

Adaptive Time Stepping

As detailed in Section 2.2.4, transient calculations typically take the form of a series of calculations at discrete time points. While the simplest method to select these time points is to employ constant time steps, this is not necessarily computationally efficient. When the reactor is evolving very quickly, shorter time steps should be taken to ensure that error incurred by the time discretization is kept to a minimum. On the other hand, when the reactor state is evolving slowly, taking shorter time steps is a waste of computational resources with minimal gain in accuracy. Figure 3.1 shows the distribution of calculation points in time with the vertical red lines for a typical power pulse with constant time steps (CTS) on the left and the optimal selection of time points with ATS on the right. This figure demonstrates how greater accuracy can be achieved with the same number of time points. Conversely, fewer time points with an optimal distribution could be used to obtain the same level of accuracy as the CTS case.

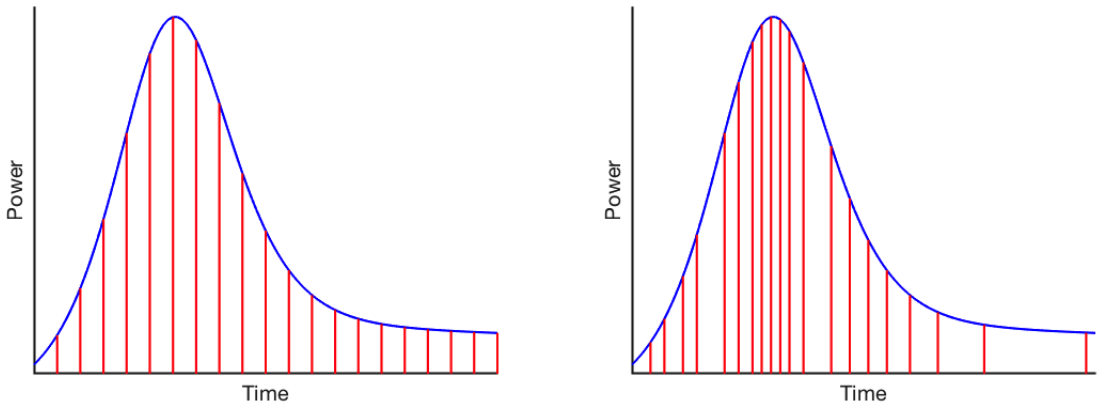

Figure 3.1: Demonstration of how ATS can redistribute the time steps more efficiently

3.1 Error Analysis of Time Discretization

Recently, Hackemack, Pounders and Boffie [22] [45] [6] have proposed a class of ATS schemes for the neutron diffusion equation based on limiting the local truncation error of the backward difference (implicit Euler) time discretization. The backward difference formulation is chosen specifically since it is used by most neutron diffusion and transport codes, including MPACT, as detailed in Section 2.2.4. While these papers focus on diffusion, the underlying methodology is equally applicable to the full transport equation, so we will present the ideas here in terms of the angular flux instead of the scalar flux. Note that the derivation is specific to the implicit Euler discretization of time, not the underlying physics being solved, so the scalar flux or total power could be substituted for the angular flux without affecting the results, as would be appropriate for the CMFD and EPKE levels of the MPACT transient calculation. The backward difference formula can be derived by applying a Taylor series expansion in time to the angular flux [35]:

$$\begin{aligned}\psi(t_{n-1}) &= \sum_{n=0}^{\infty} \frac{(t_{n-1}-t_n)^n}{n!} \frac{\partial^n \psi}{\partial t^n}|_{t_n} \\ &= \psi(t_n) - (t_n - t_{n-1}) \frac{\partial \psi}{\partial t}|_{t_n} + \frac{(t_n - t_{n-1})^2}{2} \frac{\partial^2 \psi}{\partial t^2}|_{t_n} + \mathcal{O}\left((t_n - t_{n-1})^3\right), \quad (3.1)\end{aligned}$$

where the spatial, direction, and energy dependence of ψ have been omitted for brevity. Letting $\psi^n = \psi(t_n)$ and rearranging eq. (3.1) for the first derivative yields the backward difference formula

$$\frac{\partial \psi}{\partial t}|_{t_n} = \frac{\psi^n - \psi^{n-1}}{t_n - t_{n-1}} + \tau_n, \quad (3.2)$$

where the truncation error τ_n is proportional to the time step size $h_n = t_n - t_{n-1}$ and the second derivative of the angular flux in time:

$$\tau_n = \frac{h_n}{2} \frac{\partial^2 \psi}{\partial t^2}|_{t_n} + \mathcal{O}\left(h_n^2\right). \quad (3.3)$$

From eq. (3.2), it is apparent that an effective ATS scheme should seek to limit the truncation error τ_n. This requires calculation, or at least estimation, of the second derivative in time of the angular flux, per eq. (3.3). This necessity will be addressed shortly, but first we will consider how the optimal time step size may be calculated given the second derivative and a prescribed error tolerance ϵ. It should be noted that the calculation of the second derivative and the calculation of the optimal time step size from some constraint placed on (3.3) are independent of each other.

The simplest method to determine the time step size is to directly limit the truncation error via an absolute error tolerance $\epsilon_{abs.}^{dir.}$, such that $\tau_n \leq \epsilon_{abs.}^{dir.}$. Note that the second derivative has an inherent spatial, angular, and energy dependence. Since time step size does not have such a dependence, this must be dealt with. The most robust choice is to take the minimum time step size over all dimensions, which equates the maximum second derivative (or second derivative over the angular flux for the relative criteria). The resulting absolute direct criteria is:

$$h_n = 2\epsilon_{abs.}^{dir.} \left\| \frac{\partial^2 \psi}{\partial t^2}|_{t_n} \right\|_\infty^{-1}. \tag{3.4}$$

References [22], [45], and [6] all go on to describe a relative error criterion that is *not* a direct extension of this absolute error tolerance, but actually introduces time step integration as well. That criteria will also be explored in this work, but will be denoted with the superscript *int.* to differentiate it from the direct criteria. First, however, we will introduce a more basic relative error tolerance $\epsilon_{rel.}^{dir.}$:

$$h_n = 2\epsilon_{rel.}^{dir.} \left\| \frac{\partial^2 \psi}{\partial t^2}|_{t_n} / \psi^n \right\|_\infty^{-1}. \tag{3.5}$$

Another possible constraint is to limit the incurred error integrated over the time step instead of the truncation error itself. By conservatively assuming that the truncation error is constant over the time step, this approach requires limiting the quantity $h_n \tau_n$ or $h_n \tau_n / \psi(t_n)$ for the absolute and relative constraints, respectively. The resulting time step calculations are then:

$$h_n = \sqrt{2\epsilon_{abs.}^{int.} \left\| \frac{\partial^2 \psi}{\partial t^2}|_{t_n} \right\|_\infty^{-1}} \tag{3.6}$$

and

$$h_n = \sqrt{2\epsilon_{rel.}^{int.} \left\| \frac{\partial^2 \psi}{\partial t^2}|_{t_n} / \psi^n \right\|_\infty^{-1}}. \tag{3.7}$$

These are not the only constraints possible on the truncation error, but are the most natural choices. For clarity, we will refer to the first two time step criteria given by eqns. (3.4) and (3.5) as absolute and relative direct methods, respectively, and the latter set of time step criteria given by eqns. (3.6) and (3.7) as absolute and relative integrated methods, respectively. The merits of these four criteria will be investigated in this thesis, but the general expectations of the criteria relative to each other are summarized in Table 3.1.

Table 3.1: Qualitative expectations of the ATS criteria

	Direct	Integrated
Absolute	more stringent error limit more reactive to evolution	more stringent error limit less reactive to evolution
Relative	more adaptable error limit more reactive to evolution	more adaptable error limit less reactive to evolution

Additionally, we consider the impact of replacing the angular flux in eq. (3.4) through (3.7) with the scalar flux and power. For the absolute criteria, using the scalar flux amounts to an isotropic approximation of the second derivative which will require division by 4π. As in the treatment of the time derivative term for the transient solution, eq. (2.27), we expect cancellation of error to improve the validity of this approximation since the variation in the angular dependence of the solution is typically much slower than the spatial or energy dependence [27]. For the relative criteria, the cancellation of error should be more obvious. Since the angular flux does not have a simple approximate relation to the power like to the scalar flux, we employ the following:

$$\frac{1}{\psi}\frac{\partial\psi}{\partial t} \approx \frac{1}{P}\frac{dP}{dt} \quad \rightarrow \quad \frac{\partial\psi}{\partial t} \approx \frac{\phi}{4\pi P}\frac{dP}{dt}, \tag{3.8}$$

which necessarily includes the isotropic approximation $\psi \approx \phi/4\pi$ for the absolute criteria since the purpose of the investigation is to avoid the angular flux in the first place. The validity of these approximations will be investigated in Chapter 5 and 6.

3.1.1 Finite Difference Approach

We now turn to the task of calculating the second derivative. Since we are already relying on a finite difference approach to handle the time dependence of the calculation in general, the most intuitive approach for calculating the second derivative is to again employ a finite difference method. Indeed, this is the approach taken by Hackemack, Pounders, and Boffie [22] [45] [6]. They investigate both a traditional 3-point stencil in time:

$$\frac{\partial^2\psi}{\partial t^2}|_{t_n} \approx 2\frac{h_{n-1}\psi^n - (h_{n-1}+h_n)\psi^{n-1} + h_n\psi^{n-2}}{h_{n-1}h_n(h_{n-1}+h_n)}, \tag{3.9}$$

referred to in reference [45] as the interpolated difference method, and an alternative 3-point stencil:

$$\frac{\partial^2\psi}{\partial t^2}|_{t_n} \approx \frac{h_{n-1}\psi^n - (h_{n-1}+h_n)\psi^{n-1} + h_n\psi^{n-2}}{h_{n-1}h_nh_n}, \tag{3.10}$$

referred to as the nested difference method. While the interpolated difference can be obtained in a straightforward manner from Taylor expansion, the nested difference method is obtained by dual application of the backward difference operator. Pounders shows that for an idealized surrogate model, the interpolated difference method can lead to oscillations in time step size, which results in a decrease in accuracy when the calculated time step is too large and a decrease in computational efficiency when the time step is too small. The nested difference method is proposed to resolve this issue. Indeed, the interpolated difference method is shown to exhibit these oscillations in predicted time step size for a simple slab model without feedback, while the nested difference method does not show oscillatory behavior. The nested difference method does, however, carry an additional grid-bias term that tends to zero as time step size tends to constant. Limits on the growth rate of time steps may be required to ensure this term does not affect the accuracy of the calculations.

For the transport solution, this method raises an immediate and substantial concern– it requires storing the angular flux from three time points. The MOC method is optimized to avoid storing even the angular flux of the current time point to ensure reasonable memory requirements of the code, so storing three instances of the angular flux is obviously undesirable. This concern may be circumvented by averaging the angular flux over any of space, angle, and energy, including sub-domains. The viability and tradeoffs of using any of these lower order approximations to estimate the second derivative via eqns. (3.9) or (3.10) are investigated in Chapter 5.

3.1.2 α-Eigenvalue Eigenvalue Approach

An alternative approach, and the focus of this thesis, is to employ the α-eigenvalue eigenvalue for the calculation of the second derivative in hopes that this might lead to more favorable (non-oscillatory) behavior.

From the definition of the α-eigenvalue in eq. (1.23), it is tempting to simply define the second derivative of the angular flux in time as

$$\frac{\partial^2 \psi}{\partial t^2}|_{t_n} \approx \alpha_n^2 \psi^n. \tag{3.11}$$

This, however, is a stretch of the α-eigenvalue beyond its physical representation. One only need consider that since the fundamental α-eigenvalue lies on the real axis, eq. (3.11) can never yield a negative result to realize that such a simple approach is unlikely to yield a meaningful estimation of the second derivative. Instead, we perform a backward difference estimation of the first derivative to calculate the second derivative:

$$\frac{\partial^2 \psi}{\partial t^2}|_{t_n} \approx \frac{1}{\Delta t_n}\left(\frac{\partial \psi}{\partial t}|_{t_n} - \frac{\partial \psi}{\partial t}|_{t_{n-1}}\right) = \frac{\alpha_n \psi^n - \alpha_{n-1}\psi^{n-1}}{\Delta t_n}. \tag{3.12}$$

For the relative criteria

$$\frac{1}{\psi^n}\frac{\partial^2 \psi}{\partial t^2}|_{t_n} \approx \frac{\alpha_n - \alpha_{n-1}\psi^{n-1}/\psi^n}{\Delta t_n}. \tag{3.13}$$

These approximations still utilize finite differencing, and therefore may not resolve the oscillations observed by Pounders, et al. However, previous work with MPACT has shown the second derivative to be a particularly finicky quantity [27]. The α-eigenvalue approach only requires a first derivative finite difference approximation, which we expect will be less prone to oscillations. Additionally, the quantities in eqs. (3.12) and (3.13) still depend on the angular flux, though it does require one less instance of it than the finite difference method. For both the finite difference and α methods of estimating the second derivative, we will consider the impact of substituting an isotropic approximation for the time derivative(s) of angular flux.

Chapter 4

α-eigenvalue Calculations with MPACT

Before utilizing the α-eigenvalue for ATS, MPACT must be modified to enable its calculation. Since MPACT is a source iteration solver by construct, the most obvious choice would be direct modification to calculate α instead of k. However, we recall from our initial introduction of the α-eigenvalue in Section 1.2.2 that the fundamental α-eigenvalue is not necessarily the largest magnitude eigenvalue that is located by SI. Since there is an *a priori* lower bound on the eigenspectrum, a shift could be applied to ensure that the fundamental eigenvalue is the largest in magnitude [47]. However, convergence of SI is generally proportional to the dominance ratio, or ratio of the largest eigenvalue to the next largest eigenvalue [17]. Since the lower limit of the α point eigenspecturm is $v\Sigma_t$ and neutron velocities extend into the range of 1E9 cm/s, such a shift would have detrimental effects on the convergence properties of SI, even with acceleration.

A pivot to a fundamentally different methodology such as Arnoldi's method has some appeal. Unfortunately, the operator associated with the α-eigenvalue is $\mathbf{V}\left[(\mathbf{S}+\mathbf{F})-(\mathbf{L}+\mathbf{R})\right]$. From Section 2.2.2, we recall that a foundation of MPACT is its avoidance of explicitly formulating the leakage operator. In lieu of a direct calculation of α, we turn to $k-\alpha$ iteration [13] [43] [50]. In this basic methodology, a k-eigenvalue calculation is performed for some guess of α; the estimate of α is updated from the result; and the process is repeated until the calculation yields $k=1$. This forces the α-eigenvalue to the fundamental mode that correlates with the fundamental k-eigenvalue, thereby assuaging our spectral concerns.

To implement the $k-\alpha$ iteration in MPACT, the nominal MPACT k-eigenvalue solver is wrapped in an outer iteration that estimates α and repeats the k-eigenvalue calculation until α is sufficiently converged such that $k=1.0$. For each guess of α, the total (and therefore transport) cross-sections are modified by the addition of α/v and the modified fission spectrum χ_α is calculated as necessary. The natural choice for an initial guess is $\alpha_0=0$, corresponding to the critical case $k=1$.

The three major choices for this algorithm are how often to update the α-eigenvalue, how to treat the flux after each α-eigenvalue update, and the convergence criteria. Before we can consider any of these, which we shall in the ensuing sections of this chapter, we must first establish how the α-eigenvalue is to be updated and how convergence is to be checked. First, let us consider the update. For the second α iteration and

beyond, a simple linear relationship is assumed between k and α and the value of α that would yield $k = 1.0$ is interpolated or extrapolated from this relation. After the first α iteration, however, this relation cannot be established, so a general relation is desired. For this we turn to the point kinetics equation. By performing the point kinetics derivations explicitly on the α-eigenvalue equation, the PKE equation for the amplitude function, eq. (1.31a) becomes:

$$\alpha p(t) = \frac{\rho(t) - \beta_{eff}(t)}{\Lambda(t)} p(t) + \frac{1}{\Lambda(0)} \sum_i \lambda_i C_i(t) \tag{4.1}$$

We wish to obtain a simple expression that is not dependent on the delayed neutron precursor density since these are not calculated during the steady-state calculation. One option is to simply neglect the contribution of the delayed neutrons, resulting in a simple estimate for the α-eigenvalue from a single k iteration:

$$\alpha = \frac{\rho - \beta_{eff}}{\Lambda}, \tag{4.2}$$

where $\rho = (k-1)/k$. Alternatively, this relation can be obtained directly by performing the PKE derivation on the asymptotic α-eigenvalue eq. (1.28). We note that for a critical system ($k = 1$) eq. (4.2) yields $\alpha = -\beta/\Lambda$, that does not fit with our basic conception of the α-eigenvalue. For large α, the precursor balance equation is dominated by the source term ($\beta_i \mathbf{F}\psi >> \lambda C_i$), and hence the time constant of the precursor equations approaches that of the neutron balance equation. This is precisely the assumption used in deriving eq. (1.28), so we expect eq. (4.2) to provide an approximation to the asymptotic eigenvalue, α_a. An alternative estimate suggested by Perdu [43] and further explored by Singh [50] is:

$$\alpha = \frac{\rho}{\Lambda}. \tag{4.3}$$

Eq. (4.3) can be obtained by applying the PKE derivation to the prompt α-eigenvalue eq. (1.26), and hence we expect that it will provide a good approximation to α_p. Note that α_p does fulfill our intuition that $k = 1$ corresponds to $\alpha = 0$. We will explore both variations in our calculation of the α-eigenvalues:

1. A "asymptotic α-eigenvalue", α_a, that includes the α adjustment to the fission spectrum
2. A "prompt α-eigenvalue", α_p, that does not include an adjustment to the fission spectrum

We expect that α_a should generally be consistent with eq. (4.2), while α_p will be better described by eq. (4.3).

The other major concern is how to check if α is sufficiently converged. The fundamental α-eigenvalue exists in a much larger range than the k-eigenvalue, necessitating the use of relative convergence criteria. However, if α is close to zero, relative convergence criteria can result in severely over-solving the problem. Thus, both relative and and absolute errors are calculated from successive calculations of α, and the minimum is compared to the convergence criteria:

$$\epsilon_\alpha \geq \min\left(\left| \frac{\alpha_n - \alpha_{n-1}}{\alpha_n} \right|, |\alpha_n - \alpha_{n-1}| \right). \tag{4.4}$$

For $|\alpha| \geq 1$, relative errors will be used, while for $|\alpha| < 1$, absolute errors will be used, ensuring that α is converged to a reasonable level regardless of its magnitude. A value of 2E-3 for the α-eigenvalue convergence

criteria provides a good balance between accuracy and computational efficiency

Our initial investigations will utilize a 2D single fuel pin so that we can easily compare the fine mesh flux. The pin, shown in Figure 4.1, consists of a UO_2 fuel pin with a 0.412 cm radius and zirconium cladding with a 0.476 cm outer radius surrounded by pure water moderator in a square pin with pitch 1.265 cm. Reflective boundary conditions are used on all sides, representing an infinite array of the pins. There are 6 radial divisions (3 in the fuel, 1 in the clad, 2 in the moderator) and 8 azimuthal divisions. Five different enrichments, that are detailed in Table 4.1, are analyzed to cover various levels of criticality. MPACT is run with CMFD enabled.

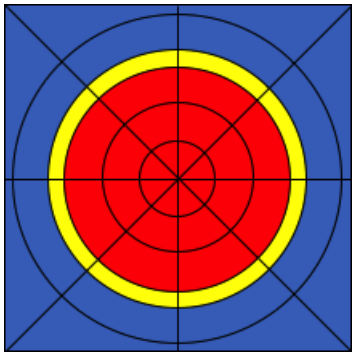

Figure 4.1: MPACT Single Pin Model

Table 4.1: Single Pin Steady State α Calculation Test Cases

Label	pin00	pin01	pin02	pin03	pin04
Enrichemnt (wt%)	1.042	1.070	1.074	1.077	1.106
k	0.990	0.999	1.000	1.001	1.010

4.1 Basic $k-\alpha$ Iteration in MPACT

Naturally, the most stable $k-\alpha$ algorithm will allow k and the associated eigenvector to converge fully before updating α, so we begin our calculations there. Algorithm 3 shows the overall structure of such an iterative scheme.

Algorithm 3: $k-\alpha$ Iteration with fully converged k with Flux Reset

```
Choose α_0 = 0
while α not converged do
    Modify Σ_t with α/v
    if calculating α_a then
        Calculate χ_α
    end
    Choose k_0 = 1, φ(r, E) = φ_0
    while k and φ not converged do
        MPACT SI iteration
    end
    Update α
end
```

For each case detailed in Table 4.1, both the α_a and α_p eigenvalues are calculated by Algorithm 3 and the results are shown in Table 4.2. Both eigenvalues are compared to the expected relations in eqs. (4.2) and (4.3), respectively. The results indicate that the majority of the computational burden lies in converging the eigenvector since each α iteration requires the same number of k iterations. Figure 4.2 further demonstrates this point with the pin04 case, showing the evolution of both the α_p eigenvalue and the k-eigenvalue in each α iteration.

Additionally, Tables 4.4 and 4.5 present a comparison of the average and maximum relative differences between the eigenvectors (fine mesh scalar flux) associated with the k and α-eigenvalues. Figure 4.3 compares the radial dependence of the k and α_p fluxes for pin04 for energy groups 2 and 8 (slowing down and thermal regions, respectively). The result clearly indicate that the k eigenvector is a close approximation to the α-eigenvector.

Finally, we note the differences between the "asymptotic" and "prompt" α-eigenvalues. Both eigenvalues are well-predicted by their respective relations given in eqs. eqs. (4.2) and (4.3). Generally, the α_p eigenvector is a bit closer to the k-eigenvector than the α_a eigenvector. This result is expected since the α_a eigenvalue uses an adjusted delayed neutron spectrum while the α_p eigenvalue uses the same spectrum as the k eigenvalue problem, and bears out in the number of iterations required to achieve convergence. The critical case also provides an interesting result for the asymptotic α-eigenvalue, requiring two extra α iterations to converge as the algorithm seems to be converging on the prompt eigenvalue for one iteration before correcting course. This seems to suggest that it might be more appropriate to use $\alpha_0 = -\beta/\Lambda$ as the initial guess rather than

$\alpha_0 = 0$ for the asymptotic α-eigenvalue.

Table 4.2: Results of $k - \alpha$ Iteration with Algorithm 3 for Steady State Pins for α_a

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_a	-308.42	-150.04	-132.17	-114.27	53.98
Predicted α_a	-321.13	-152.61	-134.07	-115.57	49.28
No. of α Iterations	4	4	6	4	4
Total No. of k Iterations	60	60	90	60	60

Table 4.3: Results of $k - \alpha$ Iteration with Algorithm 3 for Steady State Pins for α_p

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_p	-180.25	-18.33	0.00	18.40	187.12
Predicted α_p	-187.06	-18.54	0.00	18.50	183.35
No. of α Iterations	3	3	1	3	3
Total No. of k Iterations	45	45	15	45	45

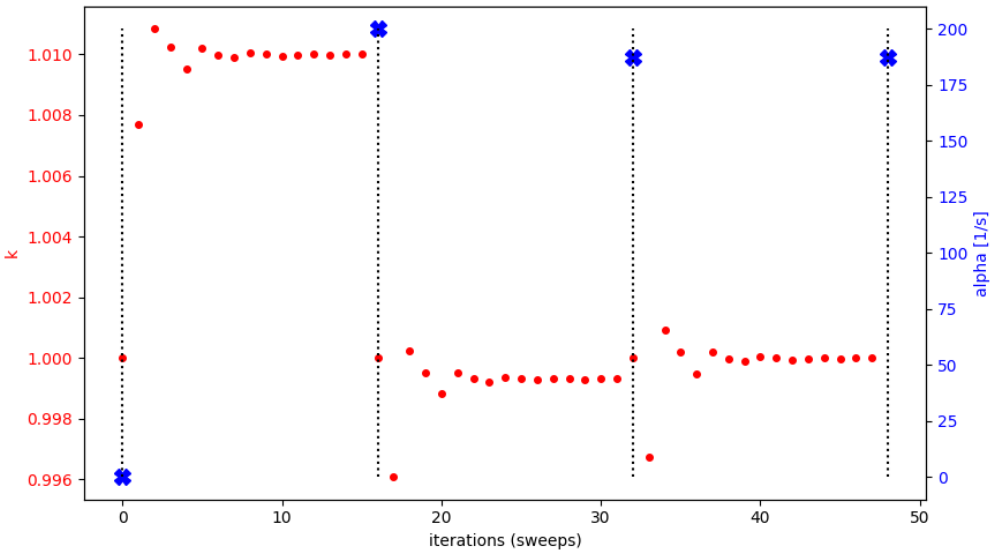

Figure 4.2: Evolution of k and α_p eigenvalues for $k - \alpha$ Iteration with Algorithm 3 for pin04

Table 4.4: α_a Flux Comparison for $k - \alpha$ Iteration with Algorithm 3 for Steady State Pins

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_a	-308.42	-150.04	-132.17	-114.27	53.98
Avg. Rel. Diff. ϕ	0.98%	0.92%	0.96%	0.99%	1.29%
Max. Rel. Diff. ϕ	1.28%	1.30%	1.31%	1.31%	1.66%

Table 4.5: α_p Flux Comparison for $k - \alpha$ Iteration with Algorithm 3 for Steady State Pins

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_p	-180.25	-18.33	0.00	18.40	187.12
Avg. Rel. Diff. ϕ	0.53%	0.49%	0.00%	0.56%	0.86%
Max. Rel. Diff. ϕ	0.76%	0.52%	0.00%	0.65%	1.78%

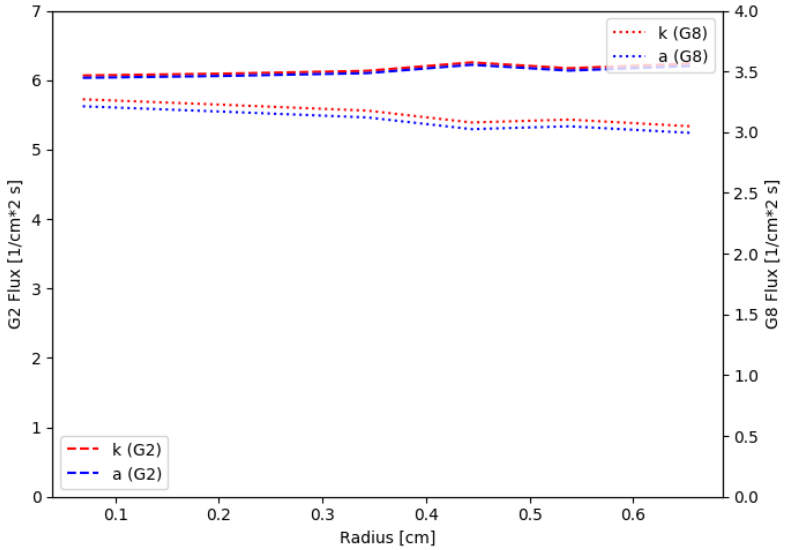

Figure 4.3: Radial Comparison of k and α_p Eigenvectors for$k - \alpha$ Iteration with Algorithm 3 for pin04

4.2 Eigenvector Treatment for $k - \alpha$ Iteration

As we have seen in Section 4.1, much of the computational expense of the k calculation is associated with converging the eigenvector (flux). This would suggest that great improvement to the computational efficiency of Algorithm 3 could be obtained with a simple modification– do not reset $\phi(\mathbf{r}, E)$ at the beginning of each iteration. As discussed in Section 1.2.2, the eigenvectors associated with the fundamental k and α-eigenvalues are not generally equal (except in the critical case), but they may be good approximations to each other as we have seen in the previous section, and certainly a better approximation than a flat guess. Thus, we might expect the α iterations beyond the first to converge much more quickly by reusing the flux from the previous k calculation, as shown in Algorithm 4.

Algorithm 4: $k - \alpha$ Iteration with fully converged k with Flux Reuse

```
Choose α_0 = 0, φ(r, E) = φ_0
while α not converged do
    Modify Σ_t with α/v
    if calculating α_a then
        Calculate χ_α
    end
    Choose k_0 = 1
    while k and φ not converged do
        MPACT SI iteration
    end
    Update α
end
```

Once again, each of the cases in Table 4.1 are run, this time with Algorithm 4 and the results are shown in Tables 4.6 and 4.7. The first α iteration of Algorithms 3 and 4 are identical, as they start with an initial guess of $\alpha_0 = 0, k_0 = 1, \phi(\mathbf{r}, E) = \phi_0$. However, subsequent α iterations converge much more quickly since the flux begins from a close approximation to the eigenvector, as expected. Figure 4.4 demonstrates this point nicely for pin04 α_p when compared to Figure 4.2.

As for the initial $k - \alpha$ implementation, we would like to characterize the difference between the k- and α-eigenvectors. Tables 4.8 and 4.9 present a comparison of the average and maximum relative differences between the eigenvectors. We would expect the k- and α-eigenvectors to show greater agreement than in the case where the flux is reset for each α iteration since the convergence is purely on the eigenvalue in the former case. This is true for the cases near critical, but actually the α-eigenvectors calculated with the reused flux show greater deviation from the k-eigenvectors further from critical. Note that this does not imply the expected eigenvector is different between the methods, only the calculation result. Figure 4.5 compares the radial dependence of the k and α fluxes for pin04 for energy groups 2 and 8 (slowing down and thermal regions, respectively).

In addition to a comparison of the k- and α-eigenvectors, the α-eigenvectors computed by Algorithms 3 and 4 are compared in Tables 4.10 and 4.11 and Figure 4.6. The results indicate that Algorithm 4 can be used to calculate the α-eigenvalues much more efficiently than Algorithm 3 with little impact on the final result. This has very important implications when we attempt to calculate the α-eigenvalue during transients, since we will already have a good estimate of the eigenvector from the most recent time step calculation.

Table 4.6: Results of $k - \alpha$ Iteration with Algorithm 4 for Steady State Pins for α_a

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_a	-308.50	-150.11	-132.21	-114.33	53.92
Predicted α_a	-321.13	-152.61	-134.07	-115.57	49.28
No. of α Iterations	4	4	6	4	5
Total No. of k Iterations	21	22	27	22	22

Table 4.7: Results of $k - \alpha$ Iteration with Algorithm 4 for Steady State Pins for α_p

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_p	-180.31	-18.39	0.00	18.35	187.06
Predicted α_p	-187.06	-18.54	0.00	18.50	183.35
No. of α Iterations	3	4	1	3	3
Total No. of k Iterations	20	20	15	18	20

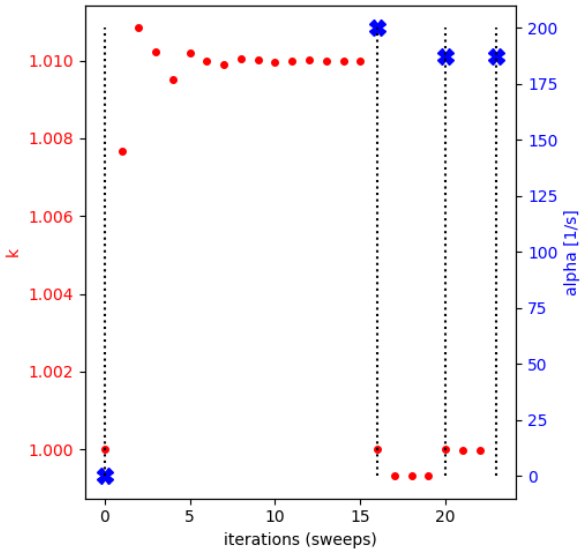

Figure 4.4: Evolution of k and α_p eigenvalues for $k - \alpha$ Iteration with Algorithm 4 for pin04

Table 4.8: α_a Flux Comparison for $k-\alpha$ Iteration with Algorithm 4 for Steady State Pins

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_a	-308.50	-150.11	-132.21	-114.33	53.92
Avg. Rel. Diff. ϕ	1.20%	0.60%	0.65%	0.39%	0.31%
Max. Rel. Diff. ϕ	1.74%	0.96%	1.00%	0.63%	1.00%

Table 4.9: α_p Flux Comparison for $k-\alpha$ Iteration with Algorithm 4 for Steady State Pins

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_p	-180.31	-18.39	0.00	18.35	187.06
Avg. Rel. Diff. ϕ	0.68%	0.07%	0.00%	0.07%	0.70%
Max. Rel. Diff. ϕ	0.88%	0.10%	0.00%	0.10%	0.93%

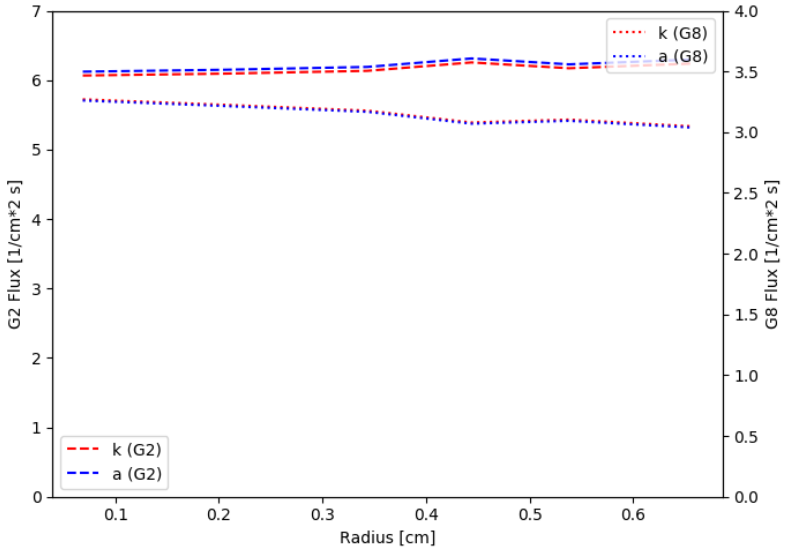

Figure 4.5: Radial Comparison of k and α_p Eigenvectors for $k-\alpha$ Iteration with Algorithm 4 for pin04

Table 4.10: Comparison of Flux for Algorithms 3 and 4 for α_p

Label		pin00	pin01	pin02	pin03	pin04
α_a	Algorithm 3	-308.42	-150.04	-132.17	-114.27	53.98
	Algorithm 4	-308.50	-150.11	-132.21	-114.33	53.92
Avg. Rel. Diff. ϕ		0.47%	0.31%	0.31%	0.69%	1.58%
Max. Rel. Diff. ϕ		0.48%	0.33%	0.33%	0.71%	1.59%

Table 4.11: Comparison of flux for Algorithms 3 and 4 for α_p

Label		pin00	pin01	pin02	pin03	pin04
α_p	Algorithm 3	-180.25	-18.33	0.00	18.40	187.12
	Algorithm 4	-180.31	-18.39	0.00	18.35	187.06
Avg. Rel. Diff. ϕ		0.38%	0.43%	0.00%	0.62%	1.46%
Max. Rel. Diff. ϕ		0.39%	0.45%	0.00%	0.63%	1.48%

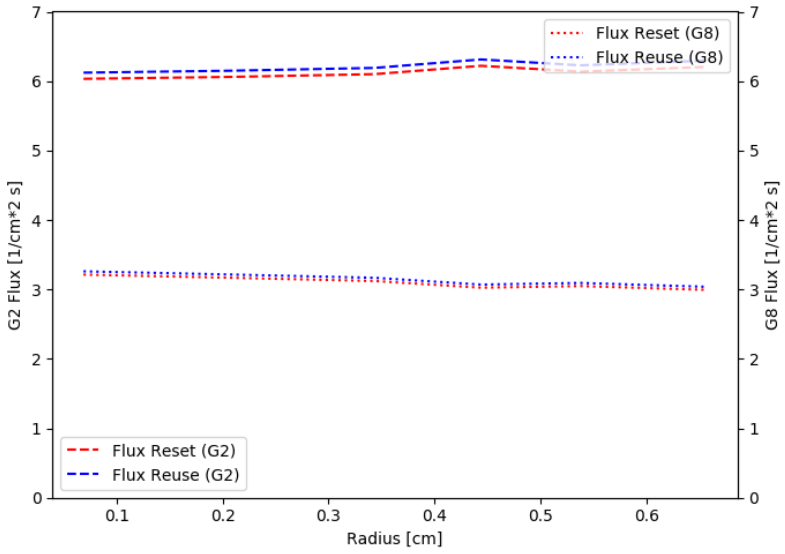

Figure 4.6: Radial comparison of α_p eigenvectors for fully converged $k - \alpha$ Iteration with Algorithm 3 for pin04

4.3 Frequency of α Update

One of the fundamental questions for the $k-\alpha$ iteration scheme is when to update α. The update can be performed after each k update, only upon a fully converged k, or anywhere in between (e.g. after every nth iteration or after k has met some partial convergence criteria). The flux would necessarily be reused from step to step. Such an algorithm was implemented in MPACT, but, unsurprisingly, proved to be unstable since k does not converge monotonically[1]. When it did converge, the results were comparable to those generated by Algorithms 3 and 4, but were less efficient than the results presented in the previous section.

4.4 α-Eigenvalue Calculation with JFNK

As an alternative to the SI methodology, MPACT was modified to operate with a JFNK solver, as in Algorithm 2 in Section 2.3. For a simple comparison, MPACT was run with SI only (no CMFD acceleration). The overall $k-\alpha$ iteration strategy is unmodified except now the k solve is accomplished via JFNK. The results from the prompt α-eigenvalue are presented in Tables 4.12 and 4.13 for Algorithms 3 and 4, respecitvely. For comparison's sake, the total number of sweeps is simply the sum of the Newton iterations and the Krylov iterations. To calculate k with pure SI requires 230-240 iterations, so the fact that we are able to complete the $k-\alpha$ iteration for less than that (at least in the case of flux reuse) is a testament to the power of JNK.

While the ability to modify MPACT to use JFNK is a fascinating topic in its own right, this solution methodology will be abandoned going forward for three primary reasons: lack of a suitable preconditioner, degradation of performance when paired with CMFD[2], and untenable storage demands. These three items are really all tied together. Preconditioning is paramount for efficient Krylov solves to reduce the storage demands by solving the linear system in fewer Krylov iterations. It was hoped that CMFD could be utilized as a preconditioner. Unfortunately, this would require CMFD to act directly on the Newton correction δu rather than the perturbed solution $u+\epsilon v$. By acting on a vector that does not approximate the scalar flux (and in fact is all but guaranteed to have negative components), the closures $\tilde{D}$ and $\hat{D}$ break down.

Table 4.12: Results of JFNK $k-\alpha$ Iteration with Algorithm 3 for Steady State Pins for α_p

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_p (SI)	-180.25	-18.33	0.00	18.40	187.12
α_p (JFNK)	-180.31	-18.39	-0.06	18.34	187.06
No. of α Iterations	3	3	3	3	3
No. of Newton Iterations	9	9	9	9	9
No. of Krylov Iterations	282	282	282	282	282

[1] For an illustration of this, refer to Figure 4.2 and note the oscillatory convergence of the k-eigenvalue. When the α eigenvalue is updated after each k iteration, these oscillations have been observed to lead to divergence in some calculations

[2] When CMFD is combined with the transport operator for the definition of an MPACT sweep, there is speed up with respect to JFNK without the CMFD operator, but the nominal CMFD-accelerated SI algorithm in MPACT is more efficient than the JFNK algorithm.

Table 4.13: Results of JFNK $k-\alpha$ Iteration with Algorithm 4 for Steady State Pins for α_p

Label	pin00	pin01	pin02	pin03	pin04
k	0.990	0.999	1.000	1.001	1.010
α_p (SI)	-180.25	-18.33	0.00	18.40	187.12
α_p (JFNK)	-180.31	-18.39	-0.06	18.34	187.06
No. of α Iterations	3	3	3	3	3
No. of Newton Iterations	7	6	5	6	7
No. of Krylov Iterations	212	181	151	181	212

4.5 Summary of Steady State $k-\alpha$ Calculation Investigations

MPACT is capable of calculating both the asymptotic and prompt α-eigenvalues through $k-\alpha$ iteration. The calculated eigenvalues show good agreement with the values predicted by PKE, giving hope that we may be able to obtain a useful approximation to the second derivative for ATS purposes directly from the PKE. Another important result of these studies is demonstration of the similarity of the eigenvectors associated with the k and α-eigenvalues, as least for the basic single pin test case examined. While there is little mathematical basis to assume such similarity (except in the critical case), the eigenvectors show very little divergence in practice. This has a very important implication in that it's generally much more efficient to calculate an eigenvalue if we already possess a good approximation to the eigenvector. Since the ultimate goal of the α calculation is to efficiently predict a desirable time step size, it may in fact be more prudent to simply use a relation derived from PKE and accept the small error incurred rather than explicitly calculate the α-eigenvalue.

We have also demonstrated that it is also possible to modify MPACT to employ a JFNK algorithm in place of the nominal SI solution algorithm. This is independent of the application to calculation of the α eigenvalue and is an extraordinarily interesting topic in its own right. However, it presents numerous challenges that are better addressed independently from ATS, so it will not be pursued further in this dissertation.

Chapter 5

ATS Calculations with Finite Differencing in MPACT

MPACT was initially developed for transient with a constant time step. Later a variable time step capability was added (by the author). For the work in this thesis, MPACT was modified to allow time steps to be calculated as the transient progresses, rather than *a priori*.

Regardless of the ATS method, the transient calculation begins with a steady state calculation to determine the initial conditions. Transients are assumed to start from a true steady state configuration, so the steady state k is used to modify the fission source throughout the transient. Since there is no time dependent data available from the steady state position, the first two time steps must be taken with a set value that was chosen as half of the nominal CTS time step size. User experience has shown that this value is typically more than sufficient to produce a well-converged solution with constant time steps, but that it is smaller than the typical time step size needed to obtain an accurate solution. This provides a good balance between the two major concerns with the initial time step size: 1) minimizing the error in the initial time steps since this has a tendency to propagate and even amplify through the transient 2) not over-solving the beginning of the transient such that the computational efficiency of the ATS schemes are dominated by this.

After the second transient step is completed, the beginning of each subsequent step begins with a time step determination. The second derivative is estimated by the finite difference method. This is a straightforward application of eqns. (3.9) or (3.10) with stored data. Once an estimate for the second derivative (relative or absolute) is obtained, eqns. (3.4) through (3.7) are then used to determine the ensuing time step size. The transient calculation proceeds unmolested through the time step.

The initial investigations will again utilize the single pin configuration described in Section 4.1 with an enrichment of 1.09%. This corresponds to a very nearly critical configuration ($k = 1.0000003$). The transient is driven by a fictitious ramp change in the enrichment over 50 ms and run to $t_{end} = 200$ ms. Transients with and without feedback were considered. MPACT was run in serial with CMFD acceleration, but with the transient multilevel calculation disabled to provide a simpler comparison. The 8-group library was employed

with 16 azimuthal angles with Chebyshev quadrature. A reference solution was generated with a constant time step size of 0.2 ms and treated as the exact solution for purposes of error quantification.

MPACT was modified to calculate and store the angular flux of the three most recent time steps in order to carry out the finite difference approximation. This requires an additional 2 arithmetic operation in the innermost loop of the MOC solver since the fine mesh angular flux is not typically calculated. In order to quantify performance, we require a measure of both computational efficiency and accuracy. For the computational performance, we will compare run times. While individual run times are not necessarily a good measure of computational efficiency since they can be adversely affected by the state of the processor(s), the sensitivity studies provide us a large sample of data points to draw general conclusions from. For a measure of accuracy, the power curves generated by a given set of modeling choices will be compared to a reference solution generated with a very fine time step (in this case 0.2 ms). For the superprompt transients, the peak power and time to peak power will be directly compared to the reference solution. For both the subprompt and superprompt transients, we will compare the entire power curve to the reference solution with the maximum relative difference, the root mean square (RMS) relative difference, and the integral of the difference:

$$\epsilon_{MAX}^{rel.} = \max_t \left| \frac{P(t) - P_{ref.}(t)}{P_{ref.}(t)} \right|, \tag{5.1a}$$

$$\epsilon_{RMS}^{rel.} = \sqrt{\left(\frac{P(t) - P_{ref.}(t)}{P_{ref.}(t)} \right)^2} \tag{5.1b}$$

$$\epsilon_{INT}^{rel.} = \int_0^{t_{end}} \left| \frac{P(t) - P_{ref.}(t)}{P_{ref.}(t)} \right| dt. \tag{5.1c}$$

Since the solutions are discrete, rather than continuous, the evaluation of eqs. (5.1) require interpolation. For the maximum and RMS quantification, the reference solution is interpolated onto the grid of the solution to be quantified, but for the integral quantification, the solution to be quantified is interpolated on the grid of the reference solution so that valuable data is not lost.

We begin by analyzing the various choices available for using the second derivative for optimal time step size determination. The parameters for consideration are the criteria used for time step determination, the tolerance used with the criteria, how the second derivative is calculated, and the limit on how much consecutive steps may be increased or decreased by, referred to as the growth limit. In addition, we will compare performance of the two finite difference schemes proposed by Hackemack [22].

5.1 Finite Difference Results without Feedback

We begin with the simpler case of no feedback since this allows us to properly evaluate the ATS method on the merits of neutron transport only, from which the methodology was developed. For the case of no feedback, we will analyze a subprompt reactivity insertion of \$0.56, which corresponds to increasing the enrichment to 1.10%. This transient has been selected since, in general, positive reactivity insertions are more challenging/interesting transients than negative reactivity insertions. While the limiting RIAs for reactor analysis are typically superprompt transients, without feedback this corresponds to a purely exponential

increase in power that deviates from any physical significance very quickly. A subprompt transient, however, maintains physical significance for low power excursions since the temperature change may be small enough that feedback is negligible anyway.

5.1.1 ATS Criteria for Finite Difference Method without Feedback

We begin by studying the four different criteria defined in eqs. (3.4) through (3.7). For each criteria, a range of tolerances covering 3 orders of magnitude were analyzed. The initial time step size was selected to be 2.5 ms to ensure that unnecessary error is not accumulated at the beginning of the transient. From there, the ATS scheme was allowed to dictate the time step size within the range of 0.5 ms to 20 ms. Consecutive time steps were also limited to a growth factor of 100%. The resulting power curves are shown in Appendix A Figures A.1, A.2, A.3, A.4 for the absolute direct, relative direct, absolute integrated, and relative integrated criteria, respectively. Quantification of the errors with respect to the reference solution are also shown in Appendix A Tables A.1, A.2, A.3, and A.4 as well. For comparison, a 5 ms constant time step case is also shown in the tables and figures. This is meant to provide a reference point for what we might expect from a reasonable simulation that balances the need for accuracy and computational efficiency. In order to compare the criteria directly, the tolerances are adjusted for each criterion until a result comparable to the 5 ms CTS case is obtained. The results are plotted in Figure 5.1 and summarized in Table 5.1.

For all criteria types, as the tolerance is tightened, the number of time steps increases and the power curve approaches the reference solution. This can be observed qualitatively in the figures and quantitatively in the tables. As general observations, the direct criteria are more sensitive to the tolerance than the integrated criteria. This sensitivity is owed to the direct criteria's proportionality to the tolerance, while the integrated criteria is proportional to the square root of the tolerance. The inverse of the second derivative shares this same relation to the criteria. However, the results do clearly indicate whether this sensitivity is advantageous or not yet.

The finite difference ATS scheme is able to produce comparable accuracy to the 5 ms CTS case with about 35% fewer time steps. However, there is a roughly 15% increase in the average calculation time for each time step ($\overline{RT}_{ts}$). This increase is owed mostly to the additional operations needed to calculate the angular flux, as we shall see shortly. Overall, we see a 20-25% improvement in run time with no adversarial effects on the solution. The timesteps used for each criteria are plotted in Figure 5.2 to show how the time step size evolves through the transient. Figure 5.3 also shows the maximum second derivative of the angular flux to confirm our intuition of the time step behavior. Between about 25 and 50 ms, the growth rate of the angular flux (and power) is reasonably linear, leading to an increase in time step size, before an inflection point between the prompt jump and the asymptotic solution causes the time step size to reduce. From Figure 5.3, we note that this is not a traditional inflection point corresponding to a second derivative of zero. It actually marks a discontinuity in the second derivative as the ramp perturbation ends where the second derivative flips from positive to negative. As the solution converges to the asymptotic solution, the growth rate is nearly linear, allowing for very large time steps with little loss of accuracy. The final fluctuation in time step size is to ensure the transient simulation ends at the desired time and does so without violating the minimum time step size or growth limit. We do note the oscillatory behavior described by Hackemack and Pounders, but

we will address this further in Section 5.1.2.

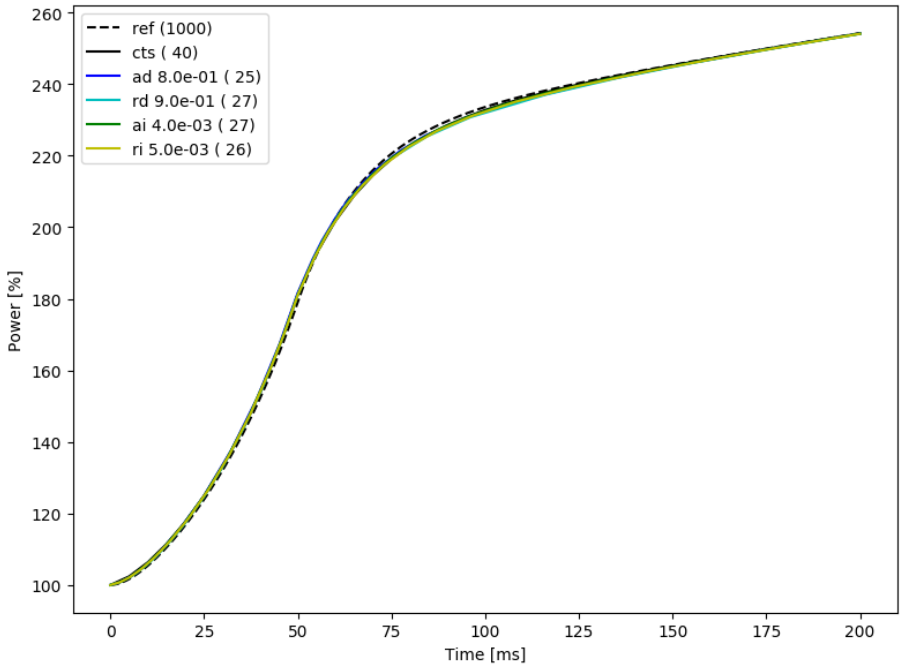

Figure 5.1: Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type, Time Steps Shown in Parentheses

Table 5.1: Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type

Crit. Type	Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	-	-	-	0.024
CTS	-	40	1.22	0.52	0.08	0.025
Absolute Direct	8.0e-01	25	1.41	0.62	0.09	0.030
Relative Direct	9.0e-01	27	1.14	0.54	0.09	0.029
Absolute Integrated	4.0e-03	27	1.12	0.58	0.09	0.029
Relative Integrated	5.0e-03	26	1.07	0.58	0.09	0.029

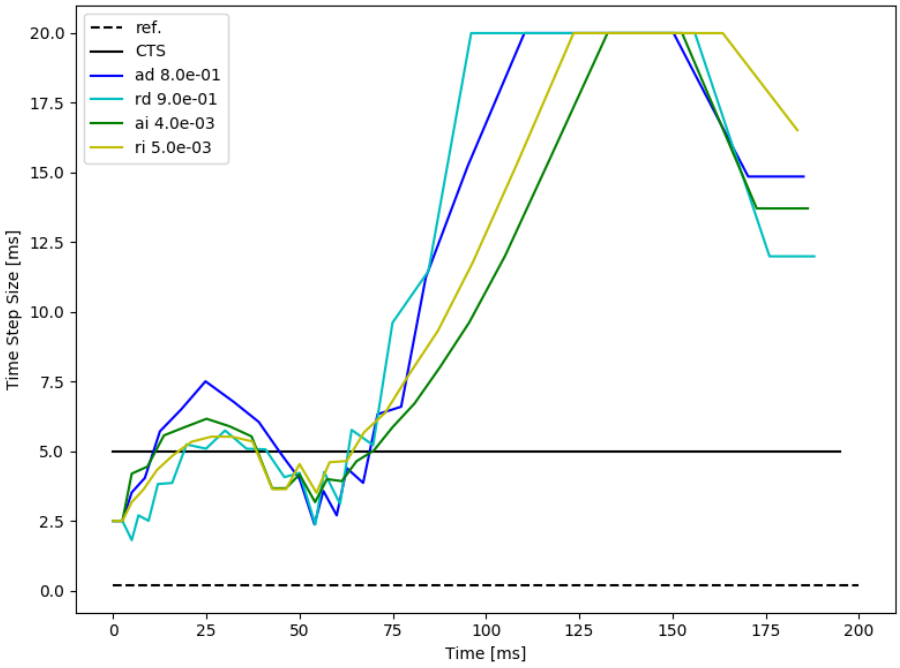

Figure 5.2: Single Pin Subprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type

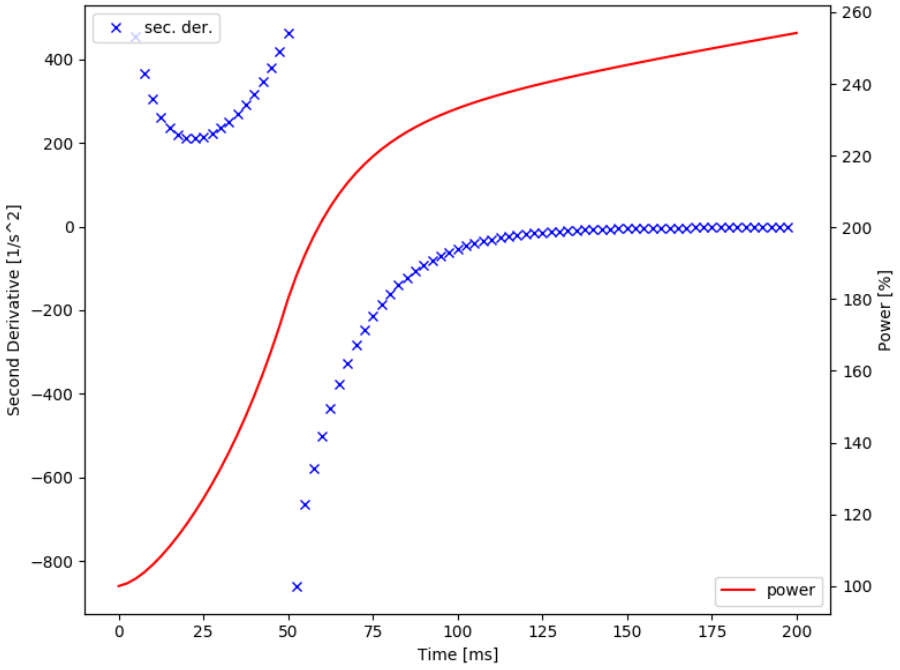

Figure 5.3: Single Pin Subprompt Transient Power and Second Derivative of the Angular Flux

5.1.2 Finite Difference Second Derivative Estimation

Having confirmed the oscillatory behavior observed by Hackemack and Pounders [22] [45], we immediately turn to their proposed solution– replacing the so-called interpolated finite difference scheme, eq. (3.9) with the nested finite difference scheme, eq. (3.10). The results presented in Figures 5.1 and 5.2 and Table 5.1 are reproduced in Figures 5.4 and 5.5 and Table 5.2 with the nested difference scheme in place of the interpolated difference scheme. As the figure shows, the nested difference scheme does indeed eliminate the oscillations[1].

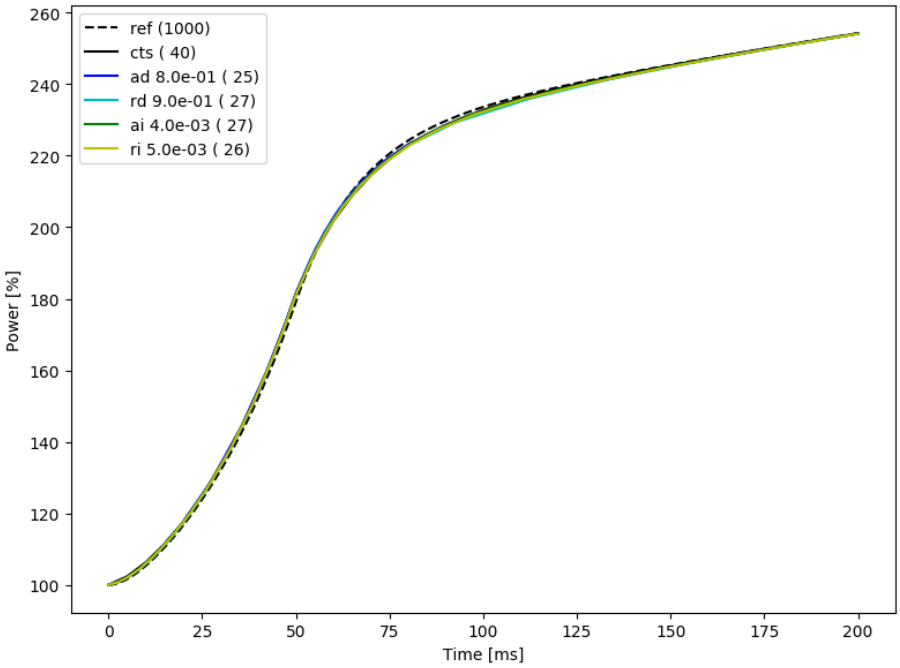

Figure 5.4: Single Pin Subprompt Power Curves for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type, Time Steps Shown in Parentheses

[1]The variation in the time step size around 50 ms is due to the end of the reactivity insertion ramp where the time step size must be adjusted to end at the desired time without violating the minimum, maximum, or growth limits. In the same vein, the time step size must be adjusted to exactly reach the end time of the transient causing some ugly, but necessary variation in the time step size for the purely practical requirement of ending the transient at the desired time without violating the time step limits. This artifact could be resolved by allowing the transient to run to an end time in a given range.

Table 5.2: Single Pin Subprompt Error Comparison for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type

Crit. Type	Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	-	-	-	0.024
CTS	-	40	1.22	0.52	0.08	0.025
Absolute Direct	8.0e-01	25	1.40	0.66	0.10	0.028
Relative Direct	9.0e-01	27	1.11	0.56	0.10	0.028
Absolute Integrated	4.0e-03	27	1.14	0.59	0.09	0.028
Relative Integrated	5.0e-03	26	1.07	0.59	0.09	0.028

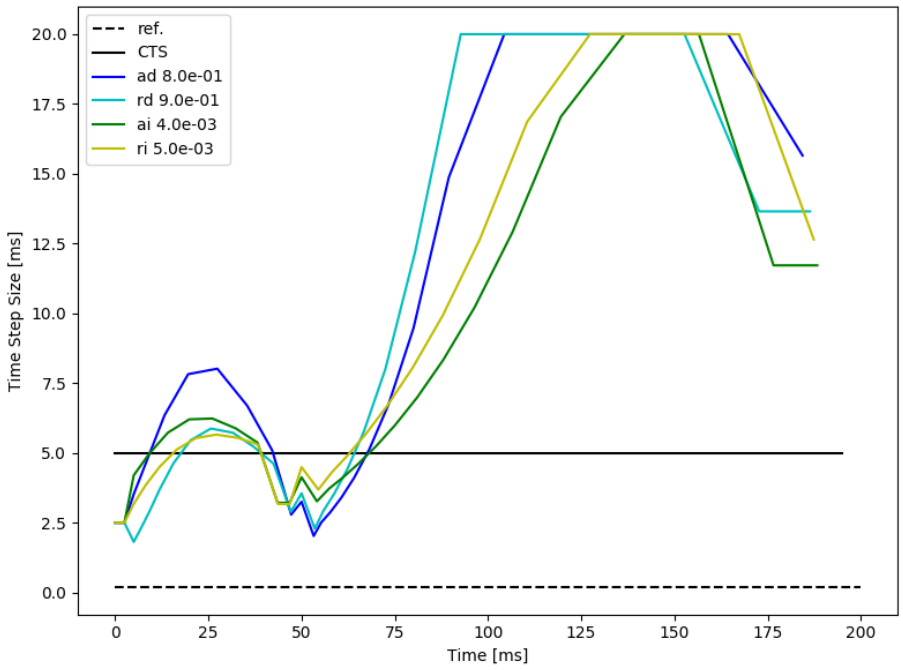

Figure 5.5: Single Pin Subprompt Time Step Size for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type

5.1.3 Finite Difference Flux Choice

While the angular flux is the "correct" quantity to track for ATS at the transport level, we will consider using the scalar flux in its place. As discussed in Section 2.2.4, the scalar flux is already used in MPACT as a stand-in for the angular flux in the time derivative term. It has been shown that cancellation of error makes the isotropic approximation of time derivative(s) of the angular flux more reasonable than a general isotropic approximation of the angular flux [27], hence we might expect the same behavior in calculating the second derivative for ATS. This offers both computational and memory savings since the angular flux need not be calculated and stored. No modifications are required to the transport solver and since the scalar flux from the previous time point is already stored for calculating the time derivative, only one additional instance of the scalar flux need be stored for the finite difference scheme. For the absolute criteria, the second derivative is divided by 4π to maintain consistency with the angular flux calculation; for the relative criteria, this is unnecessary. The results presented in Figures 5.1 and 5.2 and Table 5.1 are reproduced in Figures 5.6 and 5.7 and Table 5.3 with the scalar flux used to estimate the second derivative rather than the angular flux. The results for the scalar flux in Figures 5.6 and 5.7 and Table 5.3 are nearly identical to the results for the angular flux in Figures 5.1 and 5.2 and Table 5.1, indicating that the scalar flux is indeed a suitable substitute for the angular flux for ATS determination. Additionally, the average run time for a time step is reduced to where it is not significantly distinguishable from the CTS case.

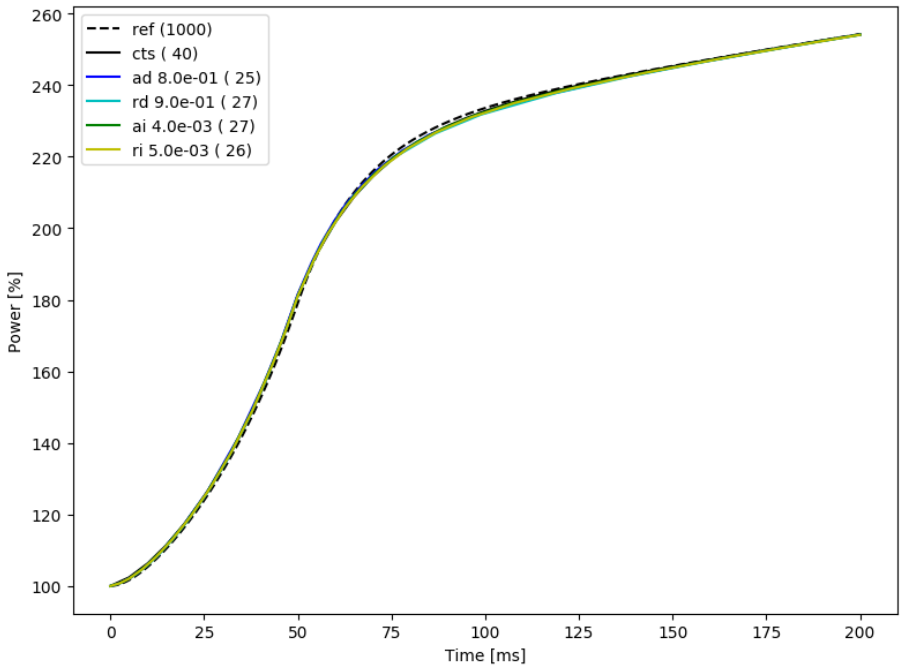

Figure 5.6: Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type, Time Steps Shown in Parentheses

Table 5.3: Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type

Crit. Type	Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	-	-	-	0.024
CTS	-	40	1.22	0.52	0.08	0.025
Absolute Direct	8.0e-01	25	1.33	0.64	0.10	0.026
Relative Direct	9.0e-01	27	1.10	0.56	0.10	0.026
Absolute Integrated	4.0e-03	27	1.17	0.60	0.09	0.026
Relative Integrated	5.0e-03	26	1.05	0.59	0.09	0.025

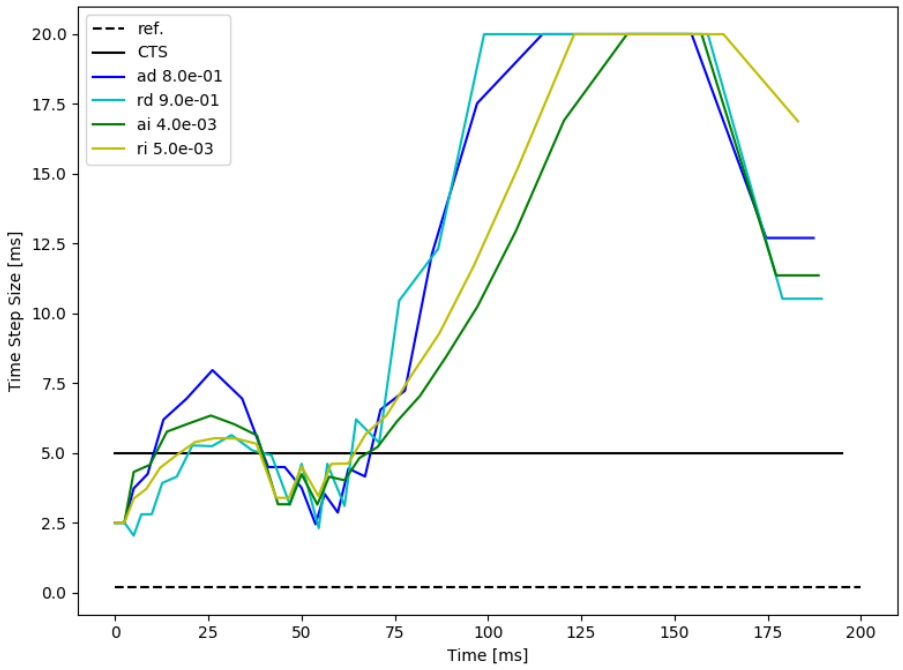

Figure 5.7: Single Pin Subprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type

We will take this one step further and consider using the power as a stand-in for the angular flux, as described in eq. (3.8). For this simple pin cell case, the logarithmic second derivative of the power agrees well enough with the angular flux that we would expect it to yield good results. This is indeed the case, as we once again reproduce Figure 5.1 and 5.2 and Table 5.1 using the second derivative of the power instead of the angular flux in Figures 5.8 and 5.9 and Table 5.4. As expected, the results are in excellent agreement.

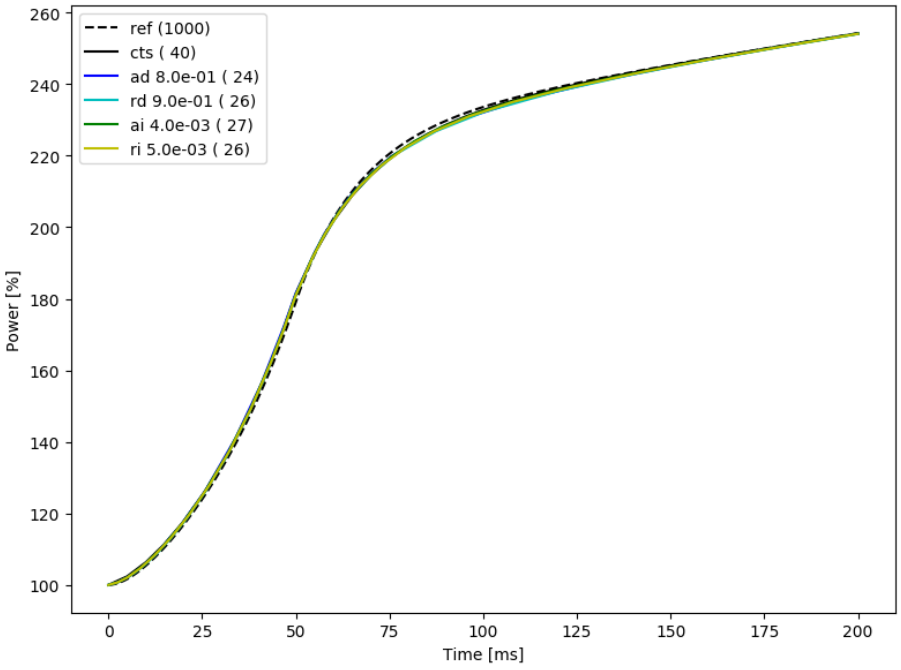

Figure 5.8: Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type, Time Steps Shown in Parentheses

Table 5.4: Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type

Crit. Type	Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	-	-	-	0.024
CTS	-	40	1.22	0.52	0.08	0.025
Absolute Direct	8.0e-01	24	1.45	0.66	0.10	0.026
Relative Direct	9.0e-01	26	1.10	0.57	0.10	0.025
Absolute Integrated	4.0e-03	27	1.16	0.60	0.09	0.025
Relative Integrated	5.0e-03	26	1.05	0.59	0.09	0.026

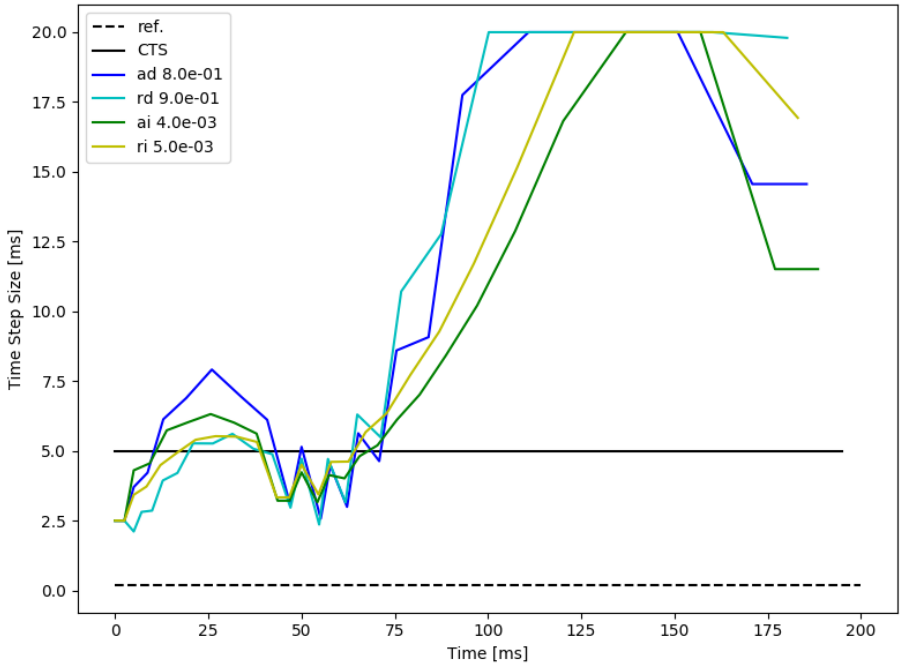

Figure 5.9: Single Pin Subprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type

5.2 Finite Difference Results with Feedback

We now introduce feedback to the model. For the proceeding studies, we induce a superprompt transient by increasing the enrichment to 1.11% This corresponds to a reactivity insertion of \$1.23, on par with a limiting RIA for reactor analysis. This also presents a much more challenging, though interesting, case for ATS since there is a large power spike over a short period of time.

5.2.1 ATS Criteria for Finite Difference Method with Feedback

As for the analysis without feedback, we begin by studying the four different criteria defined in eqs. (3.4) through (3.7). For each criteria, a range of tolerances covering three orders of magnitude were analyzed. The initial time step size was selected to be 1 ms to ensure that unnecessary error was not accumulated at the beginning of the transient. From there, the ATS scheme was allowed to dictate the time step size within the range of 0.5 ms to 20 ms with a growth limit of 100%. The resulting power curves are shown in Appendix A Figures A.5, A.6, A.7, A.8 for the absolute direct, relative direct, absolute integrated, and relative integrated criteria, respectively. Quantification of the errors with respect to the reference solution are shown in Appendix A Tables A.5, A.6, A.7, and A.8 as well. For comparison, a 2 ms constant time step case is also shown in the tables and figures. This is meant to provide a reference point for what we might expect from a reasonable simulation that balances the need for accuracy and computational efficiency. In order to compare the criteria directly, the tolerances are adjusted for each criterion until a result comparable to the 2 ms CTS case is obtained. The results are plotted in Figure 5.10 and summarized in Table 5.5.

Again, for all criteria types, as the tolerance is tightened, the number of time steps increases and the power curve approaches the reference solution. This can be observed qualitatively in the figures and quantitatively in the tables. The direct criteria are more sensitive to the tolerance (and second derivative) than the integrated criteria. However, with feedback on we notice a marked difference in performance– the direct criteria are overly sensitive to the second derivative and require significantly more time steps to produce solutions of comparable accuracy to the integrated criteria. Another observable trend is that the absolute criteria predict the peak power better, while the relative criteria predict the time to peak power better. The relative criteria yield smaller time steps at the beginning of the transient that lead to a more accurate capture of the power rise than the absolute criteria. However, as the power increases, the relative criteria allow larger absolute errors that cannot capture the peak behavior as well as the absolute criteria. This can be observed by plotting the time steps used for each criteria as in Figure 5.11, which confirms our intuition that the absolute criteria produce larger time steps at the beginning of the transient while using smaller time steps near the peak compared to the relative criteria.

In order to produce comparable accuracy as the 2ms CTS case, the ATS schemes actually require a similar number of or even *more* time steps for each of the criteria analyzed. This exposes a weakness in the ATS scheme dealing with inflection points where the second derivative passes through zero as it changes signs, as shown in Figure 5.12. The inflection points cause the predicted time step size to be large at a time when the system is actually evolving rapidly. When the second derivative is zero, the leading order error term from eq. (3.1) is actually proportional to the square of the time step size and the third derivative in time,

and, while Figure 5.12 does not show the third derivative, it is readily observable that this quantity is not negligible as the second derivative rapidly crosses from positive to negative and vice versa. Physically, the 75 ms inflection point represents the transition between the prompt excursion and the feedback-dominated portion of the transient. Initially, the change in time is purely exponential, but as the power level rises, thermal-hydraulic feedback causes a departure from the exponential behavior that briefly appears linear before the feedback effects take over. This brief linear behavior corresponds to a zero (or nearly so) second derivative that results in large time steps at moment when the transient is actually evolving quite rapidly.

Finally, we also note from Figure 5.11 that the oscillations have worsened considerably for this transient. This will be explore more thoroughly in Section 5.2.2

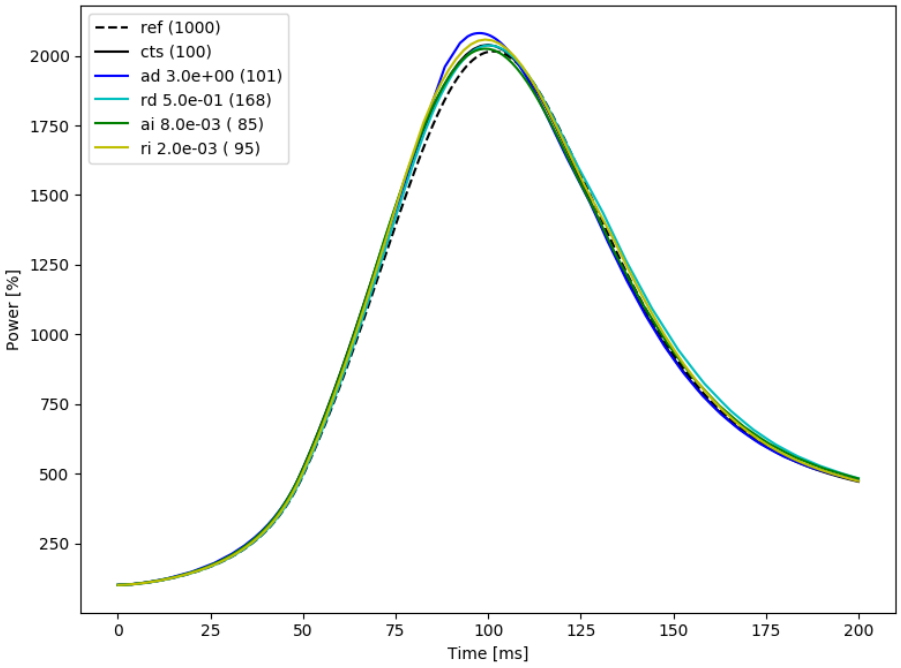

Figure 5.10: Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type, Time Steps Shown in Parentheses

Table 5.5: Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type

Crit. Type	Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	2016	101.40	-	-	-	0.061
CTS	-	100	2039	100.00	5.41	2.55	0.40	0.061
Absolute Direct	3.0e+00	101	2081	97.53	6.41	3.34	0.55	0.063
Relative Direct	5.0e-01	168	2037	99.90	5.15	1.69	0.40	0.063
Absolute Integrated	8.0e-03	85	2025	99.00	5.28	3.06	0.51	0.065
Relative Integrated	2.0e-03	92	2056	98.67	5.22	2.16	0.38	0.063

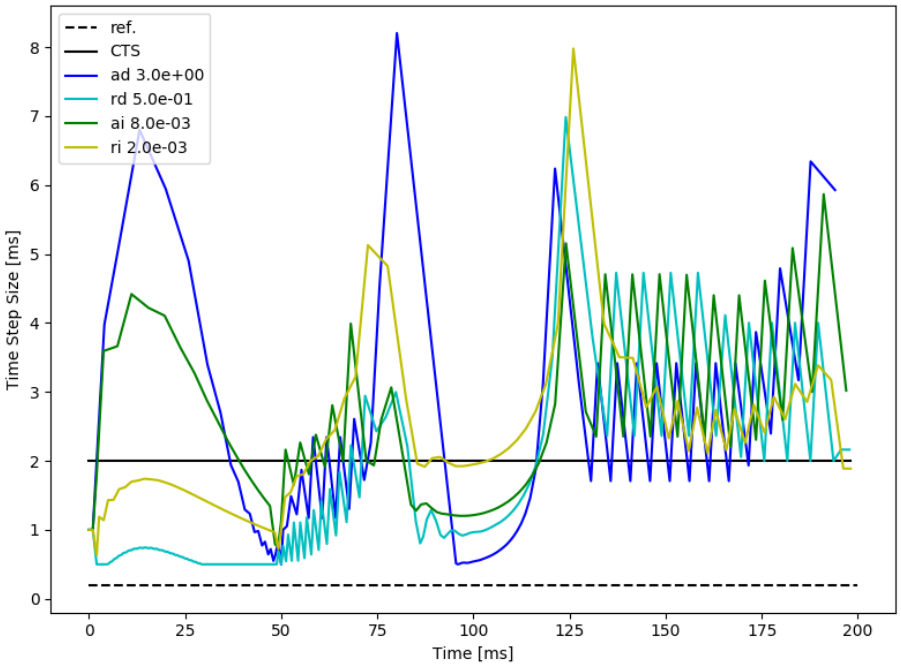

Figure 5.11: Single Pin Superprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Angular Flux for Each Criteria Type

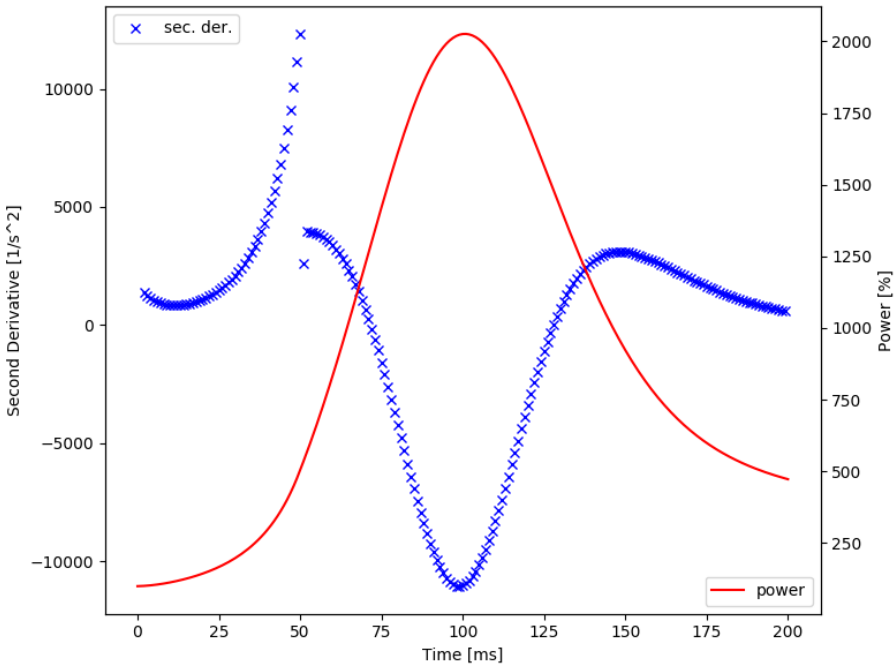

Figure 5.12: Single Pin Superprompt Transient Power and Second Derivative of the Angular Flux

5.2.2 Finite Difference Second Derivative Estimation and Growth Limits

Once again, we attempt to resolve the oscillations by replacing the interpolated finite difference scheme, eq. (3.9) with the nested finite difference scheme, eq. (3.10). The results presented in Figures 5.10 and 5.11 and Table 5.5 are reproduced in Figures 5.13 and 5.14 and Table 5.6 with the nested difference scheme in place of the interpolated difference scheme. As the figure shows, the nested difference scheme does indeed reduce, but not eliminate the oscillations. Unfortunately, this also comes at a fairly significant cost to the accuracy of the solution, as can be noted by comparing Figure 5.13 with Figure 5.10 and Table 5.6 with Table 5.5. The non-uniformity of the effects on accuracy indicate that the ATS scheme is generally performing in a less than ideal manner.

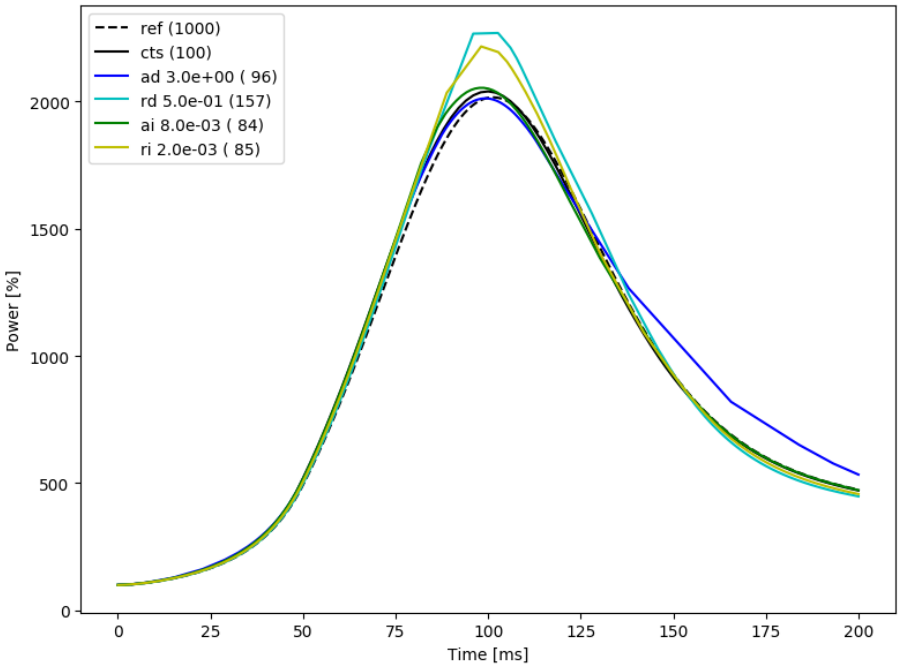

Figure 5.13: Single Pin Superprompt Power Curves for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type, Time Steps Shown in Parentheses

Pounders specifically notes in [45] the need for a growth limit on consecutive time steps to limit the error in the finite difference estimates. We will now examine what affect the growth limit has on the oscillations. For each criteria with tolerances as in Figure 5.14 and Table 5.6, cases were run with growth limits of 0% (which corresponds to constant time steps), 1%, 10%, 50%, and 100%. The results are plotted in Figures 5.15 through 5.18 for absolute direct, relative direct, absolute integrated, and relative integrated criteria, respectively. In each figure, the "ideal" time step size is plotted as well from the actual second derivative of the solution. These figures seem to suggest that the oscillations increase in magnitude with the growth limit, but there is a tradeoff. While the smaller growth limits do limit the oscillations, they restrict the ability of the time steps to adapt to the state of the system. It appears somewhere in the 50% to 100% range provides a reasonable compromise to somewhat limit the oscillations, while allowing the time steps sufficient flexibility to respond to the evolution of the system. The results shown are with the nested difference scheme, but

Table 5.6: Single Pin Superprompt Error Comparison for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type

Crit. Type	Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	2016	101.40	-	-	-	0.061
CTS	-	100	2039	100.00	5.41	2.55	0.40	0.061
Absolute Direct	3.0e+00	96	2012	99.10	20.16	5.02	1.50	0.064
Relative Direct	5.0e-01	157	2269	102.66	14.10	3.34	0.75	0.063
Absolute Integrated	8.0e-03	83	2053	98.82	6.62	3.18	0.48	0.065
Relative Integrated	2.0e-03	87	2228	98.56	10.96	3.62	0.73	0.063

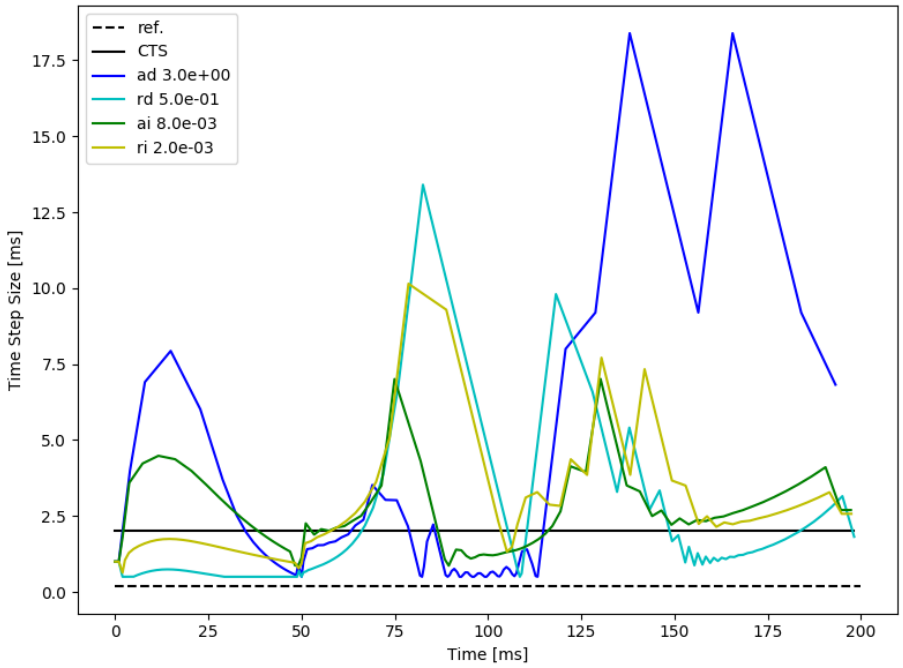

Figure 5.14: Single Pin Superprompt Time Step Size for ATS with Nested Finite Difference Method Using the Angular Flux for Each Criteria Type

similar results are observed with the interpolated difference scheme.

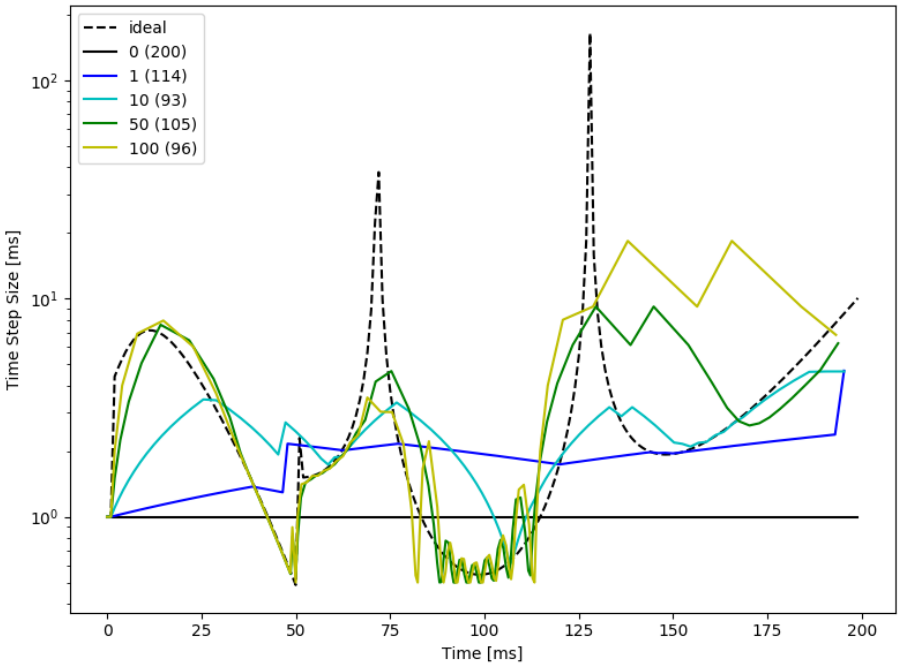

Figure 5.15: Single Pin Superprompt Time Step Size as a Function of Growth Limit for ATS with Nested Finite Difference Method Using the Angular Flux for Absolute Direct Criteria

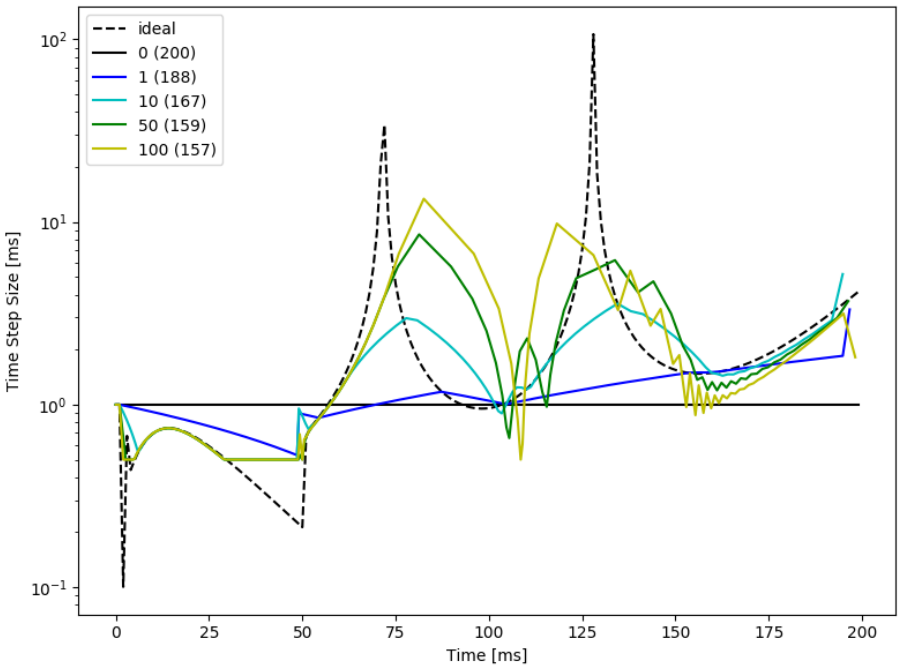

Figure 5.16: Single Pin Superprompt Time Step Size as a Function of Growth Limit for ATS with Nested Finite Difference Method Using the Angular Flux for Relative Direct Criteria

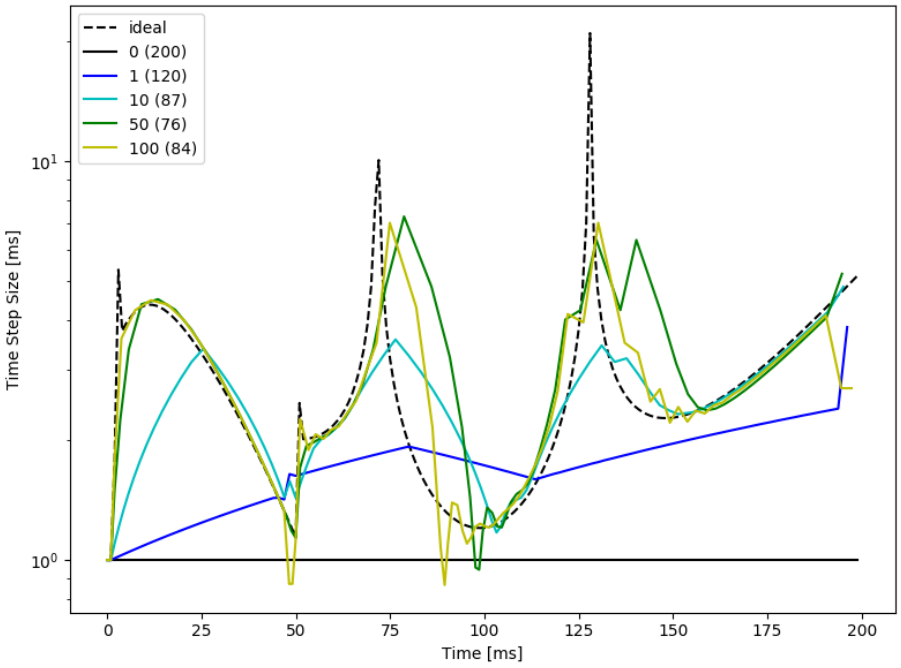

Figure 5.17: Single Pin Superprompt Time Step Size as a Function of Growth Limit for ATS with Nested Finite Difference Method Using the Angular Flux for Absolute Integrated Criteria

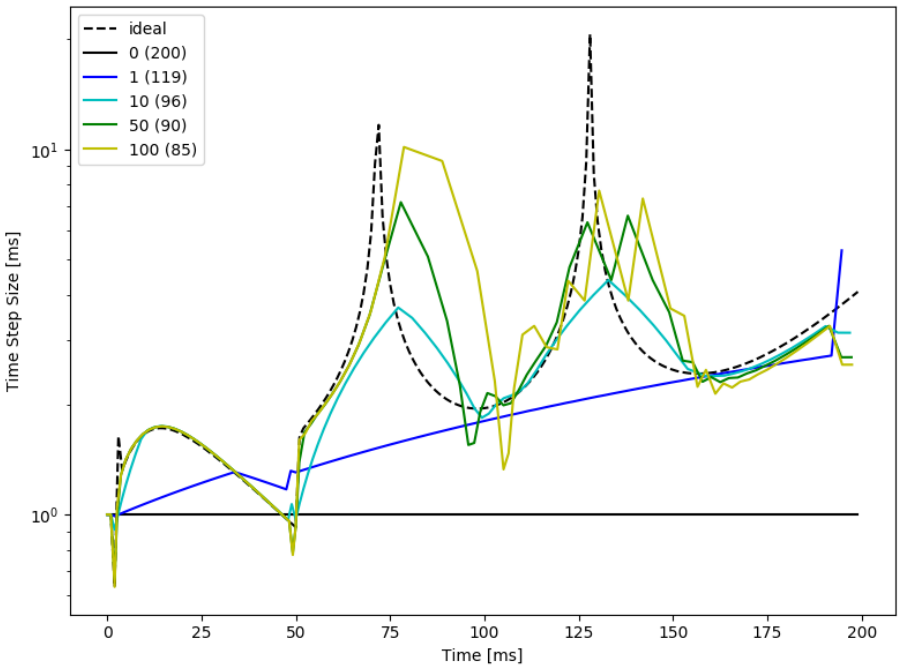

Figure 5.18: Single Pin Superprompt Time Step Size as a Function of Growth Limit for ATS with Nested Finite Difference Method Using the Angular Flux for Relative Integrated Criteria

5.2.3 Finite Difference Flux Choice

Once again, we will consider using the scalar flux in place of the angular flux for this more severe transient. The results presented in Figures 5.10 and 5.11 and Table 5.5 are reproduced in Figures 5.19 and 5.20 and Table 5.7 with the scalar flux used to estimate the second derivative rather than the angular flux.

The results for the scalar flux in Figures 5.19 and 5.20 and Table 5.7 are nearly identical to the results for the angular flux in Figures 5.10 and 5.11 and Table 5.5, indicating that the scalar flux is indeed a suitable substitute for the angular flux for ATS determination. We again take this one step further and consider using the power as a stand-in for the angular flux, as described in eq. (3.8). The validity of this approximation is examined in Figure 5.21 by plotting the logarithmic second derivatives (normalized by the value) of angular flux, scalar flux, and power. Note that these values are calculated for a case with constant time steps so as not to introduce any oscillations and distort the overall trend. The agreement is excellent, particularly between the angular flux and the scalar flux. For this simple pin cell case, the logarithmic second derivative of the power agrees well enough with the angular flux that we would expect it to yield good results. This is indeed the case, as we once again reproduce Figure 5.10 and 5.11 and Table 5.5 using the second derivative of the power instead of the angular flux in Figures 5.22 and 5.23 and Table 5.8.

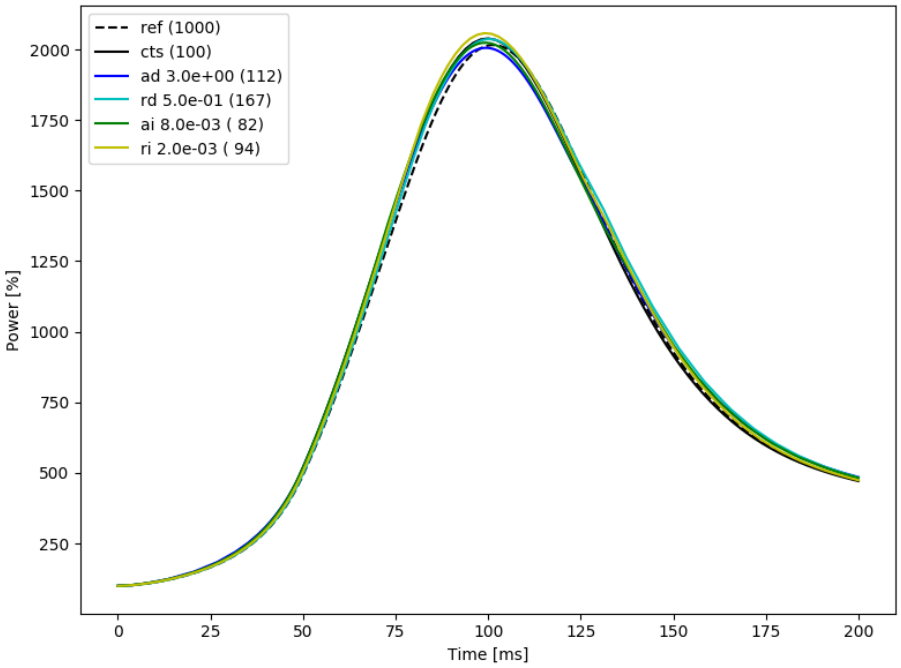

Figure 5.19: Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type, Time Steps Shown in Parentheses

Table 5.7: Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type

Crit. Type	Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	2016	101.40	-	-	-	0.061
CTS	-	100	2039	100.00	5.41	2.55	0.40	0.061
Absolute Direct	3.0e+00	112	2006	99.11	5.92	3.19	0.60	0.063
Relative Direct	5.0e-01	167	2037	99.90	5.19	1.71	0.40	0.063
Absolute Integrated	8.0e-03	82	2024	99.36	5.53	3.31	0.59	0.063
Relative Integrated	2.0e-03	94	2058	98.99	5.28	2.12	0.36	0.064

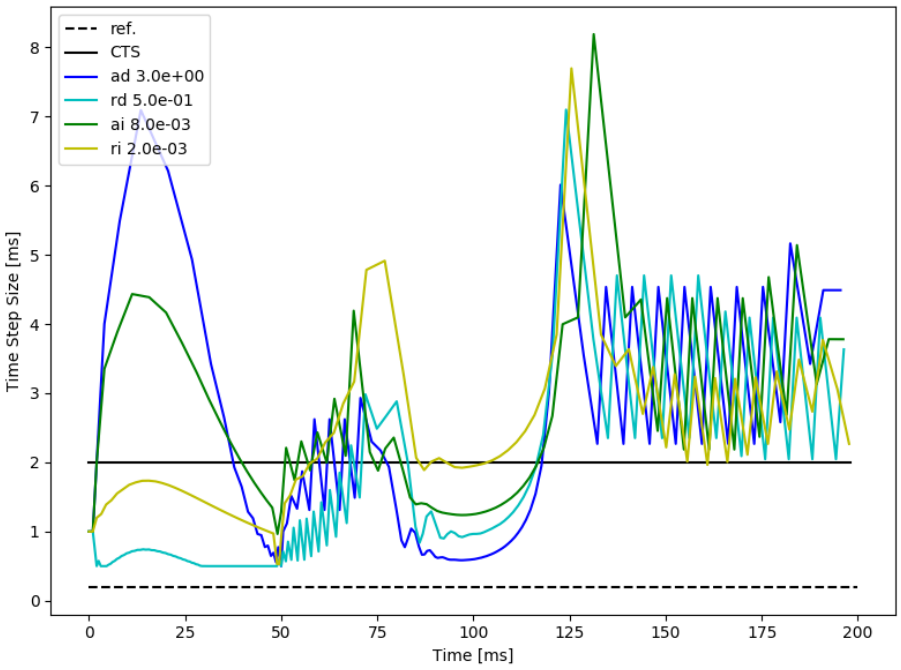

Figure 5.20: Single Pin Superprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Scalar Flux for Each Criteria Type

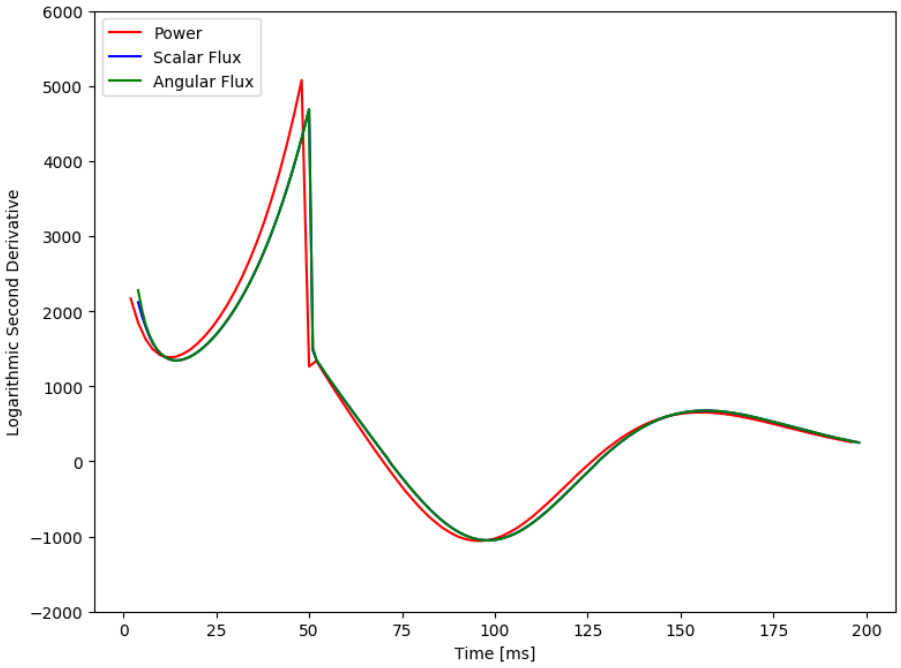

Figure 5.21: A Comparison of Logarithmic Second Derivative of Power, Scalar Flux, and Angular Flux for Single Pin Superprompt Transient

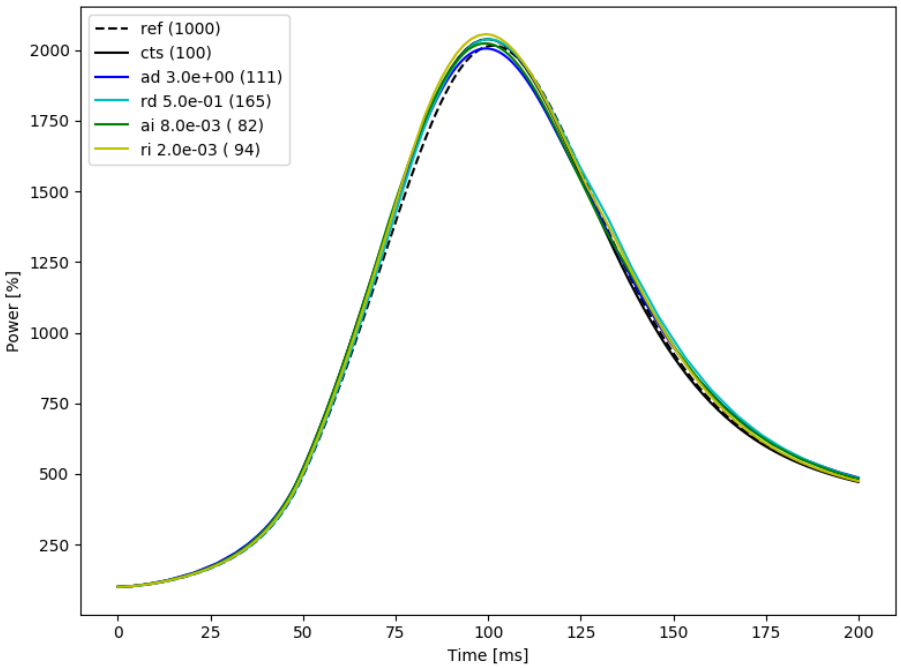

Figure 5.22: Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type, Time Steps Shown in Parentheses

Table 5.8: Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type

Crit. Type	Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	2016	101.40	-	-	-	0.061
CTS	-	100	2039	100.00	5.41	2.55	0.40	0.061
Absolute Direct	3.0e+00	111	2006	99.68	5.88	3.22	0.61	0.063
Relative Direct	5.0e-01	165	2038	99.86	5.43	1.78	0.42	0.062
Absolute Integrated	8.0e-03	82	2024	99.36	5.52	3.32	0.59	0.065
Relative Integrated	2.0e-03	94	2056	99.29	5.20	2.17	0.37	0.062

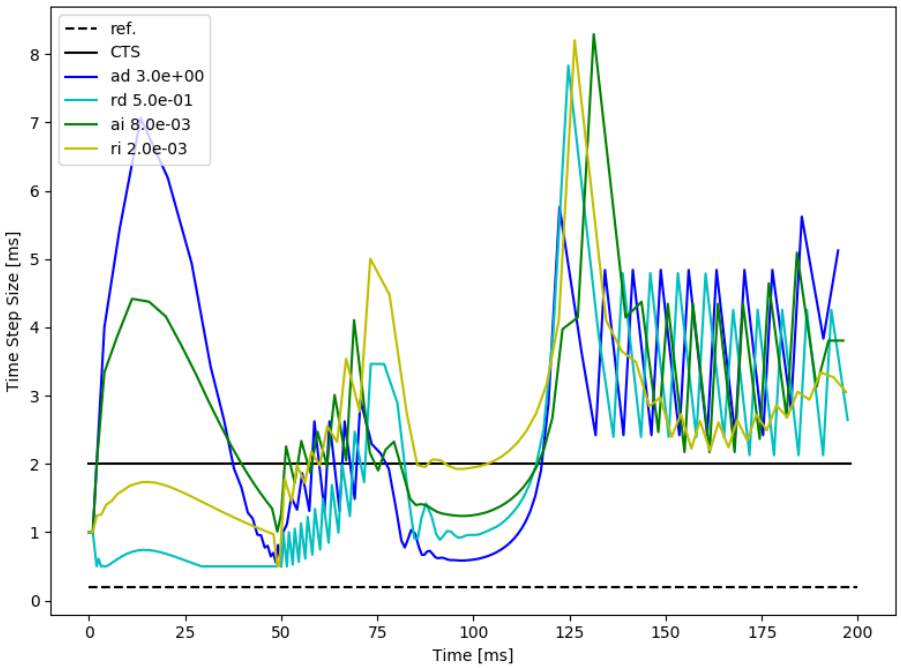

Figure 5.23: Single Pin Superprompt Time Step Size for ATS with Interpolated Finite Difference Method Using the Power for Each Criteria Type

5.3 Summary of Finite Difference Results

We have demonstrated that ATS in MPACT may be achieved via the time step selection criteria in eqs. (3.4) through (3.7) by using the finite difference method to estimate the second derivative. The method is prone to oscillations in the time step size with the standard "interpolated difference" scheme, but these can be resolved fully without feedback and partially with feedback by using the nested finite difference scheme proposed by Hackemack and Pounders. With feedback included, the nested difference scheme incurs an additional accuracy penalty, negating its utility. We have shown that the scalar flux may serve as a surrogate for the angular flux without significantly affecting the accuracy. Furthermore, the power itself may be an acceptable surrogate for cases without a sharp spatial dependence.

Finally, we have verified that the proposed ATS scheme works well in the absence of inflection points and can reduce run times by over 30%. However, the introduction of feedback assures the existence of such inflection points for superprompt transients. This challenges the underlying ATS scheme by violating our assumption that the higher order error terms are negligible, thereby predicting large time step sizes when the system is actually evolving rapidly. As a result, the ATS scheme incurs large error during a crucial time in the transient development, degrading the overall performance of the scheme.

Chapter 6

Adaptive Time Stepping Calculations with the α-Eigenvalue in MPACT

From the apparent difficulties of the finite difference ATS method, we turn to the α-eigenvalue. As for the finite difference method, the MPACT calculation begins with a steady state calculation to determine the initial conditions. Unlike the finite difference method, the α method can proceed after the first time step since it only requires data from one previous time point. After the first transient step is completed, the beginning of each subsequent step begins with a time step determination. The transient calculation is essentially "paused". This involves storing the current iterate and toggling a series of parameters within MPACT. In accordance with the results from Section 4.2, we begin the $k-\alpha$ iteration with the current flux and estimating α_0 as the α value from the previous time step. In the case of the initial α calculation, this would be 0. The α-eigenvalue is generally computed as per Algorithm 4 with the addition of the delayed neutron source directly from the transient solution. The fission source during the $k-\alpha$ calculation is multiplied by the steady state k calculated before initiating the transient. As for the transient solution itself, this is a requirement for consistency with the idea that the transient begins from a steady-state condition. Upon completion of the $k-\alpha$ iteration, the transient calculation is restored with only the α value remaining from the $k-\alpha$ calculation. Once an estimate for the second derivative (relative or absolute) is obtained, eqns. (3.4) through (3.7) are then used to determine the ensuing time step size. The transient calculation proceeds as usual. An alternative calculation of α from the EPKE will also be considered that is less computationally demanding.

6.1 α-Eigenvalue Calculation Method

While we have already demonstrated that MPACT is capable of calculating the α-eigenvalue in Chapter 4, both the prompt and asymptotic eigenvalues calculated correspond to a system in some sort of equilibrium. For the case of α_a, it as an equilibrium of asymptotic growth/decay; for α_p, the delayed neutron precursors are in equilibrium (no change in time). During an actual transient, neither of these conditions are fulfilled, casting doubt on the utility of these eigenvalues in accurately describing the time evolution of the system. Therefore, we will consider a modification to the α-eigenvalue calculation obtained directly from the transient

formulation of the transport equation solved by MPACT, eq. (2.29). By making the usual replacement of the time derivative with the α-eigenvalue, the time derivative terms disappear from the transient source term S^n_{tr} and are instead represented with augmentation of the total cross section by α/v. From the definitions in eqs. (2.30) and (2.31), this corresponds to setting $A(E) = 0$ and the second term of C to 0, so $C(\mathbf{r}, E) = \chi_d(\mathbf{r}, E)\tilde{S}^{n-1}_d(\mathbf{r})$. Hence the equation being solved is:

$$\mathbf{\Omega} \cdot \nabla \psi^n(\mathbf{r}, \mathbf{\Omega}, E) + \left[\Sigma_t(\mathbf{r}, E) + \frac{\alpha^n}{v(E)}\right] \psi^n(\mathbf{r}, \mathbf{\Omega}, E) = \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{r}, E' \to E, \mathbf{\Omega} \cdot \mathbf{\Omega}')\psi^n(\mathbf{r}, \mathbf{\Omega}', E')ld\Omega' dE' + \frac{\chi^n_{tr}(\mathbf{r}, E)}{4\pi} \int_0^\infty \int_{4\pi} \nu\Sigma_f(\mathbf{r}, E')\psi^n(\mathbf{r}, \mathbf{\Omega}', E')d\Omega' dE' + \chi_d(\mathbf{r}, E)\tilde{S}^{n-1}_d(\mathbf{r}), \quad (6.1)$$

where

$$\chi^n_{tr} = \chi^n_p(\mathbf{r}, E)(1 - \beta) + \omega^n(\mathbf{r})\chi^n_d(E). \quad (6.2)$$

To differentiate this version of the α-eigenvalue from α_a and α_p calculated for the steady-state case, we will refer to this as the transient α-eigenvalue, α_t. For confirmation that this is the most desirable formulation of the α-eigenvalue, we calculate α_a, α_p, and α_t for the single pin superprompt transient and compare them to the time constant (logarithmic first derivative) of the power in Figure 6.1. This leaves no doubt that α_t is the quantity of interest for use with ATS.

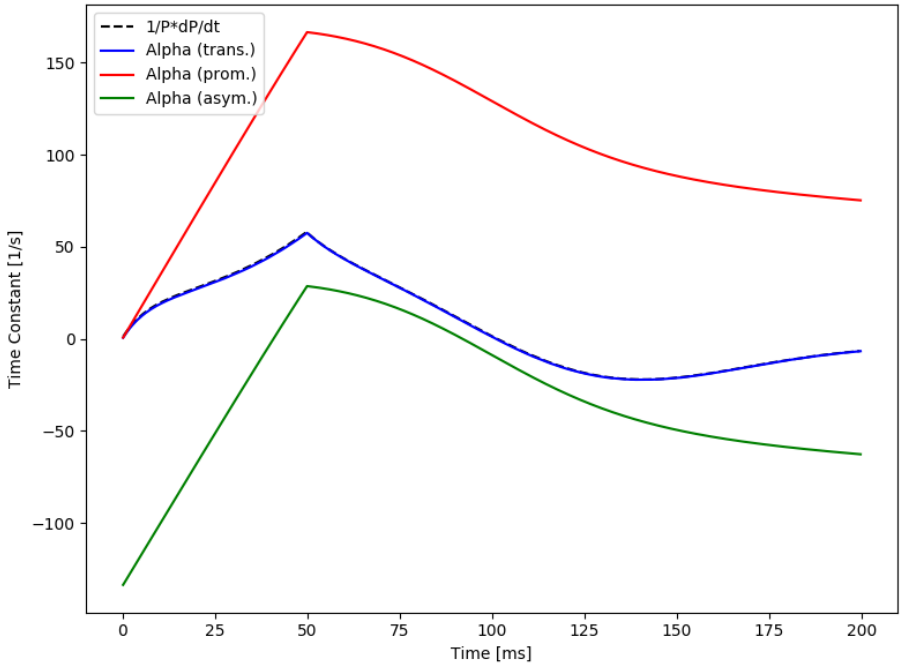

Figure 6.1: A Comparison of the Different α-Eigenvalues and the Logrithmic Derivative of Power for Single Pin Superprompt Transient

Since α_p and α_a were predicted so well by the PKE, we might expect that α_t can also be calculated directly from the PKE at a much lower computational expense than the full $k - \alpha$ iteration. This possibility will

be explored and compared to the full $k-\alpha$ iteration results. Further discussion of the α-eigenvalue will be centered on α_t and we will therefore drop the subscript.

6.2 α-Eigenvalue Results without Feedback

As for the finite difference method, we begin by examining the subprompt positive reactivity insertion described in Section 5 and 5.1 without thermal-hydraulic feedback.

6.2.1 Adaptive Time Stepping Criteria with the α-Eigenvalue without Feedback

Once again, we start by studying the four different criteria defined in eqs. (3.4) through (3.7) for a range of tolerances with an initial time step size of 2.5 ms, a minimum time step size of 0.5 ms, a maximum time step size of 20 ms, and a growth limit of 100%. The resulting power curves are shown in Appendix A Figures A.9, A.10, A.11, A.12 for the absolute direct, relative direct, absolute integrated, and relative integrated criteria, respectively. Quantification of the errors with respect to the reference solution are shown in Appendix A Tables A.9, A.10, A.11, and A.12 as well. For comparison, a 5 ms constant time step case is also shown in the tables and figures. This is intended to provide a reference point for what we might expect from a reasonable simulation that balances the need for accuracy and computational efficiency.

We observe the same general trends for the α-eigenvalue ATS as for the finite difference ATS. As the tolerances are tightened, the number of time steps increases and approaches the reference solution. Once again, we compare the criteria by optimizing the tolerance for each criteria to most closely match the errors of the 5 ms CTS case. The results are plotted in Figure 6.2 and summarized in Table 6.1. Without feedback, there is no meaningful difference between the various criteria other than the obvious observation that the direct criteria are more sensitive to the tolerance than relative criteria. Also similar to the finite difference method, all criteria yield a 30-35% reduction in the number of time steps required to achieve similar accuracy as the CTS case. Unfortunately, the $k-\alpha$ iteration drives the computational cost of each time step up by a similar amount, resulting in only a net 5-10% reduction in transient run time. This is not a surprising result, and actually offers some encouragement since there is still a net reduction in time step size.

The main driver for investigating the α-eigenvalue as an alternative to the finite difference method is the oscillations of the predicted time step observed for the latter method. An examination of the time steps predicted by the α-eigenvalue is shown in Figure 6.3. The encouraging feature of this plot is that the oscillations observed for the finite difference method have indeed been resolved.

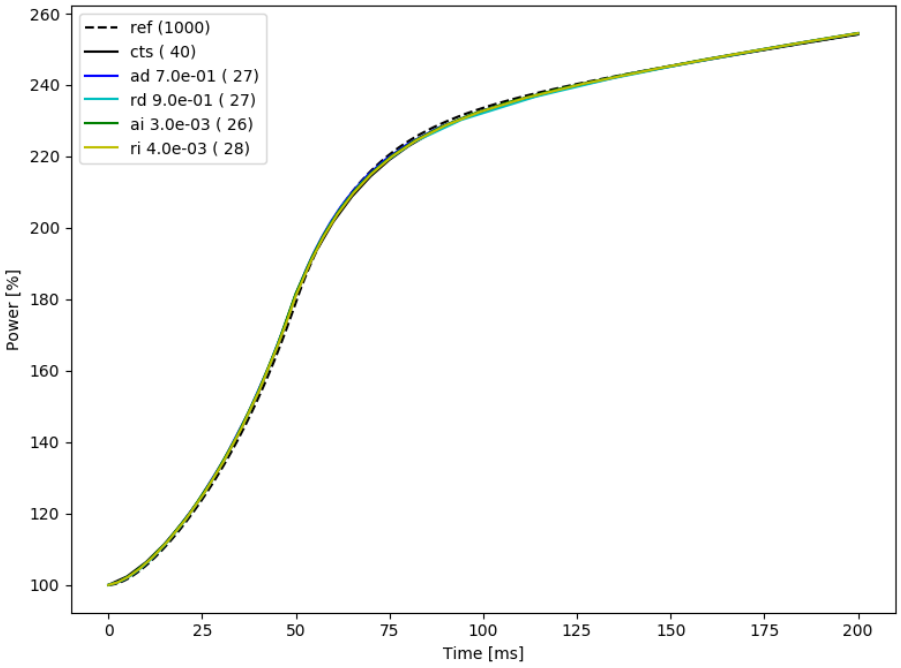

Figure 6.2: Single Pin Subprompt Power Curves for ATS with α Method Using the Angular Flux for each Criteria Type, Time Steps Shown in Parentheses

Table 6.1: Single Pin Subprompt Error Comparison for ATS with α Method Using the Angular Flux for each Criteria Type

Crit. Type	Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	-	-	-	0.047
CTS	-	40	1.22	0.52	0.08	0.058
Absolute Direct	7.0e-01	27	1.27	0.62	0.08	0.076
Relative Direct	9.0e-01	27	1.22	0.57	0.08	0.083
Absolute Integrated	3.0e-03	26	1.39	0.62	0.08	0.079
Relative Integrated	4.0e-03	28	1.08	0.56	0.07	0.080

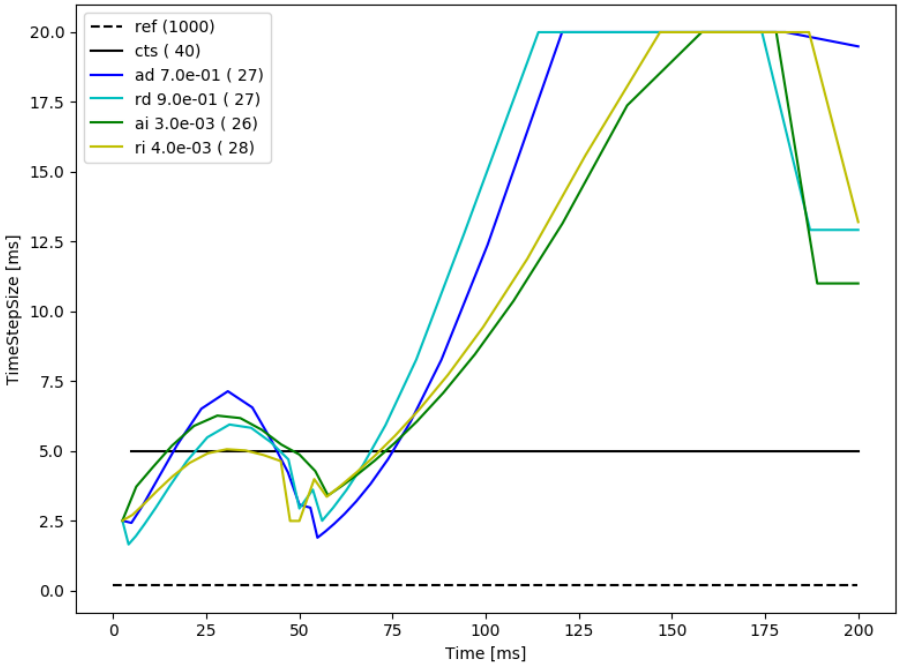

Figure 6.3: Single Pin Subprompt Time Step Size for ATS with α Method Using the Angular Flux for each Criteria Type

6.2.2 α-Eigenvalue Flux Choice

Since the α-eigenvalue calculation does not require the angular flux, we would like to substitute the scalar flux for the angular flux in the second derivative estimates given in eqs. (3.12) and (3.13). For the absolute criteria, this requires division by 4π, but this is unnecessary for the relative criteria. The cases from Table 6.1 are run with the scalar flux in lieu of the angular flux, with the results shown in Table 6.2. The power curves are not reproduced since there is no visually discernible difference. This reduces the run time as well as the storage requirements without any noticable effect on accuracy.

Table 6.2: Single Pin Subprompt Time Comparison for ATS with α Method Using the Scalar Flux for each Criteria Type

Crit. Type	Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	-	-	-	0.047
CTS	-	40	1.22	0.52	0.08	0.058
Absolute Direct	7.0e-01	26	1.49	0.65	0.08	0.072
Relative Direct	9.0e-01	27	1.22	0.57	0.08	0.073
Absolute Integrated	3.0e-03	25	1.33	0.64	0.08	0.073
Relative Integrated	4.0e-03	28	1.08	0.56	0.07	0.071

6.2.3 Calculation of the α-Eigenvalue by Point Kinetics for ATS without Feedback

In the interest of computational efficiency, an alternative calculation of the α-eigenvalue by means of the PKE was implemented. The PKE coefficients ($\rho(t)$, $\beta(t)$, $\lambda(t)$ and $\Lambda(t)$) are already computed at each time step. They are then linearly interpolated over the time step and the resulting PKEs are solved via a quadratic method on a fine time step. The solution of this 1D system of equations has negligible computational overhead compared to the MOC calculations, and the final few fine steps can be used to determine the α-eigenvalue for the end of the time step. The cases from Table 6.1 are run with this alternative methodology and the results are shown in Table 6.3. The results show no loss of accuracy with run times per time step that are essentially identical to CTS. The α-eigenvalues as calculated by the full $k-\alpha$ and PKE methodologies are shown in Figure 6.4, as are the resulting second derivatives in Figure 6.5. The results show excellent agreement without feedback.

Table 6.3: Single Pin Subprompt Time Comparison for ATS with PKE α Method Using the Scalar Flux for each Criteria Type

Crit. Type	Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	-	-	-	0.047
CTS	-	40	1.22	0.52	0.08	0.058
Absolute Direct	7.0e-01	28	1.18	0.57	0.09	0.055
Relative Direct	9.0e-01	28	1.01	0.53	0.09	0.064
Absolute Integrated	3.0e-03	25	1.30	0.61	0.09	0.054
Relative Integrated	4.0e-03	28	0.94	0.54	0.08	0.041

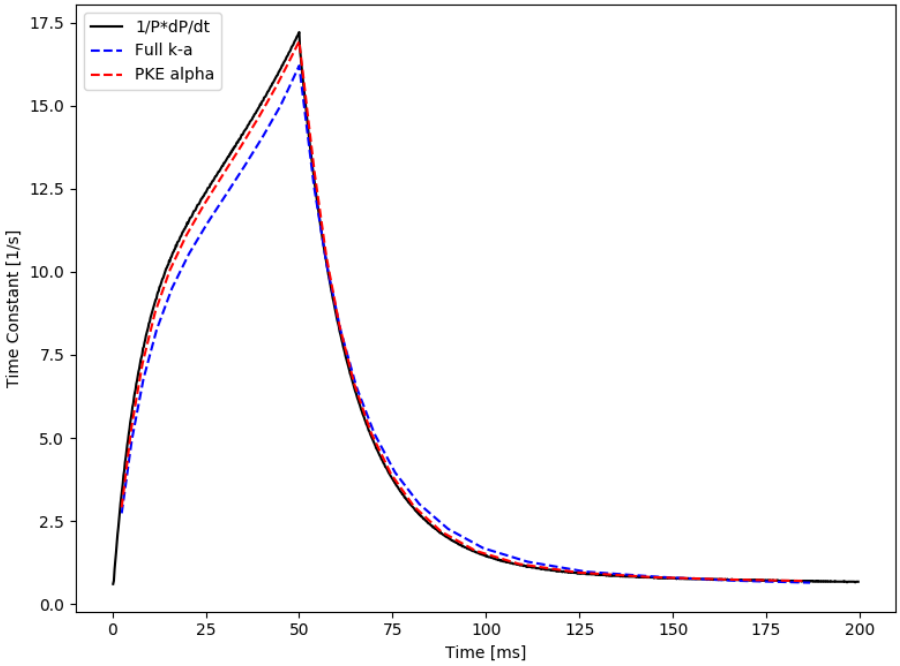

Figure 6.4: A Comparison of Time Constants for the α-Eigenvalue Calculation Methods without Feedback

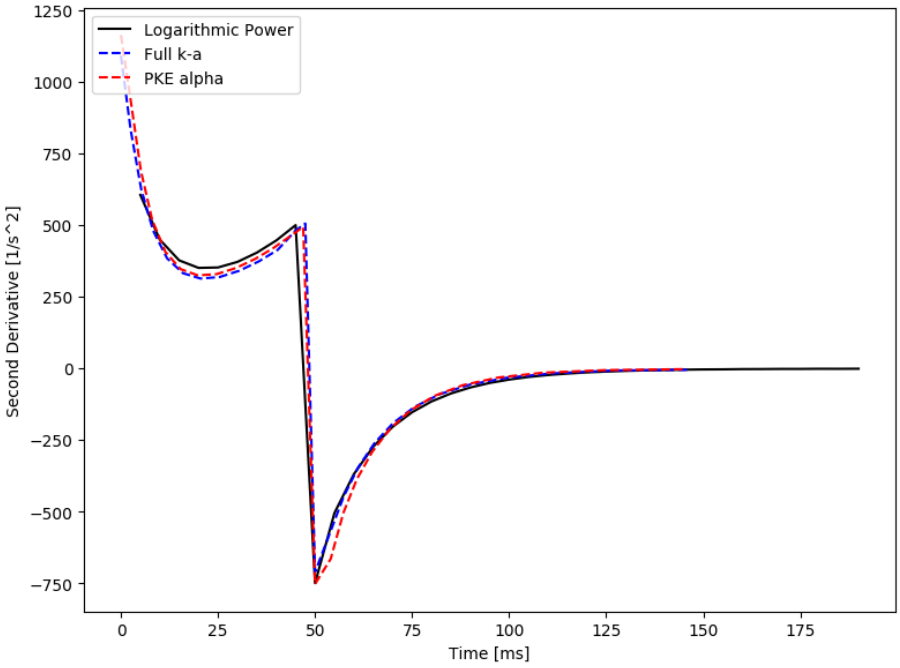

Figure 6.5: A Comparison of Second Derivatives for the α-Eigenvalue Calculation Methods without Feedback

6.3 α-Eigenvalue Results with Feedback

6.3.1 Adaptive Time Stepping Criteria with the α-Eigenvalue

Yet again, we begin by studying the four different criteria defined in eqs. (3.4) through (3.7) for a range of tolerances with an initial time step size of 1 ms, a minimum time step size of 0.5 ms, a maximum time step size of 20 ms, and a growth limit of 100%. The resulting power curves are shown in Appendix A Figures A.13, A.14, A.15, A.16 for the absolute direct, relative direct, absolute integrated, and relative integrated criteria, respectively. Quantification of the errors with respect to the reference solution are shown in Appendix A Tables A.13, A.14, A.15, and A.16 as well. For comparison, a 5 ms constant time step case is also shown in the tables and figures. This is intended to provide a reference point for what we might expect from a reasonable simulation that balances the need for accuracy and computational efficiency. We compare the criteria by optimizing the tolerance for each criteria to most closely match the errors of the 2 ms CTS case. The results are plotted in Figure 6.6 and summarized in Table 6.4.

We observe the same general trends for the α-eigenvalue method as for the finite difference method with feedback. The direct criteria appear to be overly sensitive to the second derivative in comparison to the integrated criteria, resulting in more time steps being required to obtain a similar level of accuracy. As for the finite difference method, the α-eigenvalue method is challenged by the inflection points where the second derivative is zero (or nearly so) but the higher order error terms are not negligible. This is examined further in Section 6.3.3. An examination of the time steps predicted by the α-eigenvalue is shown in Figure 6.7. Here we note the absence of oscillations, a much desired improvement from the finite difference method. Recall that even the nested difference scheme was unable to resolve oscillations fully for the superprompt transient with feedback.

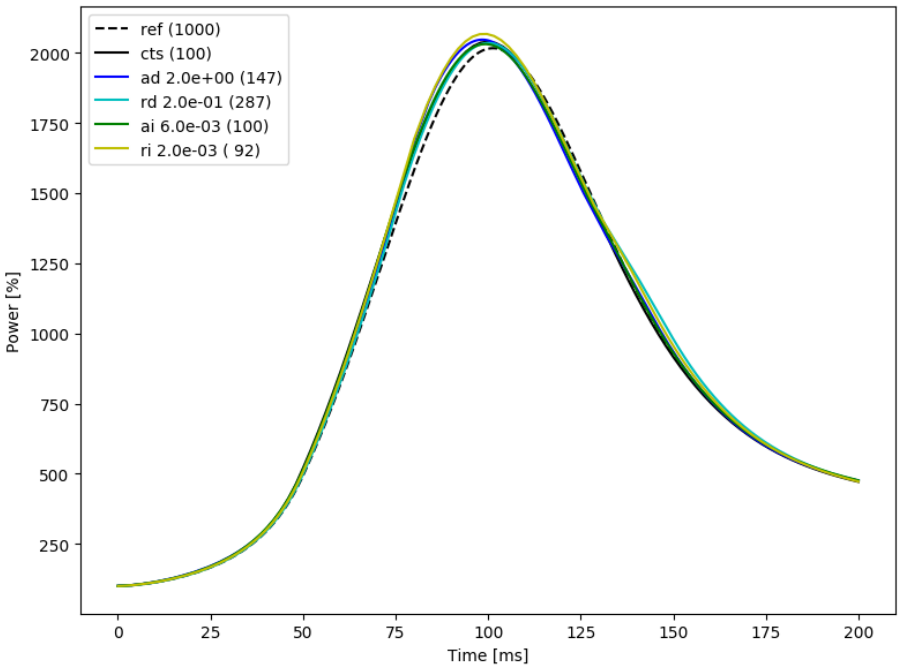

Figure 6.6: Single Pin Superprompt Power Curves for ATS with α Method Using the Angular Flux for each Criteria Type, Time Steps Shown in Parentheses

Table 6.4: Single Pin Superprompt Error Comparison for ATS with α Method Using the Angular Flux for each Criteria Type

Crit. Type	Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	2016	101.40	-	-	-	0.063
CTS	-	100	2039	100.00	5.41	2.55	0.40	0.064
Absolute Direct	2.0e+00	147	2046	98.20	6.48	2.60	0.41	0.097
Relative Direct	2.0e-01	287	2034	100.08	5.88	1.94	0.32	0.104
Absolute Integrated	6.0e-03	100	2031	99.34	5.39	2.66	0.40	0.104
Relative Integrated	2.0e-03	92	2067	99.64	6.98	2.52	0.43	0.104

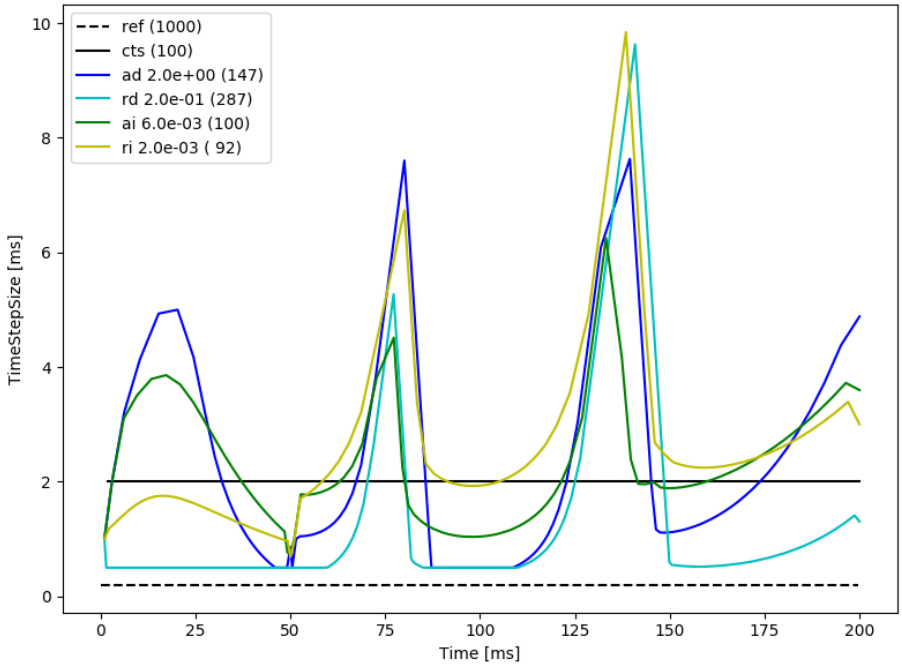

Figure 6.7: Single Pin Superprompt Time Step Size for ATS with α Method Using the Angular Flux for each Criteria Type

6.3.2 α-Eigenvalue Calculation by Point Kinetics with Feedback

Again, we explore employing an alternative calculation of the α-eigenvalue by means of the PKE in order to reduce the computational expense of estimating the second derivative. The cases from Table 6.4 are run with this alternative methodology and the results are shown in Table 6.5. The results are decidedly mixed with the direct criteria actually showing a general improvement in accuracy while the integrated criteria is degraded. The run times per time step are again reduced to on par with CTS. We plot the time steps predicted by the PKE method in Figure 6.9 and immediately notice a worrying feature– substantial oscillation. This is particularly pronounced for the direct criteria.

We explore this further by plotting the α-eigenvalues for the relative integrated criteria with the full $k - \alpha$ and PKE methods and comparing them to the logarithmic first derivative of power in Figure 6.10, as well as the resulting second derivative in Figure 6.11. Unlike without feedback, we notice that α and the second derivative as calculated by the PKEs deviate meaningfully from the expected value. This deviation in α is then amplified by the second derivative calculation, introducing the unwanted oscillations.

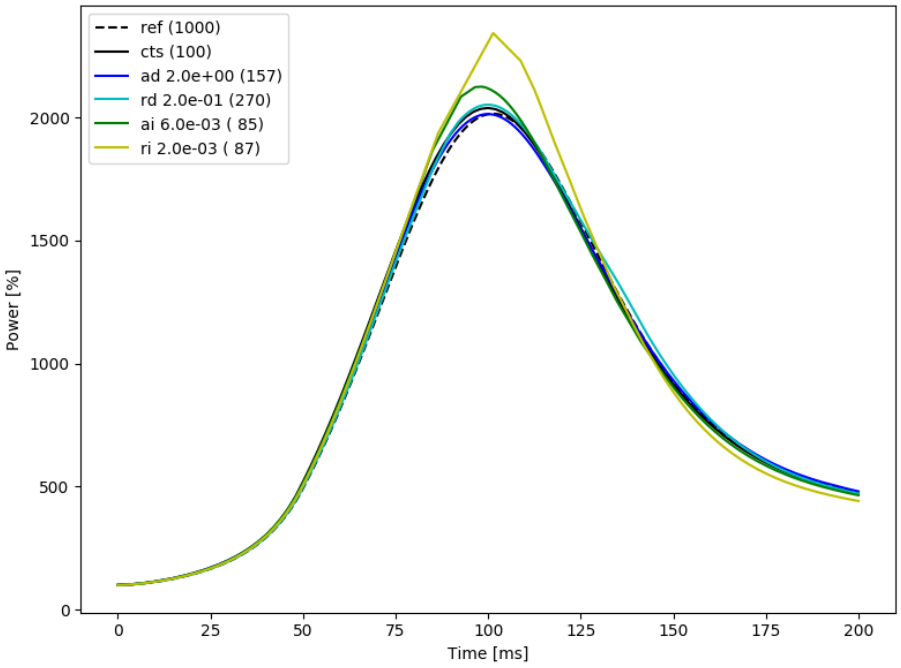

Figure 6.8: Single Pin Superprompt Power Curves for ATS with α Method Using the Scalar Flux for each Criteria Type, Time Steps Shown in Parentheses

Table 6.5: Single Pin Superprompt Error Comparison for ATS with α Method Using the Scalar Flux for each Criteria Type

Crit. Type	Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	-	1000	2016	101.40	-	-	-	0.063
CTS	-	100	2039	100.00	5.41	2.55	0.40	0.064
Absolute Direct	2.0e+00	157	2014	100.12	3.11	1.86	0.34	0.064
Relative Direct	2.0e-01	270	2052	100.00	3.87	1.27	0.23	0.062
Absolute Integrated	6.0e-03	85	2126	98.35	7.61	3.37	0.63	0.064
Relative Integrated	2.0e-03	87	2343	101.34	16.25	5.10	1.01	0.067

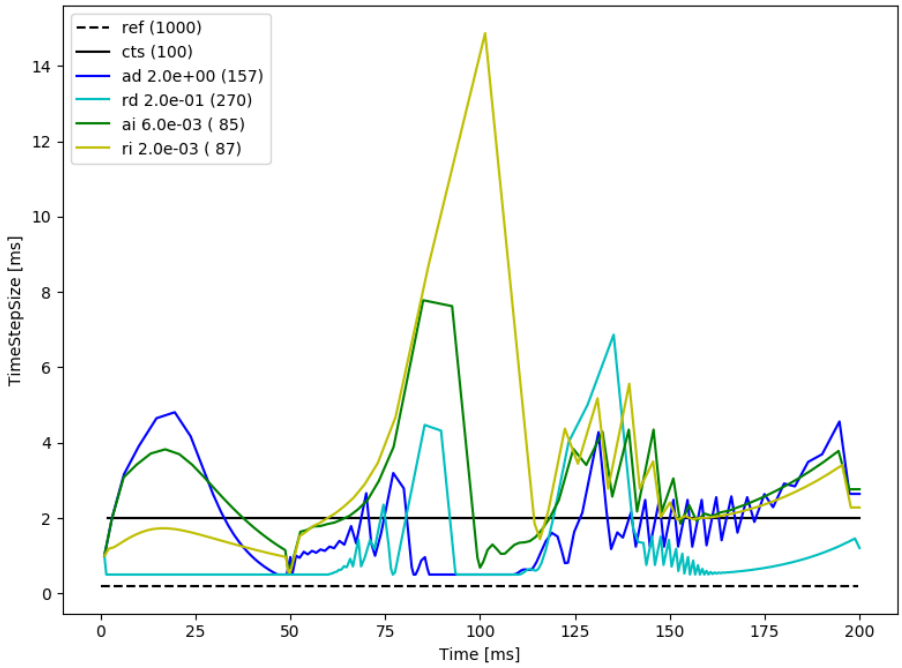

Figure 6.9: Single Pin Superprompt Time Step Size for ATS with α Method Using the Scalar Flux for each Criteria Type

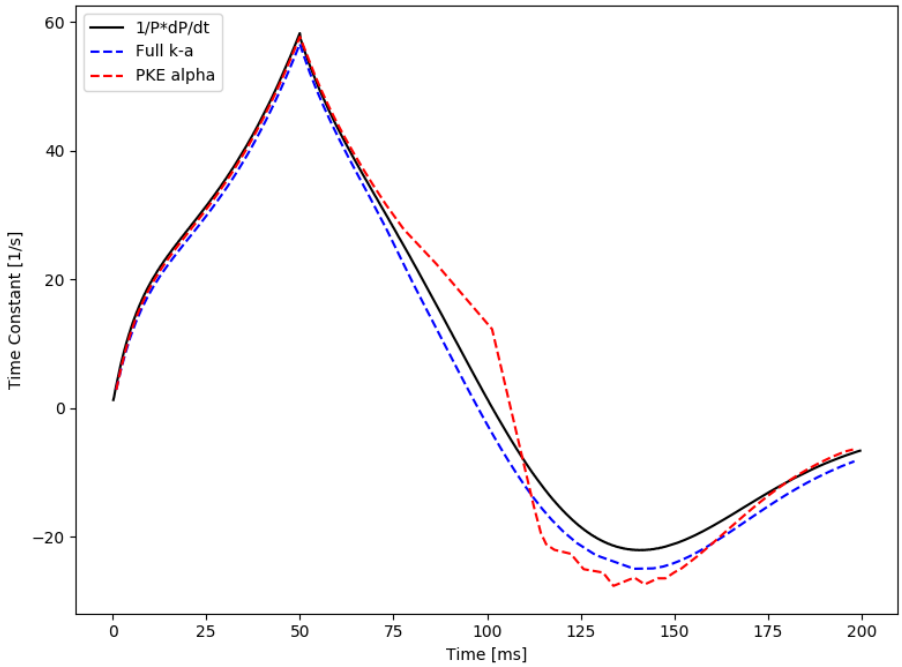

Figure 6.10: A Comparison of Time Constants for the α-Eigenvalue Calculation Methods with Feedback

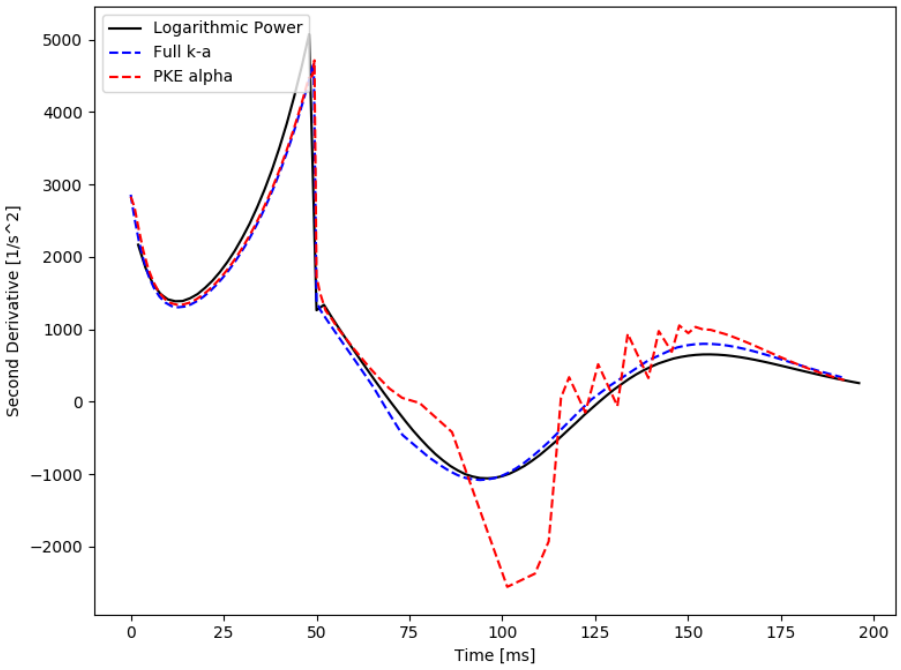

Figure 6.11: A Comparison of Second Derivatives for the α-Eigenvalue Calculation Methods with Feedback

6.3.3 Examination of the Role of Feedback

In order to understand the poor performance of the ATS methodology for the superprompt transient with feedback, we examine the relative integrated criteria results from Figure A.16 and Table A.16 more closely. Note that this is applicable to the ATS scheme in general, not the specific α or finite difference formulation of it. While all of the criteria demonstrate the same basic trends, the relative integrated criteria provide the clearest example. First we examine the evolution of the α-eigenvalue through the transient to ensure that it is consistent with the time constant. The time constant of the power and the α-eigenvalues are shown in Figure 6.12, showing excellent agreement as the tolerance is tightened.

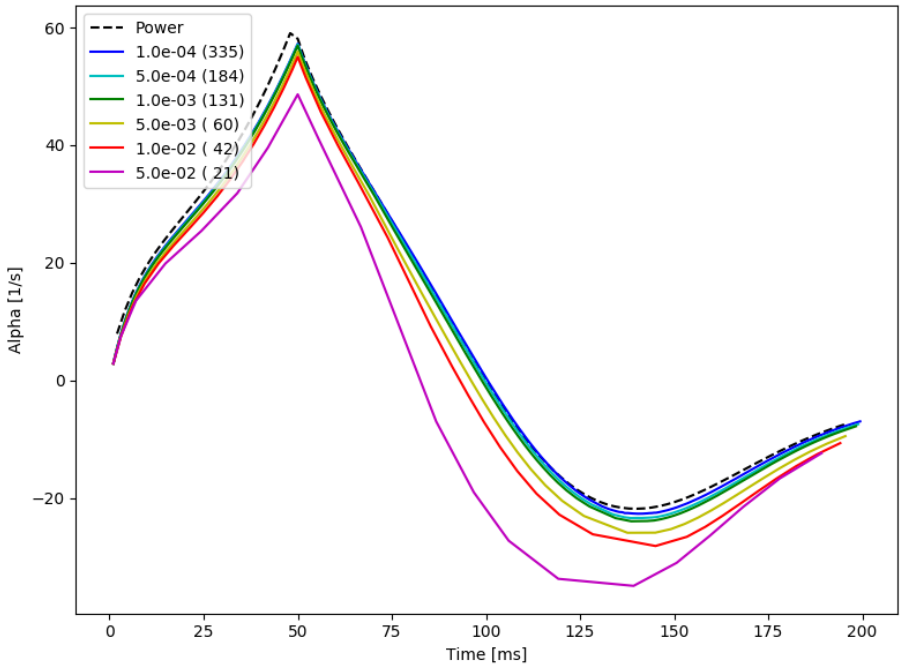

Figure 6.12: Single Pin Superprompt α-Eigenvalue and Transient Time Constant for ATS with Relative Integrated Criteria

Next, we turn to an examination of the second derivative. This is shown in Figure 6.13. We see that the second derivative crosses the y-axis at around 75 and 125 ms, corresponding to the inflection points. Where the second derivative is nearly zero, but changing rapidly, as occurs at 75 ms, the higher order terms that were neglected in the Taylor expansion may actually contribute substantially to the error. As was discussed in Section 5.2.1, physically, the inflection point represents the departure from the purely exponential behavior of the neutronic solution to the feedback-dominated peak behavior. From this physical interpretation, it is obvious that this represents a critical time in the development of the transient. This intuition is confirmed by plotting the reference rate of change of the fuel temperature in Figure 6.14, where we note the fuel temperature is rising rapidly at 75 ms. In order to investigate the relative performance of ATS, Figure 6.15 plots the error in the fuel temperature rate of change for CTS and a single ATS case (relative integrated criteria with a tolerance of 1.0E-3) compared to the reference case. Here we see that the ATS case, chosen because it produces similar accuracy to the CTS case but requires more 30% more time steps, does a better

job of capturing the rate of fuel temperature change up until 75 ms. By taking a large time step in that region, the feedback is being improperly accounted for, to the detriment of the accuracy at large. The accuracy of the solution cannot recover from this critical misstep without sacrificing the computational efficiency of the method.

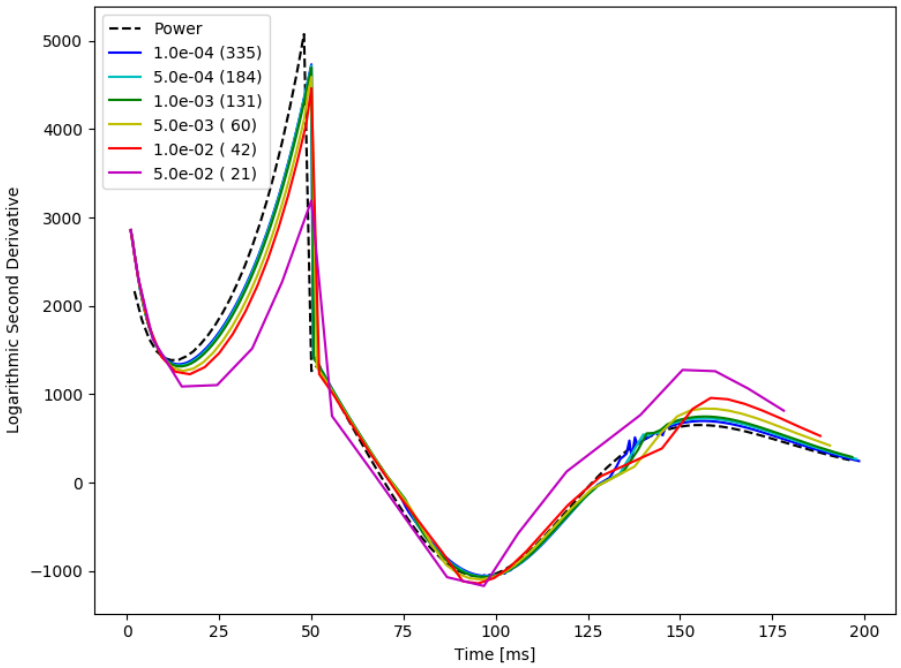

Figure 6.13: Single Pin Superprompt Logarithmic Second Derivative Estimated from α for ATS with Relative Integrated Criteria

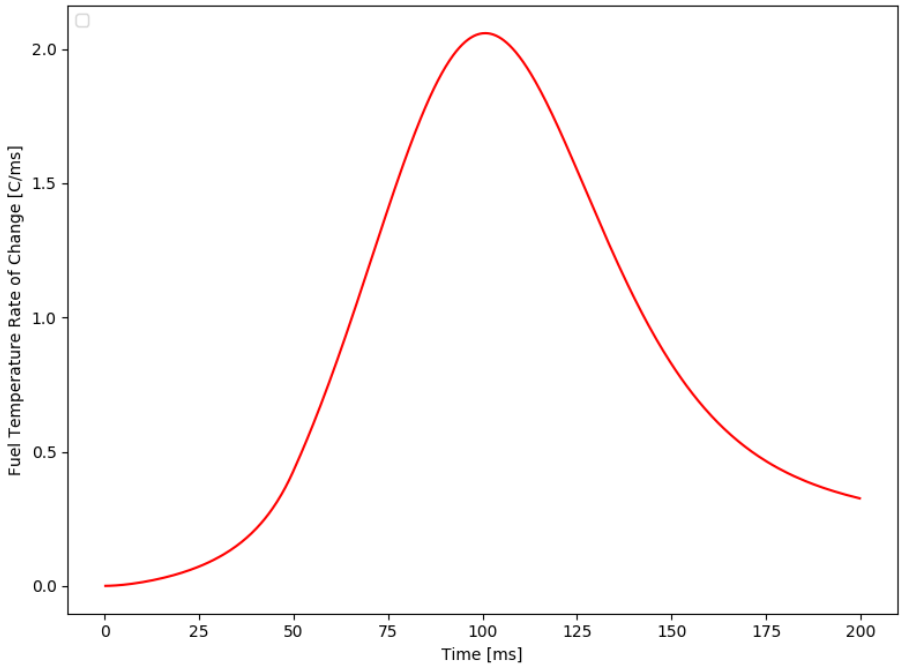

Figure 6.14: Single Pin Superprompt Reference Fuel Temperature Rate of Change

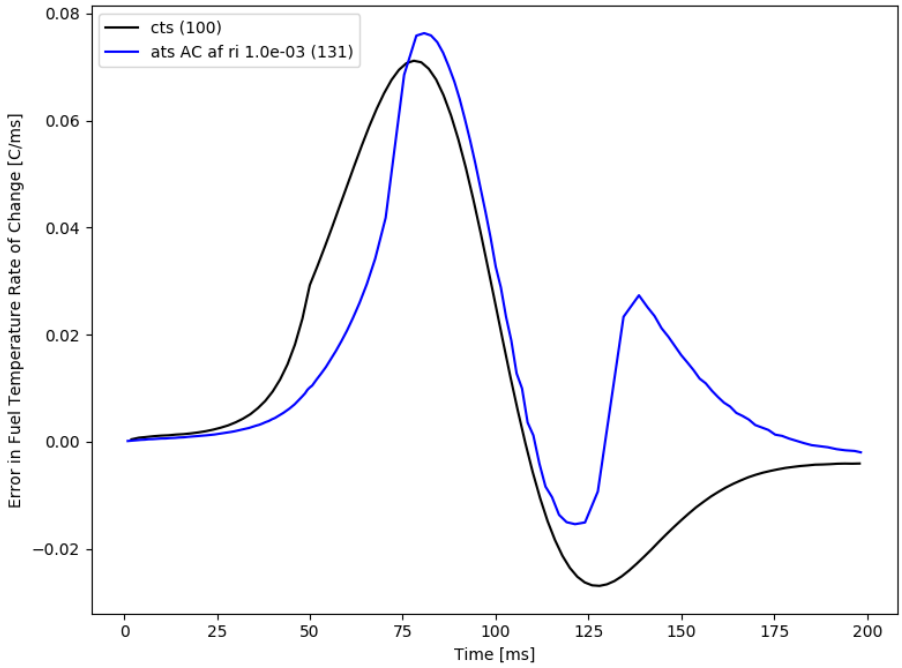

Figure 6.15: Single Pin Superprompt Error in Fuel Temperature Rate of Change

6.4 Summary

MPACT has been modified to calculate the α-eigenvalue via $k - \alpha$ iteration and via the PKEs during the transient in order to estimate the second derivative of the angular flux for use with ATS. The α-eigenvalue method is able to perform at the same level as the finite difference method in terms of accuracy and time steps required *without* oscillations in the time step size. However, the computational expense of the α-eigenvalue is significantly higher than the finite difference method. Use of the point kinetics equations to estimate the α-eigenvalue can be used in place of the full $k - \alpha$ iteration. This method brings the computational cost in line with the finite difference method while also avoiding oscillatory behavior in the case of no feedback. However, with feedback, the oscillations reappear with the less robust PKE method of α calculation.

The α ATS method does not improve upon the finite difference method for handling inflection points of superprompt transients with feedback since this is a weakness of the underlying ATS scheme. It does, however, remove the oscillations from the time step size, providing a more efficient and desirable ATS scheme overall.

Chapter 7

Benchmarks Calculations

The initial investigations of the ATS methodologies presented in Chapters 5 and 6 utilize a single pin to facilitate detailed examination of the results. However, we are generally interested in modeling more realistic cases, so we apply the results from the preceding chapters to two more complex models– the SPERT mini-core and the C5G7-TD benchmark problem TDW1b.

7.1 SPERT Mini-Core

The SPERT III E-Core reactor was an experimental reactor in Idaho in the 1960's. It was specifically designed for reactor transient experiments and provides some of the only available data to benchmark full scale transient calculations against. As such, it has been an important aspect of validation of MPACT transient capabilities. The author produced KENO models for direct comparison to MPACT as part of the early validation work [9]. These were then tested with the MPACT internal thermal-hydraulics model [14] and eventually coupled to the thermal-hydraulics code COBRA-TF by the author [19].

Here we test the ATS capability on a 2D mini-core version of the SPERT experiments that models 16 fuel assemblies with quarter-core symmetry. Details of the SPERT configuration and experiments can be found in any of the aforementioned references or the original technical reports [18], [44], and [36]. The 2D SPERT mini-core encompasses the 16 central fuel assemblies modeled in quarter-core symmetry. The resulting model consists of four fuel assemblies, which include two of the nominal 5x5 assemblies, one of the control rod follower 4x4 assemblies, and one of the transient 4x4 assemblies, shown in Figure 7.1. For all assemblies, each pin consists of 4.8 wt% UO_2 fuel with a diameter of 1.0068 cm and 0.0508 cm thick stainless steel cladding surrounded by water with a pitch of 1.4859 cm. Each assembly is enclosed in a steel channel box with an overall assembly pitch of 7.62 cm. The transient assembly in the lower right corner of Figure 7.1 also contains one quarter of the cruciform transient rod composed of 1.35 wt% borated steel. It is the removal of this rod that drives the transient. These four assemblies are superimposed into the corner of the actual reactor to capture the steel and water reflector regions surrounding the core, that accounts for the somewhat choppy nature in Figure 7.1. This model presents an excellent test case for our methods owing to the sharp spatial heterogeneity. The transient is induced by withdrawing the transient rod over 50 ms; this

amounts to a reactivity insertion of \$1.10.

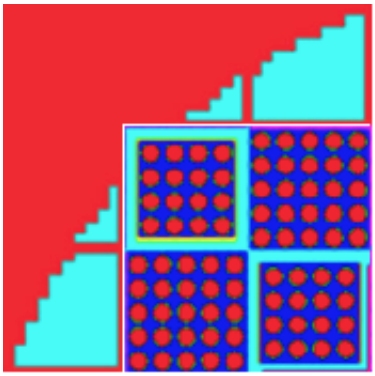

Figure 7.1: SPERT Mini-core Overall Configuration

7.1.1 SPERT Mini-Core Results

The SPERT cases are run with the MPACT simplified TH for 0.2 s. The 47-group library is employed with 16 azimuthal angles with the Chebyshev azimuthal quadrature. CMFD acceleration is used but the multilevel method is not. All cases were run in parallel on 9 processors. An initial time step size of 2 ms was utilized with a minimum of 1 ms, a maximum of 50 ms, and a growth limit of 100%.

As for the single pin studies, MPACT was run with each criteria type at a tolerance that produces similar errors as the CTS case with 2 ms time steps. The resulting power curves are shown in Figure 7.2 and the error quantification in Table 7.1. The results are generally consistent with what we have observed for the single pin cases. The reduction in time steps is greater for this transient, mostly owing to the sharper power pulse, which has a longer tail that allows the ATS scheme to really show its value. However, the more complex geometry degrades the performance of the $k-\alpha$ calculation, resulting in a significant increase in the overall run time per time step. The net result is a noticeable increase in the overall run time. We next investigate using the PKE estimation of α in lieu of the full $k-\alpha$ calculation, but before we do, let us comment on the performance of the individual criteria.

As we have noted before, the direct criteria appear to be overly sensitive to changes in the second derivative and generally require far more time steps to produce a solution of comparable accuracy when compared to the integrated criteria when there is thermal-hydraulic feedback. The question of absolute versus relative criteria does not have as clear of a "winner." However, if we examine the time steps predicted by the methods, as in Figure 7.3, we note that both of the absolute criteria show a proclivity for oscillations. Furthermore, the large increases at the beginning of the transient present opportunity for error to accumulate and propagate early on. This is further evidenced by examination of the α-eigenvalues computed by the methods compared to the logarithmic time derivative of power shown in Figure 7.4, where we note a significant deviation for the absolute criteria early in the transient. As such, it appears that the relative integrated criteria is the best choice, though this conclusion is not universally demonstrative.

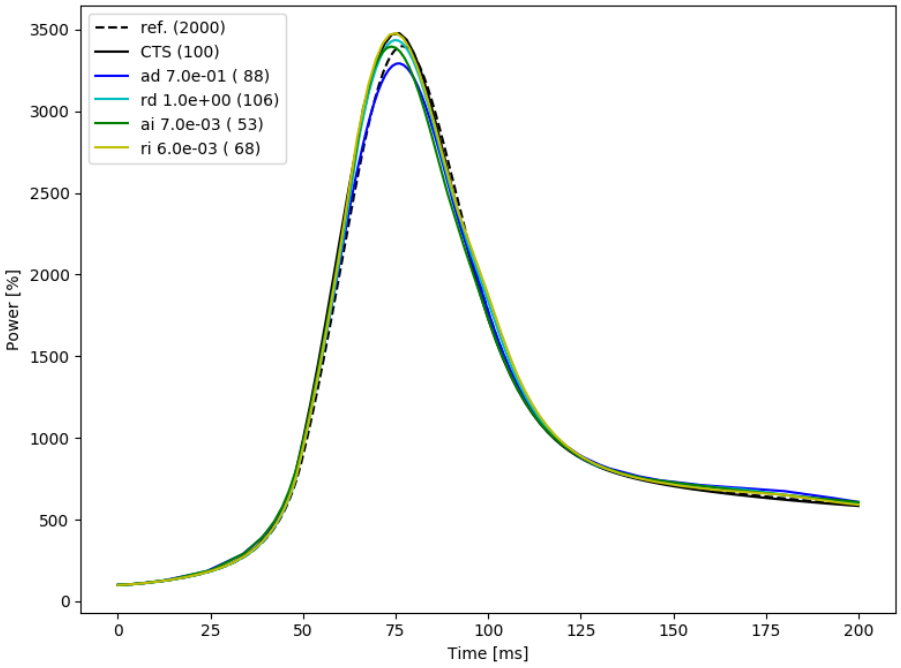

Figure 7.2: SPERT Power Curves with $k-\alpha$ Alpha ATS

Table 7.1: Error Comparison for ATS with $k-\alpha$ Alpha Method For Each Criteria

Crit. Type	Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\bar{RT}_{ts}$
ref.	-	2000	3397	76.80	-	-	-	1.228
CTS	-	100	3477	76.00	10.89	3.65	0.49	2.334
Absolute Direct	7.0e-01	88	3292	76.14	8.89	3.77	0.74	6.199
Relative Direct	1.0e+00	106	3435	74.75	7.38	2.54	0.42	5.982
Absolute Integrated	7.0e-03	53	3395	73.85	9.51	5.35	0.79	6.780
Relative Integrated	6.0e-03	68	3478	75.22	9.73	3.31	0.45	7.384

We again explore the option of calculating the α-eigenvalue by means of the PKEs instead of the costly $k-\alpha$ iteration and present the results in Figures 7.5 through 7.7 and Table 7.2. The accuracy of the relative criteria is degraded, while the absolute criteria show little difference. However, examination of the time step size in 7.6 again reveals the presence of oscillations, defeating the purpose of employing the α-eigenvalue in the first place.

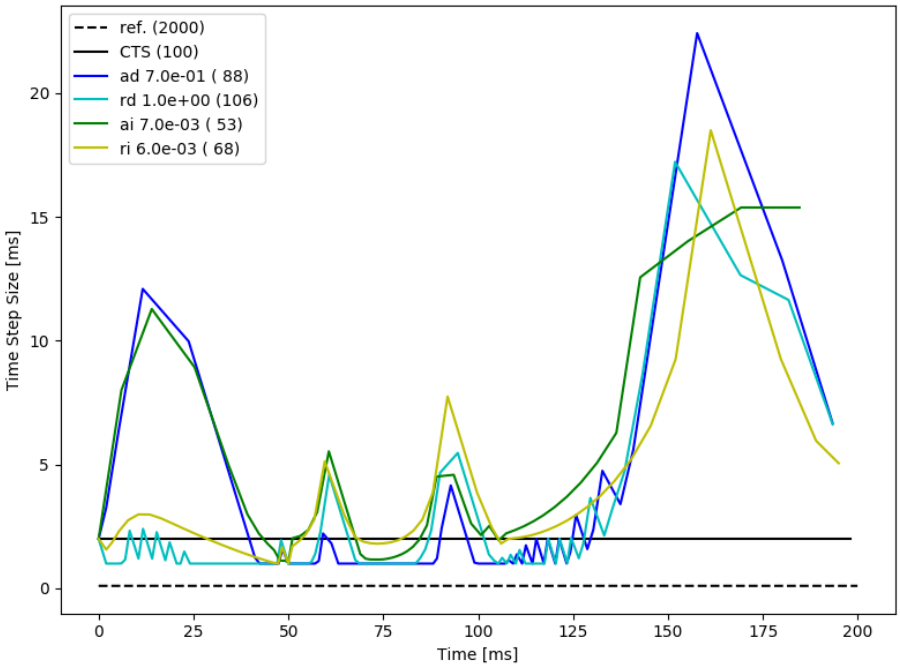

Figure 7.3: Time Step Sizes for SPERT with $k - \alpha$ Alpha ATS

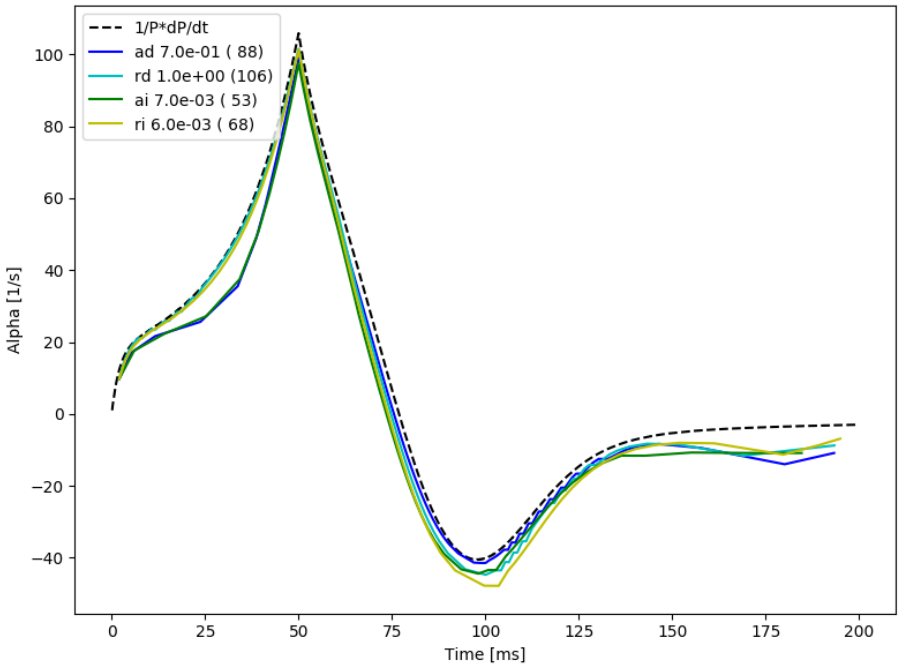

Figure 7.4: Time Constants for SPERT with $k - \alpha$ Alpha ATS

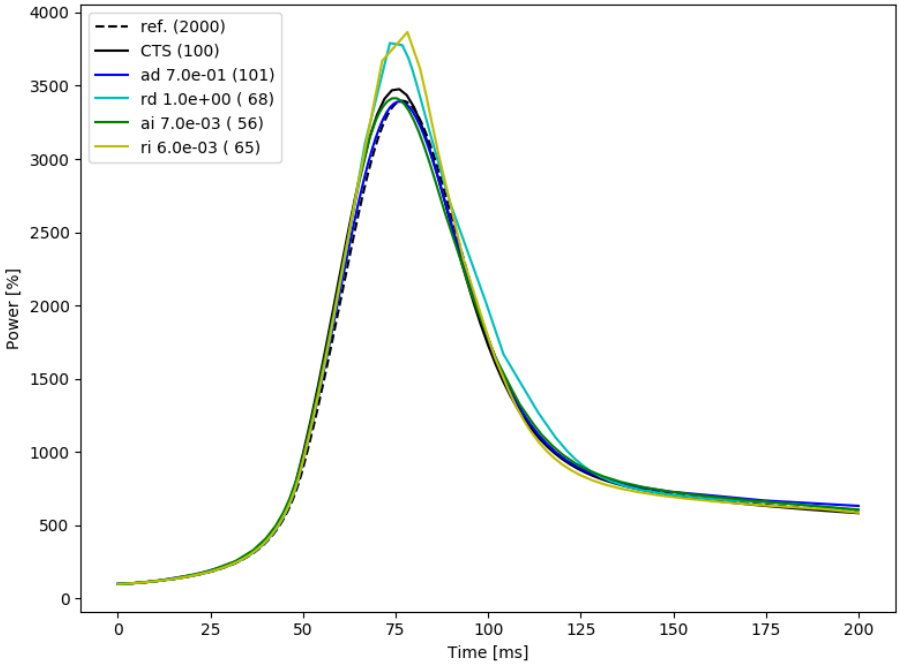

Figure 7.5: SPERT Power Curves with PKE Alpha ATS

Table 7.2: Error Comparison for ATS with PKE Alpha Method For Each Criteria

Crit. Type	Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\bar{RT}_{ts}$
ref.	-	2000	3397	76.80	-	-	-	1.228
CTS	-	100	3477	76.00	10.89	3.65	0.49	2.334
Absolute Direct	7.0e-01	101	3392	76.56	7.24	2.42	0.53	2.093
Relative Direct	1.0e+00	68	3791	73.45	13.89	4.93	0.83	2.661
Absolute Integrated	7.0e-03	56	3415	75.20	9.51	5.03	0.74	2.343
Relative Integrated	6.0e-03	65	3866	78.20	14.18	4.51	0.65	2.671

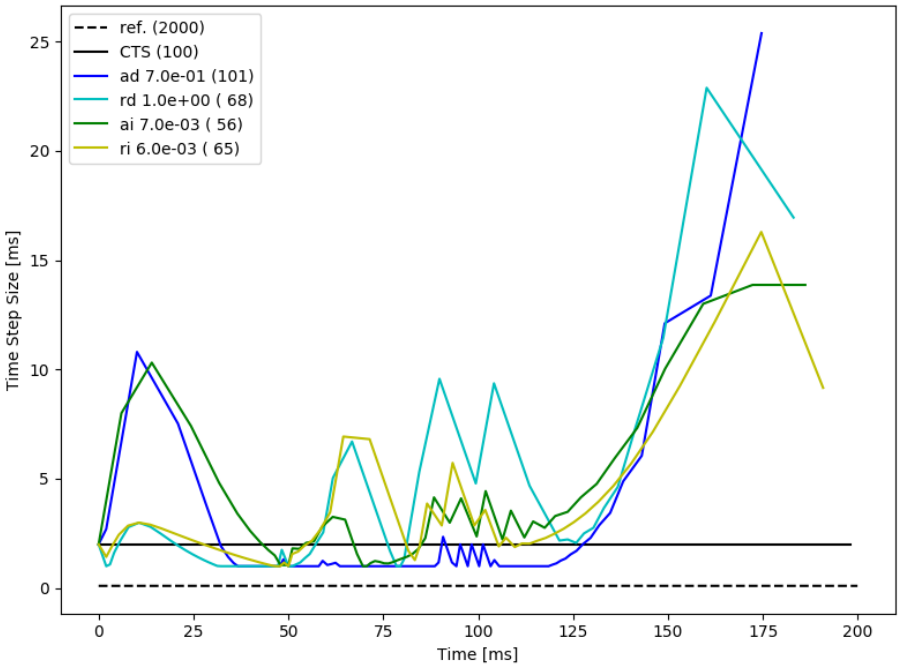

Figure 7.6: Time Step Sizes for SPERT with PKE Alpha ATS

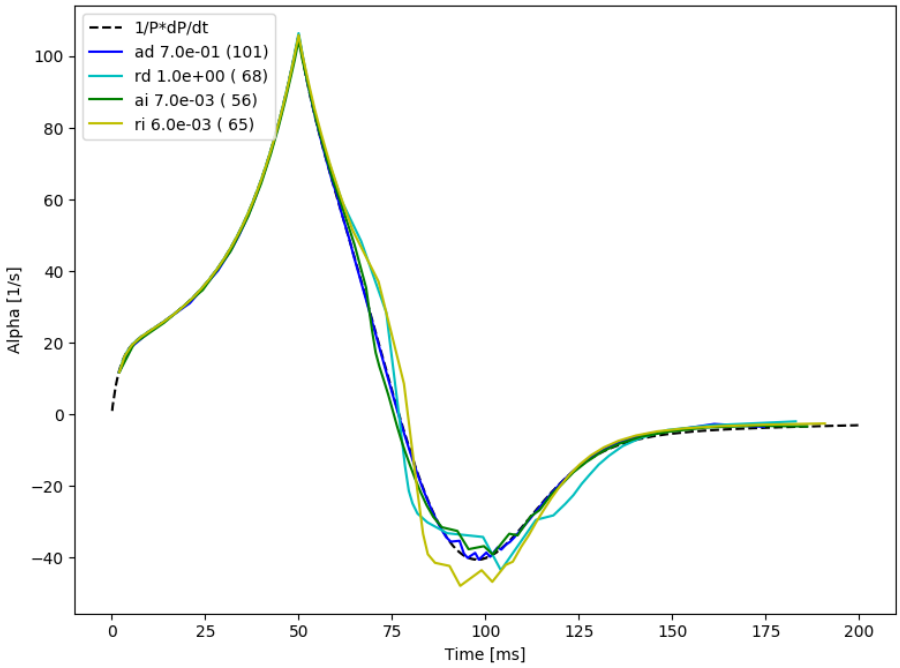

Figure 7.7: Time Constants for SPERT with PKE Alpha ATS

7.1.2 SPERT Summary

The SPERT results have confirmed the trends we have observed for ATS with the α-eigenvalue for a single pin case with feedback for a more complex geometry. The main points are summarized here:

- The α-eigenvalue calculated through $k-\alpha$ iteration may be used to accurately approximate the second derivative of the angular flux.
- The $k-\alpha$ calculation adds a significant computational overhead, limiting the efficacy of the ATS scheme to reduce overall run time.
- The point kinetics estimation of α is not as robust as the $k-\alpha$ result, and is prone to oscillations with feedback.
- The integrated criteria are preferable to the direct criteria since the latter are overly sensitive to the second derivative.
- The relative criteria are more robust when combined with the α-eigenvalue than the absolute criteria.

7.2 C5G7-TD Benchmark

The C5G7-TD benchmark problems consist of a hypothetical miniature light water reactor [7]. The benchmarks have been well studied and provide a rigorous test for state-of-the-art neutron transport codes without spatial homogenization. The reactor consists of 16 fuel assemblies that are modeled by four assemblies in quarter-core symmetry with reflective boundary conditions on two adjacent sides and vacuum boundary conditions on the other two sides. This is illustrated in Figure 7.8. Half of the assemblies are fueled with UO_2, while the other half are fueled by mixed oxide (MOX) fuel at three different enrichment levels– 4.3%, 7.0%, and 8.7%. Each assembly consists of a 17x17 grid of fuel pins with a central fission chamber and 24 guide tubes for the control rods. Figure 7.9 provides a detailed overview of the four assemblies. Each pin cell consists of two regions, as shown in Figure 7.10. The inner region is the pin with a radius of 0.54 cm and consists of the homogenized fuel, gap, and clad. The outer moderator region has a pitch of 1.26 cm containing water. All macroscopic cross sections are specified in a 7-group format as part of the benchmark.

For this work, we have selected the TDW1b C5G7-TD benchmark. This corresponds to a control rod withdrawal, represented in 2D by a ramp change in the material composition of the control rod regions. All control rods begin inserted 1/3 of the way into the core. The bank 1 control rods are withdrawn over 2.0 s. However, a reactor trip is initiated at 1.0 s, and the control rods are reinserted over 1.5 seconds, such that no more material movement occurs after this point. The transient is then allowed to run to 10 s [7]. The position of the bank 1 control rods are shown in red in Figure 7.11.

MPACT models were already developed for this work as part of the MPACT validation suite [49], and were run with fine time steps (0.1 ms) to generate a reference solution and 2 ms constant time steps to generate a solution for comparison. The reference solution was only generated out to 3.0 s to conserve computational resources. MPACT was run with the benchmark-provided 7-group library and 64 azimuthal angles with Chebyshev azimuthal quadrature. CMFD acceleration was used, but not the multilevel method, except for the reference case, which did utilize the multilevel method. All runs were performed in parallel on 16 processors. An initial time step size of 2 ms was utilized with a minimum of 1 ms, a maximum of 50 ms, and a growth limit of 100%.

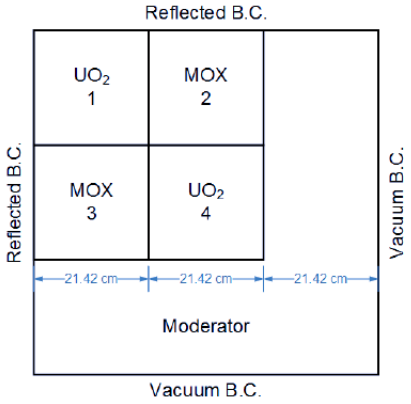

Figure 7.8: C5G7-TD 2D Overall Configuration

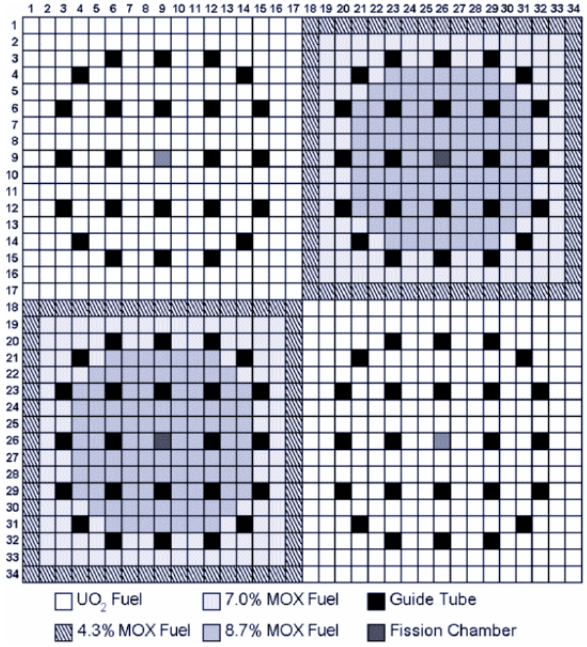

Figure 7.9: C5G7-TD Assembly Detail

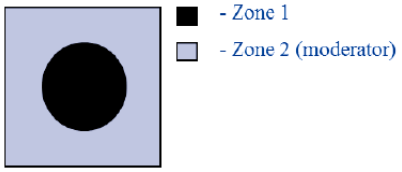

Figure 7.10: C5G7-TD Pin Cell Layout

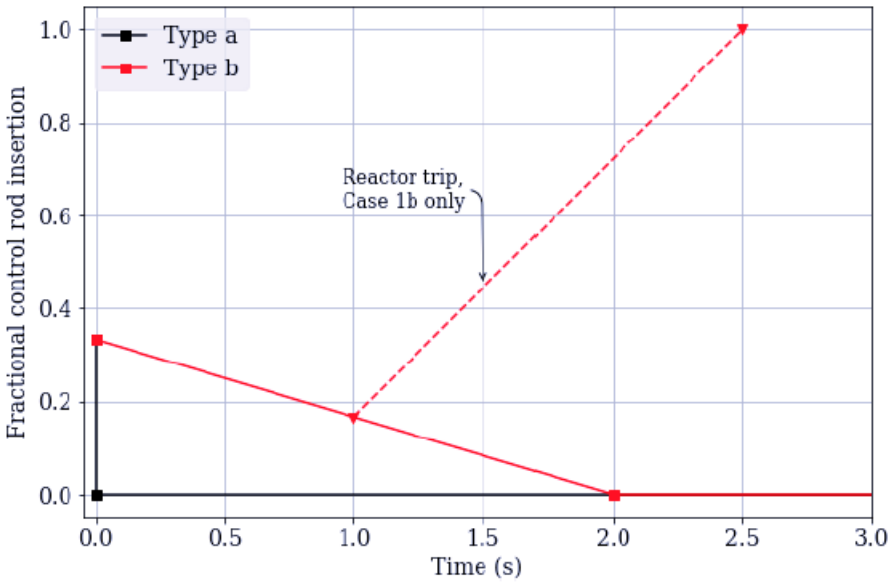

Figure 7.11: C5G7 TDW1 Specification for Control Rod Movement

7.2.1 C5G7-TD Simulations

As for the single pin studies, MPACT was run with each criteria type at a tolerance that produces similar results as the CTS case with 2 ms time steps. However, given the lengthy run times, this was done much more coarsely, varying the tolerance only by powers of 10. The general evolution of the transient power pulse is shown in Figure 7.12. Since the power pulse is so severe without feedback, the peak is blown up in Figure 7.13 for comparison. We forego our usual error tables because the length and severity of the transient greatly distort the results. The peak power is the most useful quantification of error in this instance anyway, and that is available in Figure 7.13. All of the ATS cases shown in that figure required 4.5±0.5 hrs to run, while the CTS case required over 14.5 hours. An examination of the time step sizes in Figure 7.14 reveals the obvious– the prolonged period of the transient without material movement after 2.5 ms is ripe for long time steps. In fact, the ATS performance could be even more dramatic without the 50 ms cap.

Finally, we consider the calculation method of the α-eigenvalues as well, and plot these in Figure 7.15. In addition to the full $k-\alpha$ results, the PKE estimation of α is also shown for ATS runs with absolute and relative integrated criteria that are in excellent agreement with the $k-\alpha$ results, as we have previously observed without feedback. No separate plot is shown for the power curves generated by the PKE runs since they are indistinguishable from the $k-\alpha$ results. However, we note that the run time is reduced by a factor of 2 in these cases since the PKE solution is effectively free compared to the computational cost of the transient step.

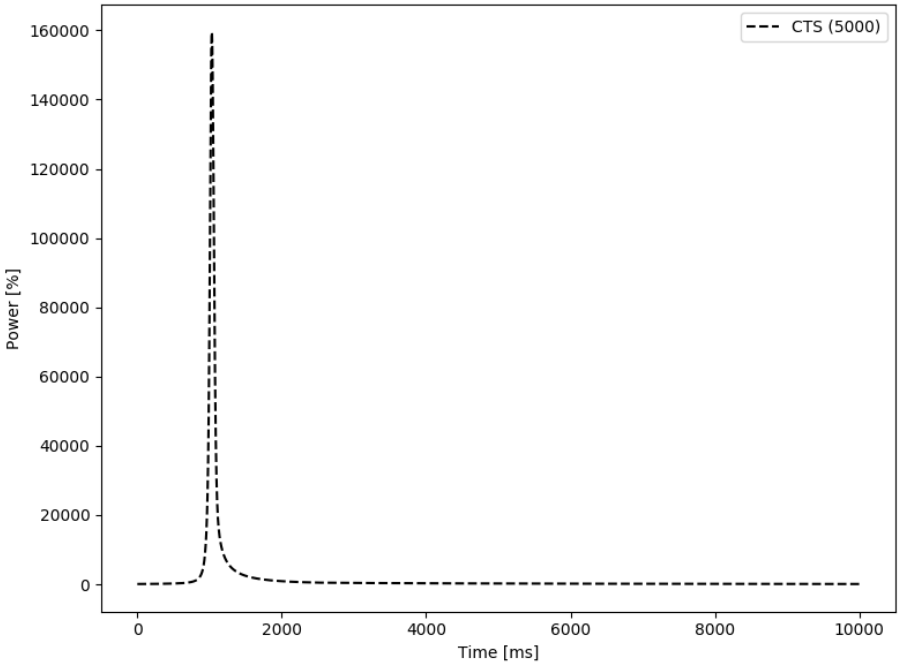

Figure 7.12: C5G7-TD Power Curves with $k-\alpha$ Alpha ATS

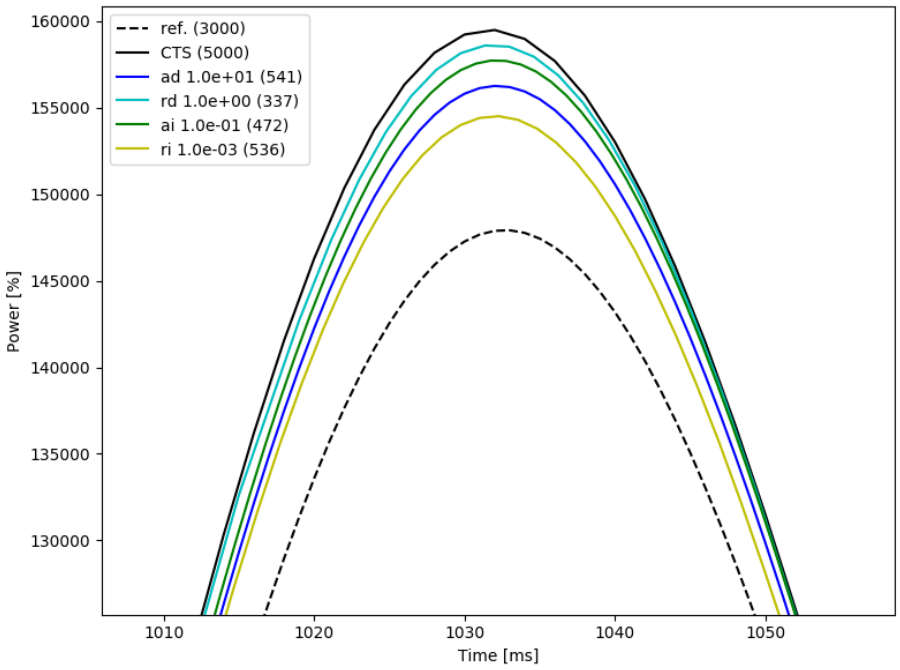

Figure 7.13: Closeup of C5G7-TD Power Curves Peaks with $k - \alpha$ Alpha ATS

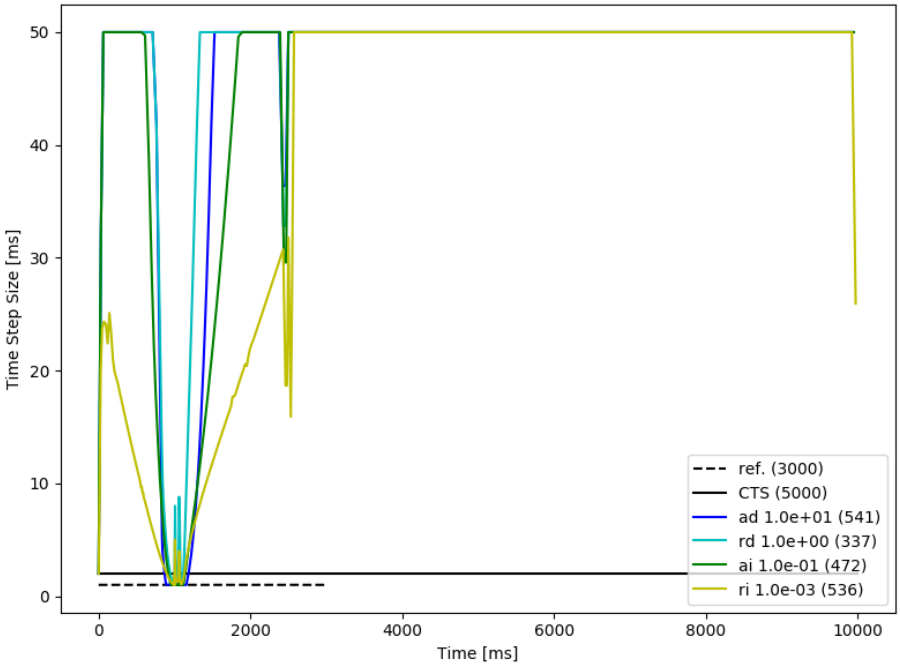

Figure 7.14: Time Step Sizes for C5G7-TD with $k - \alpha$ Alpha ATS

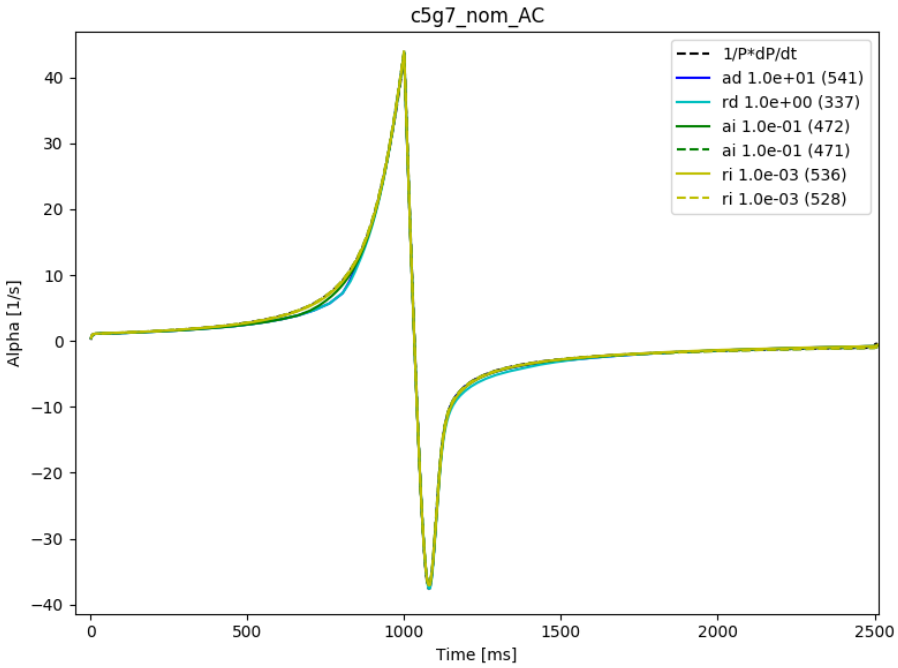

Figure 7.15: Time Constants for C5G7-TD with $k - \alpha$ Alpha ATS

7.2.2 C5G7-TD Summary

The C5G7-TD results have confirmed the trends we have observed for ATS with the α-eigenvalue for a single pin case with feedback for a more complex geometry and demonstrated the power of ATS. The main points are summarized here:

- The ATS scheme achieves drastic speedup for simulations that contain a significant period without material perturbations.
- For simulations with a significant period of no material perturbation, the computational overhead of the $k - \alpha$ iteration is acceptable since the number of solution points can be reduced drastically.
- Without feedback, the point kinetics estimation of α provides an alternative to $k - \alpha$ iteration with essentially no computational overhead.

Chapter 8

Summary and Conclusions

An adaptive time stepping algorithm has been developed for the neutronic solution of MPACT that can reduce the number of time steps required to model a transient without adversely affecting the accuracy of the calculation. This is accomplished by limiting the leading order error term of the time discretization scheme, which has been shown to be proportional to the second derivative of the angular flux in time. The second derivative may be estimated by a finite difference scheme or through the α-eigenvalue.

The finite difference method ostensibly requires storage of the angular flux, but we have shown that the scalar flux may be substituted without loss of accuracy. The finite difference method is also prone to oscillations, which can be at least partially resolved by employing a nested difference finite difference scheme instead of the standard interpolated difference.

As an alternative to the finite difference method, an α-eigenvalue method is explored. The α-eigenvalue can be calculated through the $k-\alpha$ iteration or through the point kinetics equation. The former is more accurate and robust, while the latter is significantly more efficient. The α-eigenvalue methodology can produce similar results to the finite difference methodology without the proclivity for oscillations.

Both methods are shown to perform well without thermal-hydraulic feedback. The introduction of feedback, however, introduces inflection points in the evolution of the angular flux that violate one of the assumptions from which the ATS scheme was derived– that the higher order error terms are negligible. The ATS scheme can still be effective, particularly if there is a significant period without material perturbations, but requires a tighter tolerance than would otherwise be necessary to produce results of comparable accuracy.

8.1 Future Work

Much work remains on this subject, including, but not limited to, the following:

1. The α-eigenvalue (or finite differencing) methodology must be modified to account for the inflection points that occur during superprompt transients with feedback. The most logical extension of this work would be to place a limit on the third derivative in time as well, but this raises substantial

concerns. The second derivative has already shown itself to be prone to oscillations when computed numerically. Furthermore, the second derivative is discontinuous at the commencement or end of material perturbations. It may be possible to address this weakness around inflection points by directly accounting for the evolution of thermal-hydraulic feedback in the ATS scheme. This is no simple task as the evolving thermal-hydraulic conditions must be examined for their effect on neutron transport. Thus, the time constants of the temperature/density changes that are easily calculable provide no value, and derivatives with respect to the thermal-hydraulic parameters, which are notoriously difficult to characterize, are required instead.

2. The solution from the $k-\alpha$ iteration should be explored for use in the transient solution itself. Since the transport solution is obtained in the $k-\alpha$ iteration, it stands to reason that this could be used to replace the transport sweeps of the transient calculation. This would represent a fundamental shift in transient solution methodology for MPACT, but the idea of combining the time step size determination with the transient step itself is tantalizing. The eigenvector solution of the $k-\alpha$ iteration needs to be thoroughly examined to ensure that it suitably captures the evolution of the flux. The advancement of the material perturbations through the transient seems to provide the most significant question and may require iteration.

3. A less aggressive measure than the previous item would be to explore using the result of the $k-\alpha$ iteration to update the fission source for the ensuing transient step.

4. The methodology presented in this thesis should be extended to 3D simulations. This is not a formal hurdle, simply a matter of implementation, as has been demonstrated by utilizing CMFD acceleration with the α-eigenvalue.

5. This work has essentially neglected the ingenious TML method of MPACT transient calculations that allows coarser time steps than would otherwise yield an accurate solution. While the general framework of the ATS method is equally applicable to TML, modifications may be needed to the criteria to properly account for any changes to the leading order error that TML introduces. The author investigated this, but was unable to determine precisely how the TML methodology affected the leading order error term. Additionally, adaptive time stepping at the CMFD and PKE levels of the transient could also be implemented.

6. The JFNK work contained in this thesis was abandoned before being fully explored. Despite many hurdles, the idea that the solution could be generated via JFNK, while a meaningful eigenvalue could be extracted from the matrix formed during the Krylov steps, remains enticing.

Appendices

Appendix A

Tolerance Studies for each Criteria

A.1 Finite Difference ATS without Feedback

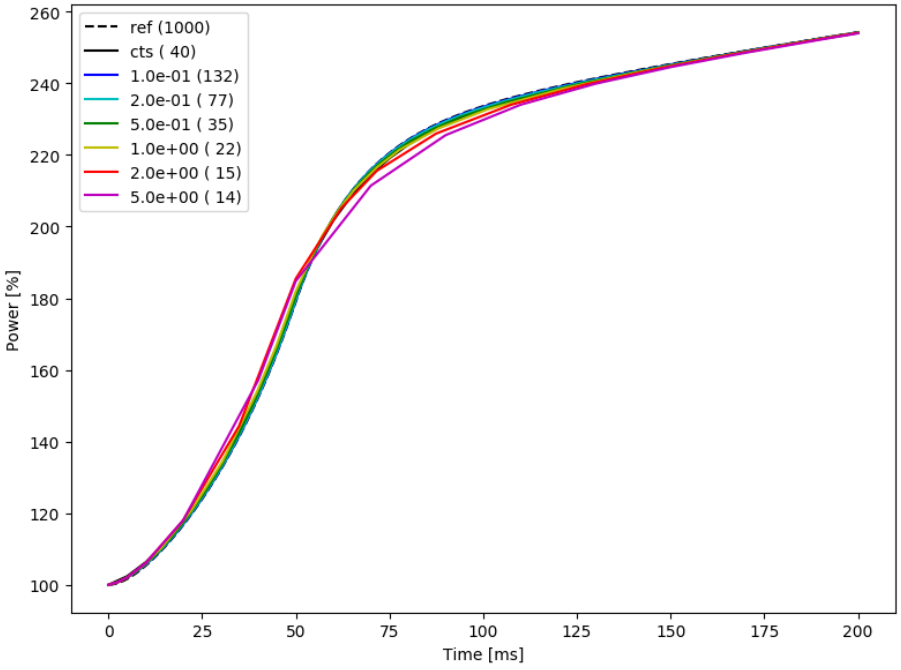

Figure A.1: Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.1: Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances

Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	-	-	-	0.024
CTS	40	1.22	0.52	0.08	0.025
1.0e-01	132	0.39	0.13	0.02	0.028
2.0e-01	77	0.39	0.21	0.03	0.026
5.0e-01	35	0.79	0.44	0.07	0.028
1.0e+00	22	1.54	0.73	0.11	0.029
2.0e+00	15	3.54	1.18	0.20	0.030
5.0e+00	14	3.16	1.45	0.28	0.030

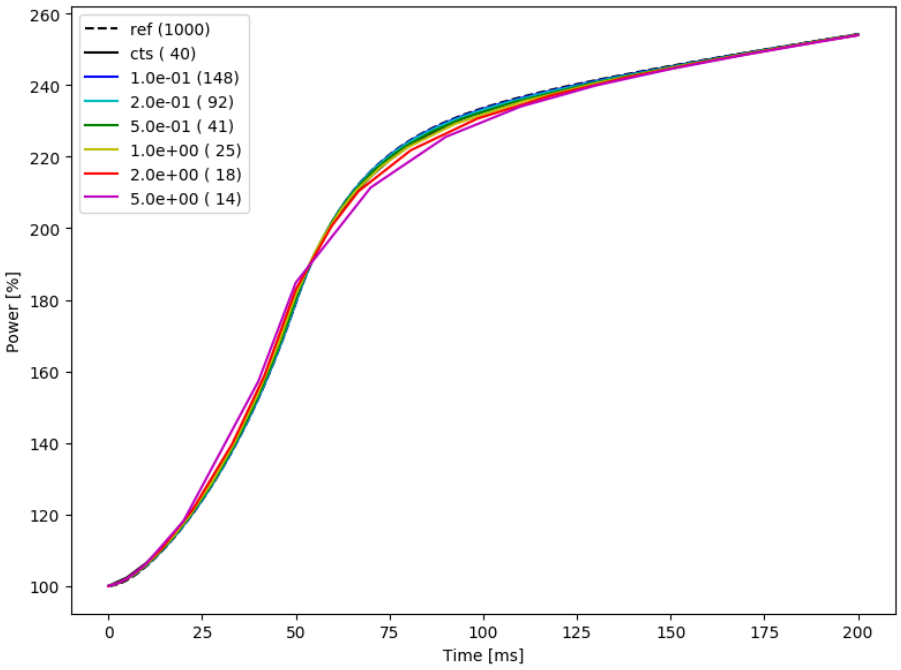

Figure A.2: Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.2: Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances

Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	-	-	-	0.024
CTS	40	1.22	0.52	0.08	0.025
1.0e-01	148	0.39	0.12	0.02	0.026
2.0e-01	92	0.39	0.18	0.03	0.026
5.0e-01	41	0.60	0.35	0.06	0.026
1.0e+00	25	1.27	0.59	0.10	0.029
2.0e+00	18	2.09	0.91	0.16	0.030
5.0e+00	14	3.16	1.45	0.28	0.031

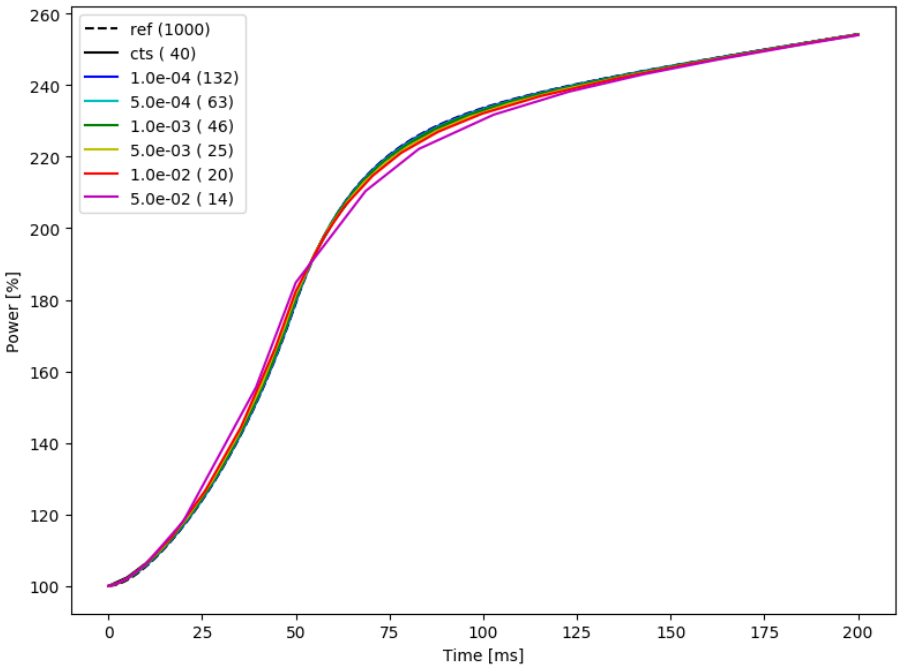

Figure A.3: Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.3: Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances

Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	-	-	-	0.024
CTS	40	1.22	0.52	0.08	0.025
1.0e-04	132	0.39	0.13	0.02	0.027
5.0e-04	63	0.39	0.25	0.03	0.027
1.0e-03	46	0.58	0.33	0.05	0.026
5.0e-03	25	1.36	0.65	0.10	0.028
1.0e-02	20	1.85	0.83	0.13	0.028
5.0e-02	14	3.17	1.41	0.25	0.030

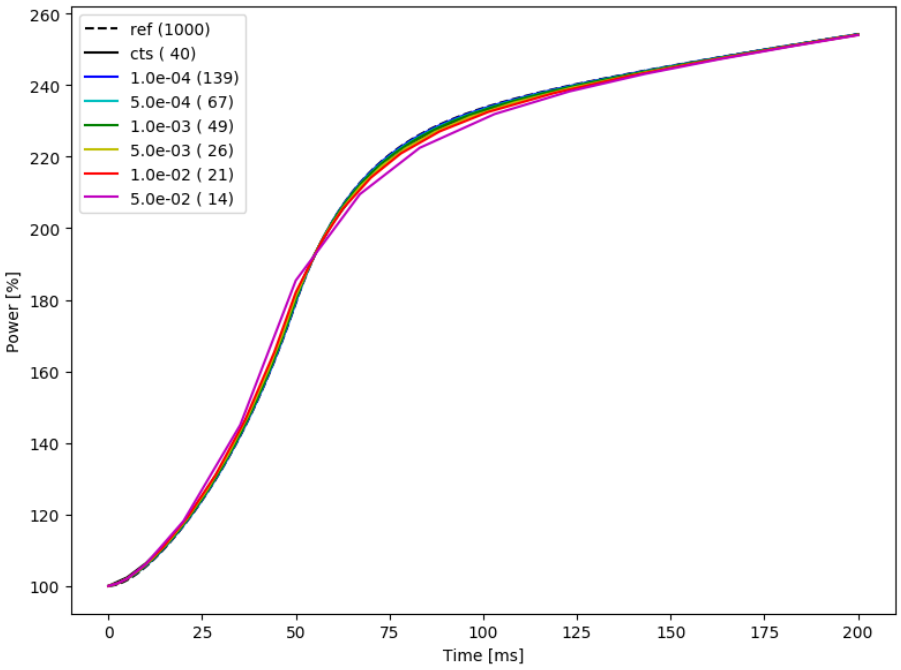

Figure A.4: Single Pin Subprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.4: Single Pin Subprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances

Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	-	-	-	0.024
CTS	40	1.22	0.52	0.08	0.025
1.0e-04	139	0.39	0.13	0.02	0.026
5.0e-04	67	0.39	0.23	0.03	0.026
1.0e-03	49	0.50	0.30	0.04	0.026
5.0e-03	26	1.07	0.58	0.09	0.029
1.0e-02	21	1.64	0.76	0.12	0.029
5.0e-02	14	3.49	1.31	0.24	0.031

A.2 Finite Difference ATS with Feedback

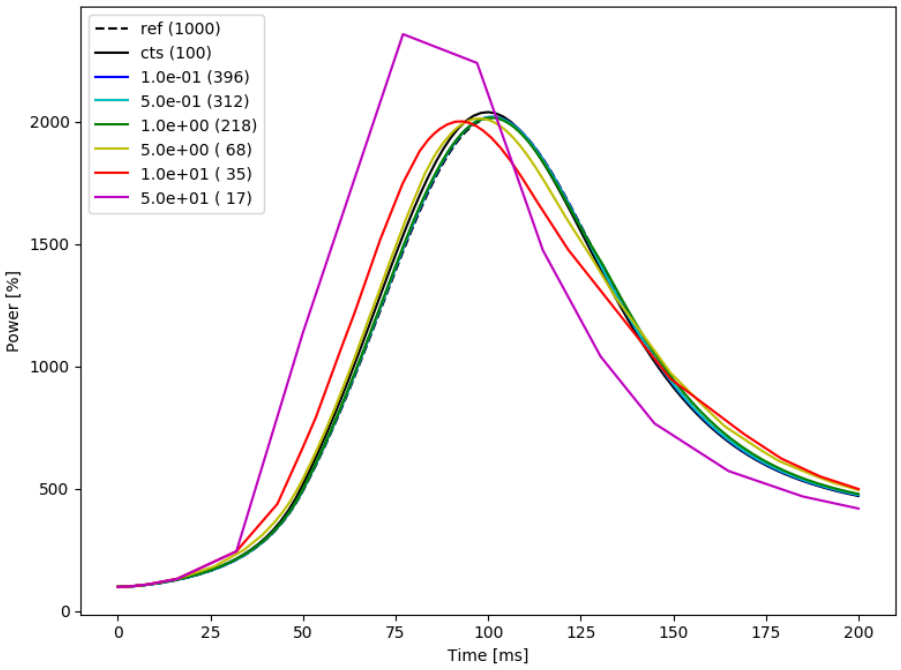

Figure A.5: Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.5: Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances

Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	2016	101.40	-	-	-	0.061
CTS	100	2039	100.00	5.41	2.55	0.40	0.061
1.0e-01	396	2020	100.94	0.86	0.42	0.07	0.062
5.0e-01	312	2019	100.88	1.01	0.55	0.09	0.062
1.0e+00	218	2016	100.56	2.16	1.21	0.24	0.063
5.0e+00	68	2013	97.86	11.19	5.99	1.13	0.064
1.0e+01	35	2001	93.04	36.09	14.44	2.51	0.066
5.0e+01	17	2357	77.00	131.05	43.05	6.99	0.071

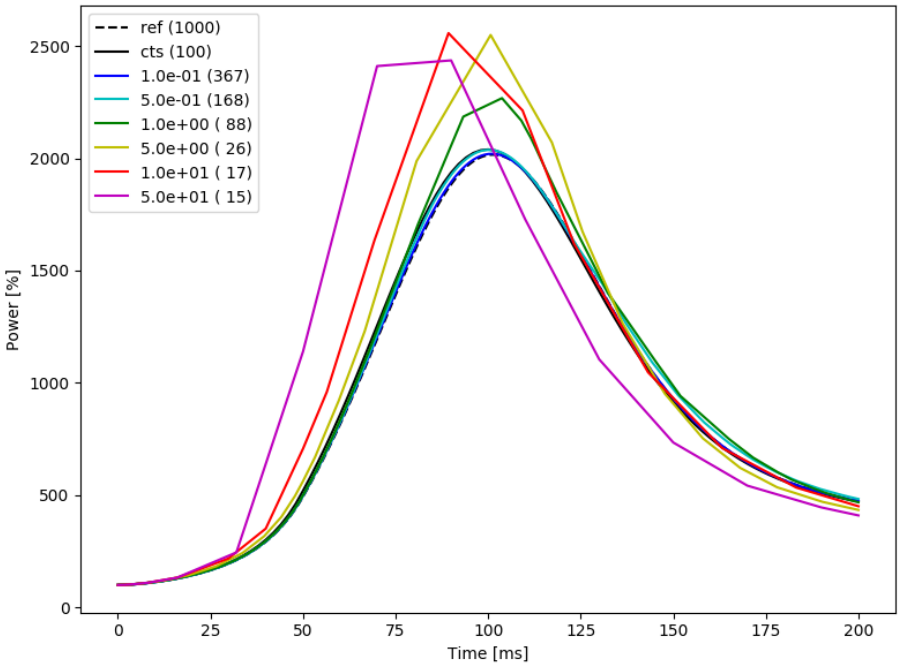

Figure A.6: Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.6: Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances

Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	2016	101.40	-	-	-	0.061
CTS	100	2039	100.00	5.41	2.55	0.40	0.061
1.0e-01	367	2020	101.04	0.90	0.43	0.06	0.063
5.0e-01	168	2037	99.90	5.15	1.69	0.40	0.063
1.0e+00	88	2267	103.73	12.82	3.40	0.86	0.063
5.0e+00	26	2549	100.67	26.46	10.93	2.05	0.069
1.0e+01	17	2557	89.29	43.33	19.88	2.86	0.073
5.0e+01	15	2436	90.00	131.05	43.53	7.40	0.073

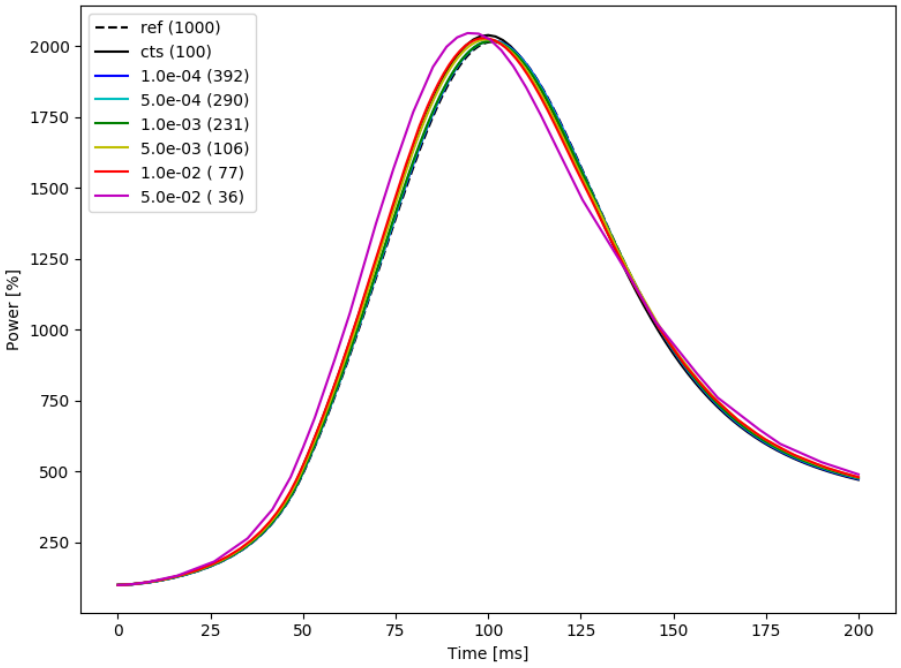

Figure A.7: Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.7: Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances

Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	2016	101.40	-	-	-	0.061
CTS	100	2039	100.00	5.41	2.55	0.40	0.061
1.0e-04	392	2020	101.17	0.86	0.42	0.07	0.062
5.0e-04	290	2019	100.77	1.13	0.60	0.10	0.063
1.0e-03	231	2017	100.68	1.55	0.88	0.15	0.062
5.0e-03	106	2024	99.53	3.99	2.37	0.40	0.063
1.0e-02	77	2029	99.12	5.90	3.42	0.56	0.063
5.0e-02	36	2046	94.45	18.52	9.04	1.45	0.066

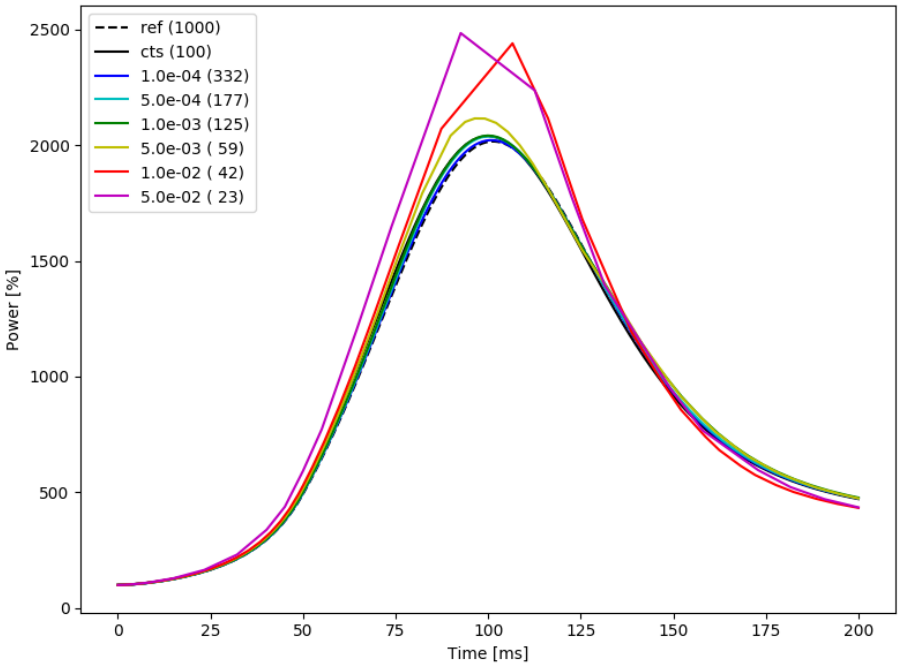

Figure A.8: Single Pin Superprompt Power Curves for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.8: Single Pin Superprompt Error Comparison for ATS with Interpolated Finite Difference Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances

Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	2016	101.40	-	-	-	0.061
CTS	100	2039	100.00	5.41	2.55	0.40	0.061
1.0e-04	332	2021	100.93	1.31	0.51	0.07	0.063
5.0e-04	177	2038	100.00	2.87	1.00	0.17	0.062
1.0e-03	125	2041	99.86	3.52	1.73	0.33	0.063
5.0e-03	59	2115	96.40	8.54	3.60	0.61	0.066
1.0e-02	42	2439	106.58	22.71	7.59	1.51	0.066
5.0e-02	23	2484	92.59	28.25	12.48	2.16	0.073

A.3 Alpha Eigenvalue ATS without Feedback

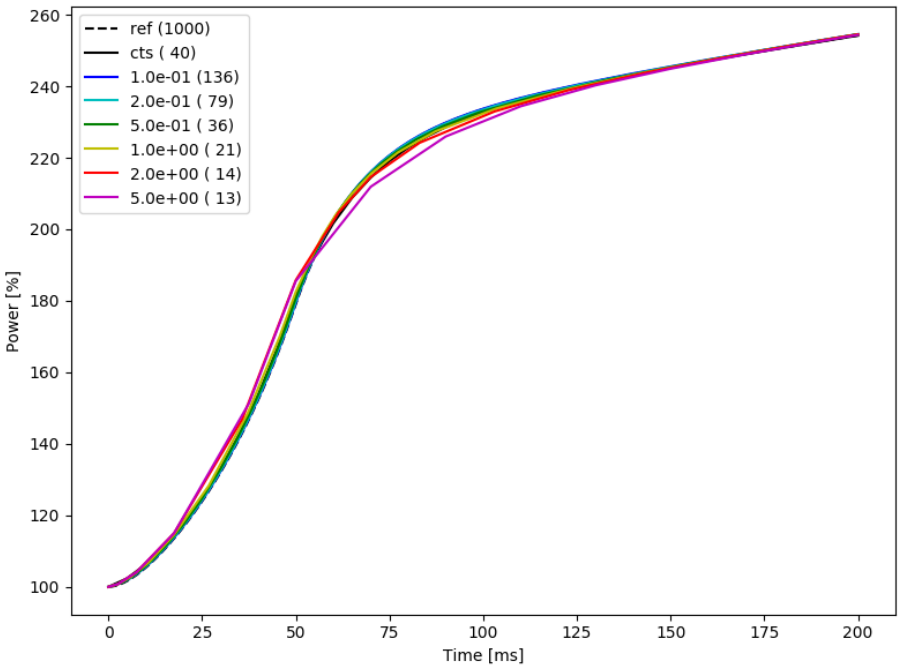

Figure A.9: Single Pin Subprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.9: Single Pin Subprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances

Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	-	-	-	0.047
CTS	40	1.22	0.52	0.08	0.058
1.0e-01	136	0.29	0.16	0.02	0.124
2.0e-01	79	0.35	0.24	0.03	0.108
5.0e-01	36	0.88	0.47	0.06	0.113
1.0e+00	21	2.00	0.84	0.11	0.116
2.0e+00	14	3.70	1.29	0.20	0.126
5.0e+00	13	3.56	1.50	0.27	0.141

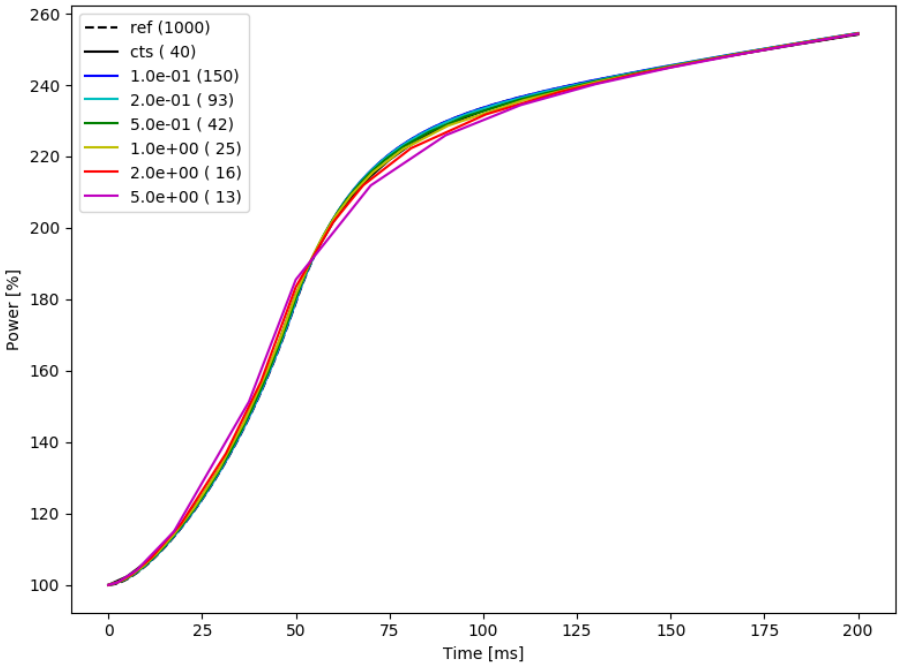

Figure A.10: Single Pin Subprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.10: Single Pin Subprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances

Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	-	-	-	0.047
CTS	40	1.22	0.52	0.08	0.058
1.0e-01	150	0.29	0.15	0.02	0.127
2.0e-01	93	0.29	0.20	0.02	0.117
5.0e-01	42	0.67	0.36	0.05	0.103
1.0e+00	25	1.27	0.62	0.09	0.126
2.0e+00	16	2.41	1.00	0.16	0.131
5.0e+00	13	3.56	1.50	0.27	0.172

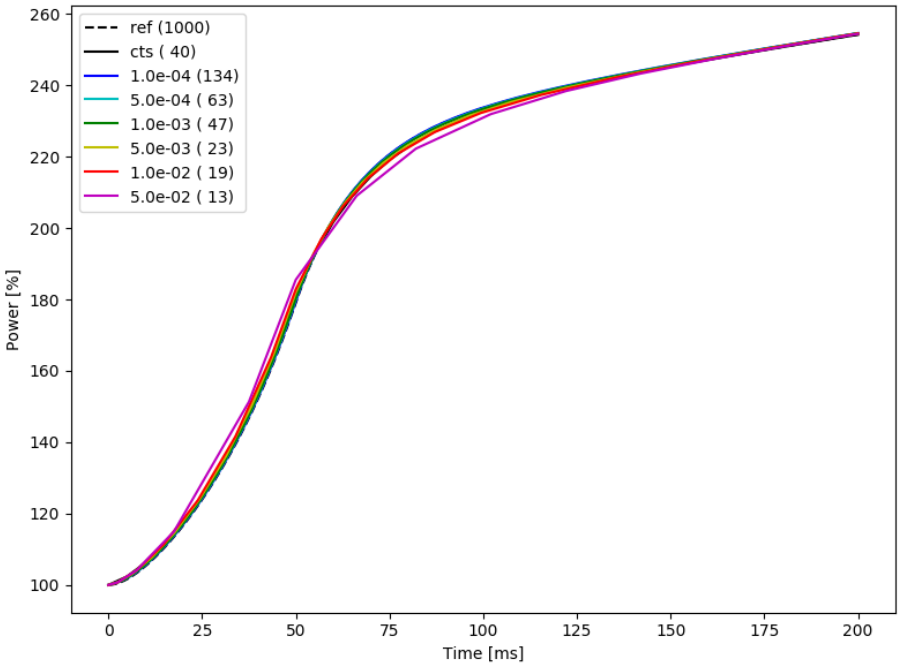

Figure A.11: Single Pin Subprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.11: Single Pin Subprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances

Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	-	-	-	0.047
CTS	40	1.22	0.52	0.08	0.058
1.0e-04	134	0.29	0.17	0.03	0.151
5.0e-04	63	0.48	0.27	0.03	0.116
1.0e-03	47	0.65	0.35	0.04	0.130
5.0e-03	23	1.59	0.69	0.09	0.142
1.0e-02	19	2.07	0.88	0.12	0.118
5.0e-02	13	3.56	1.45	0.24	0.165

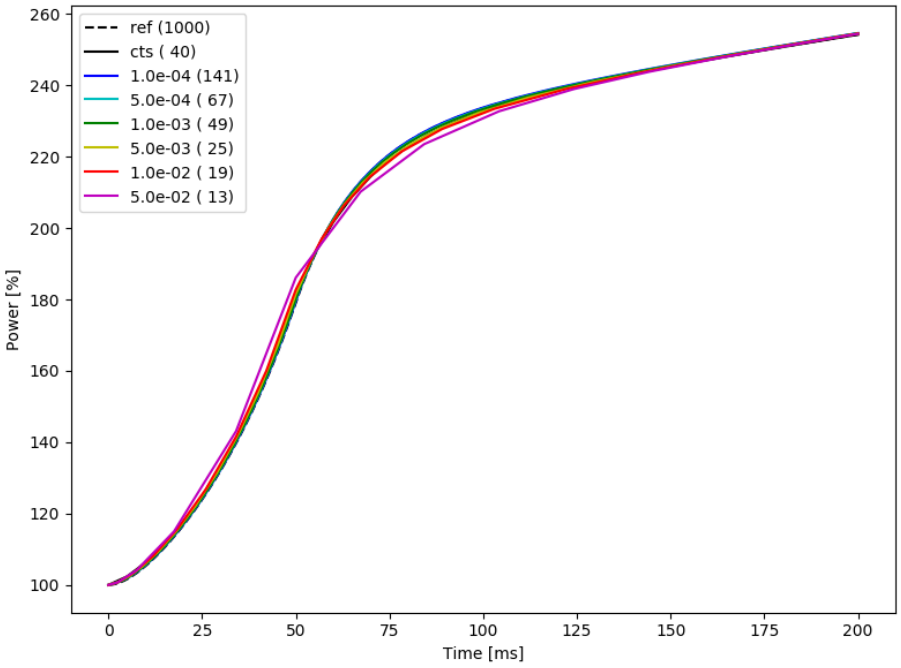

Figure A.12: Single Pin Subprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.12: Single Pin Subprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances

Tol.	No. of Time Steps	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	-	-	-	0.047
CTS	40	1.22	0.52	0.08	0.058
1.0e-04	141	0.29	0.16	0.02	0.152
5.0e-04	67	0.42	0.25	0.03	0.133
1.0e-03	49	0.57	0.32	0.04	0.127
5.0e-03	25	1.33	0.61	0.08	0.135
1.0e-02	19	2.00	0.81	0.11	0.163
5.0e-02	13	3.89	1.42	0.24	0.148

A.4 Alpha Eigenvalue ATS with Feedback

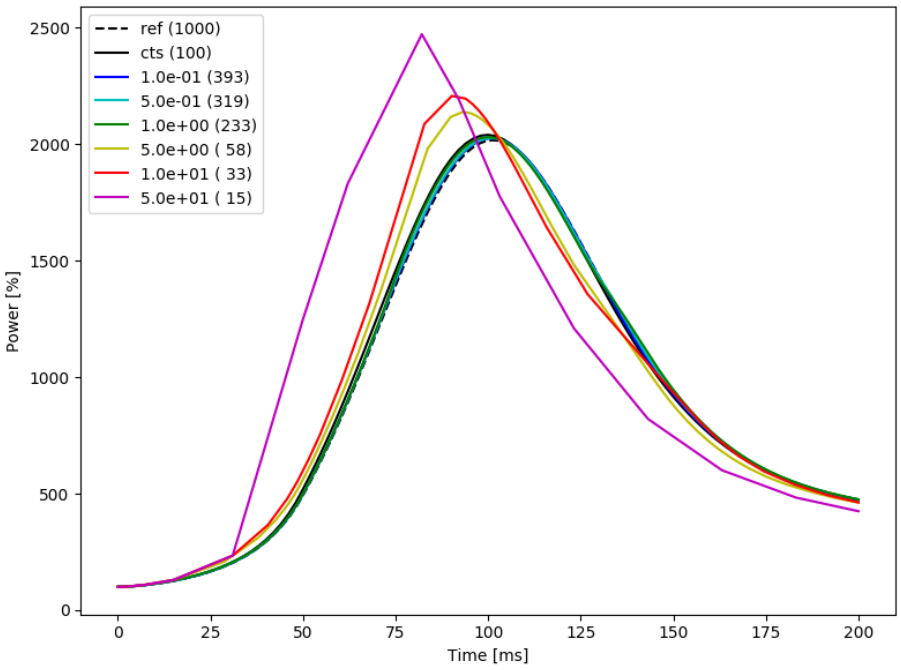

Figure A.13: Single Pin Superprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.13: Single Pin Superprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Absolute Direct Criteria for a Range of Tolerances

Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	2016	101.40	-	-	-	0.063
CTS	100	2039	100.00	5.41	2.55	0.40	0.064
1.0e-01	393	2024	101.16	1.15	0.55	0.09	0.095
5.0e-01	319	2025	100.56	1.72	0.79	0.13	0.094
1.0e+00	233	2029	99.91	3.36	1.49	0.25	0.096
5.0e+00	58	2136	92.96	16.55	8.64	1.59	0.101
1.0e+01	33	2206	90.21	24.30	12.30	1.95	0.105
5.0e+01	15	2471	82.08	153.27	49.61	7.68	0.110

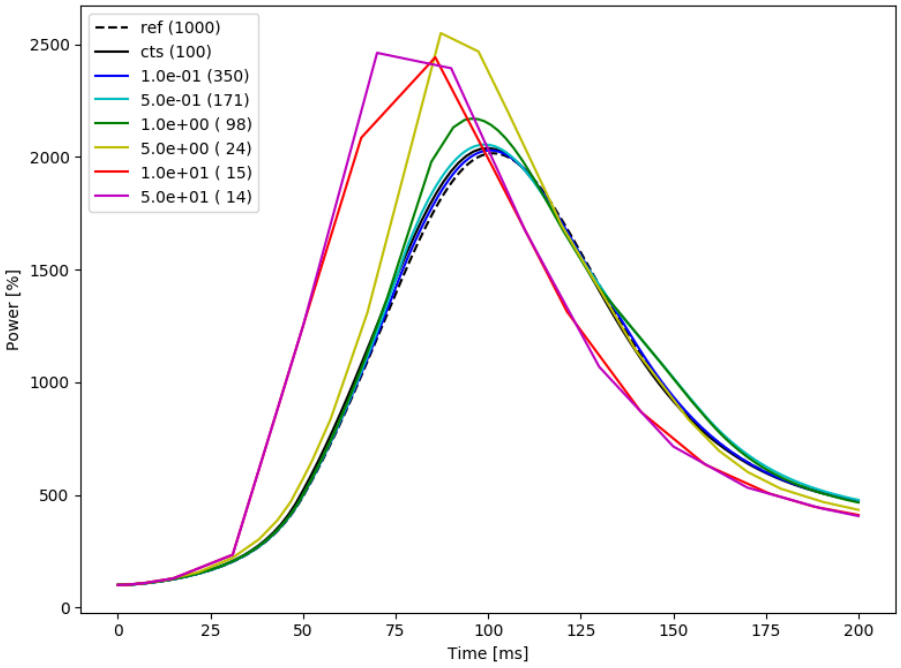

Figure A.14: Single Pin Superprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.14: Single Pin Superprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Relative Direct Criteria for a Range of Tolerances

Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	2016	101.40	-	-	-	0.063
CTS	100	2039	100.00	5.41	2.55	0.40	0.064
1.0e-01	350	2029	100.33	2.32	0.83	0.14	0.096
5.0e-01	171	2055	99.02	9.95	2.84	0.56	0.102
1.0e+00	98	2169	95.31	13.80	3.64	0.75	0.095
5.0e+00	24	2549	87.21	40.58	13.17	2.11	0.117
1.0e+01	15	2442	85.72	153.27	48.54	7.78	0.109
5.0e+01	14	2462	70.00	153.27	50.18	8.24	0.120

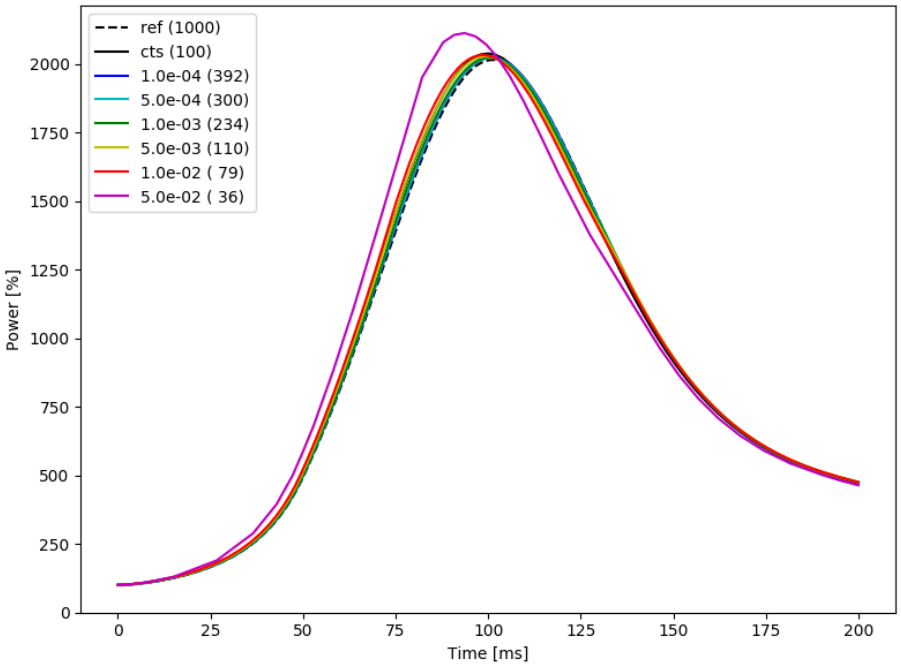

Figure A.15: Single Pin Superprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.15: Single Pin Superprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Absolute Integrated Criteria for a Range of Tolerances

Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	2016	101.40	-	-	-	0.063
CTS	100	2039	100.00	5.41	2.55	0.40	0.064
1.0e-04	392	2024	100.89	1.10	0.54	0.08	0.095
5.0e-04	300	2024	100.77	1.54	0.72	0.10	0.101
1.0e-03	234	2024	100.25	2.51	1.07	0.15	0.096
5.0e-03	110	2029	99.39	4.84	2.39	0.37	0.098
1.0e-02	79	2034	98.44	7.14	3.53	0.53	0.100
5.0e-02	36	2113	93.66	19.07	9.78	1.64	0.105

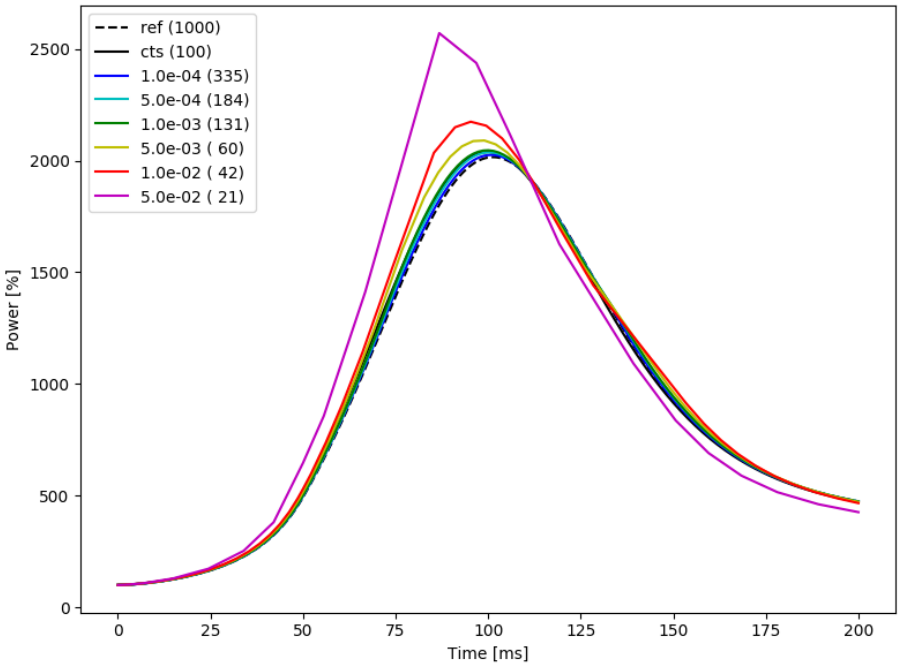

Figure A.16: Single Pin Superprompt Power Curves for ATS with Alpha Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances, Time Steps Shown in Parentheses

Table A.16: Single Pin Superprompt Error Comparison for ATS with Alpha Method Using the Angular Flux and Relative Integrated Criteria for a Range of Tolerances

Tol.	No. of Time Steps	Peak Power (%)	Time to Peak (ms)	$\epsilon^{rel.}_{MAX}$ (%)	$\epsilon^{rel.}_{RMS}$ (%)	$\epsilon^{rel.}_{INT}$ (%)	$\overline{RT}_{ts}$ (s)
ref.	1000	2016	101.40	-	-	-	0.063
CTS	100	2039	100.00	5.41	2.55	0.40	0.064
1.0e-04	335	2026	100.90	1.55	0.63	0.10	0.102
5.0e-04	184	2037	100.28	3.05	1.15	0.20	0.102
1.0e-03	131	2046	100.05	4.49	1.66	0.28	0.104
5.0e-03	60	2090	98.97	9.64	3.81	0.63	0.106
1.0e-02	42	2174	95.28	15.65	5.81	0.97	0.110
5.0e-02	21	2571	86.79	42.69	17.03	2.98	0.118

Bibliography

[1] M.L. Adams, E.W. Larsen, "Fast Iterative Methods for Discrete-Ordinates Particle Transport Calculations," *Prog. Nucl. Energy*, **40**, 1, 3-159 (2002).

[2] J. Askew, "A Characteristics Formulation of the Neutron Transport Equation in Complicated Geometries," Tech. Rep. AEEW-R-1108, United Kingdom Atomic Energy Authority (1972).

[3] Y. Ban, T. Endo, A. Yamamoto, "A Unified Approach for Numerical Calculation of Space-Dependent Kinetic Equation" *J. Nucl. Sci. Technol.*, **49**, 496-515 (2012).

[4] R. Bednarz, "Spectrum of the Boltzmann Operator with an Isotropic Thermalization Kernel" *Proceedings of the Symposium on Pulsed Neutron Research*, **I**, 259-270 (1965).

[5] B.R. Betzler, "Calculating Alpha Eigenvalues and Eigenfunctions with a Markov Transition Rate Matrix Monte Carlo Method," PhD Dissertation, University of Michigan (2014).

[6] J. Boffie, J.M. Pounders, "An Adaptive Time Step Control Scheme for the Transient Diffusion Equation," *Ann. Nucl. Energy*, **116**, 280-289 (2018).

[7] V.F. Boyarniov, P.A. Fomichenko, J. Hou, K. Ivanov, A. Aures, W. Zwermann, K. Velkov, "Deterministic Time-Dependent Neutron Transport Benchmark without Spatial Homogenization (C5G7-TD)," Organization for Economic Cooperation and Development/Nuclear Energy Agency Report, Version 1.9 (2018).

[8] Y. Cao, "Space-Time Kinetics and Time-Eigenfunctions," PhD Dissertation, University of Michigan (2008).

[9] L. Cao, A. Gerlach, Y. Xu, T. Downar, J.C. Lee, "Neutronics Modeling of the SPERT III E-Core Critical Experiments with MPACT and KENO," *Ann. Nucl. Energy*, **80**, 207-218 (2015).

[10] D.G. Cacuci, Y. Ronen, Z. Shayer, J.J. Wagschal, Y. Yeivin, "Eigenvalue-Dependent Neutron Energy Spectra: Definitions, Analyses, and Applications," *Nucl. Sci. Eng.*, **81**, 432-442 (1982).

[11] J.Y. Cho, H.G. Joo, K.S. Kim, S.Q. Zee, "Three-dimensional heterogeneous whole core transport calculation employing planar MOC solutions," *Trans. Am. Nucl. Soc.*, **87**, 234–236 (2002).

[12] N.Z. Cho, G.S. Lee, C.J. Park, "Fusion method of characteristics and nodal method for 3D whole core transport calculation," *Trans. Am. Nucl. Soc.*, **86**, 322–324 (2002).

[13] V. Colombo, P. Ravetto, "Determination of neutron transport equation eigenvalues as lambda matrix latent roots," *Kerntechnik*, **54**, 38-43 (1989).

[14] T. Downar, Y. Xu, A. Zhu, B. Kochunas, D. Jabaay, "Transient Capability in MPACT with Internal Heat Conduction Feedback for Analysis of PWR Reactivity Insertion Accidents," CASL Technical Report CASL-U-2016-1188-000 (2016).

[15] J. J. Duderstadt, L. J. Hamilton, *Nuclear Reactor Analysis*, Wiley (1976).

[16] S. Dulla, E.H. Mund, P. Ravetto, "The Quasi-Static Method Revisited," *Prog. Nucl. Energy*, **50**, 908-917 (2008).

[17] E. Dumonteil, T. Courau, "Dominance Ratio Assessment and Monte Carlo Criticality Simulations: Dealing with High Dominance Ratio Systems," *Nucl. Tech.*, **172**, 120-131 (2009).

[18] J. Durgone, "SPERT III Reactor Facility: E-CORE Revision," AEC Research and Development Report IDO-17036 (1965).

[19] A. Gerlach, B. Kochunas, B. Salko, "Validation of RIA with SPERT," CASL Technical Report CASL-U-2018-1643-000 (2018).

[20] D. F. Gill, "Newton-Krylov Methods for the Soultion of the k-Eigenvalue Problem in Mulitgroup Neutronics Calculations," PhD Dissertation, Pennsylvania Sate University (2009).

[21] S. Goluoglu, H.L. Dodds, "A Time-Dependent, Three-Dimensional Neutron Transport Methodology," *Nucl. Sci. Eng.*, **139**, 248–261 (2001).

[22] M. W. Hackemack, J. M. Pounders, "Implementation of an A Priori Time Step Estimator for the Multigroup Neutron Diffusion Equation in Asynchronously Coupled RELAP5-3D," *Proc. PHYSOR 2014*, Kyoto, Japan, Sep. 28 – Oct. 3 (2014).

[23] K.F. Hansen, B.V. Koen, W. Little, "Stable Numerical Solutions of the Reactor Kinetics Equations," *Nucl. Sci. Eng.*, **22**, 51-59 (1965).

[24] A. Henry, "The Application of Reactor Kinetics to the Analysis of Experiments," *Nucl. Sci. Eng.*, **3**, 52-70 (1958).

[25] A. Henry, N.J. Curlee, "Verification of a Method for Treating Neutron Space-Time Problems," *Nucl. Sci. Eng.*, **4**, 727-744 (1958).

[26] A.J. Hoffman, J.C. Lee, "Low-Order Approximations to the Angular Flux Time Derivative for Transport-based Reactor Kinetics," *Trans. Am. Nucl. Soc.*, **108**, 777-780 (2013).

[27] A.J. Hoffman, J.C. Lee, "A Time-Dependent Neutron Transport Method of Characteristics Formulation with Time Derivative Propagation," *J. Comput. Phys.*, **307**, 696-714 (2016).

[28] K. Jörgens, "An Asymptotic Expansion in the Theory of Neutron Transport," *Comm. Pure Appl. Math.*, **XI**, 219-242 (1958).

[29] D.A. Knoll, D.E. Keyes, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications," *J. Comput. Phys.*, **193**, pp. 357-397 (2004).

[30] D.A. Knoll, H. Park, K. Smith, "Application of the Jacobian-Free Newton-Krylov Method in Computational Reactor Physics," *International Conference on Mathematics, Computational Methods, and Reactor Physics*, Saratoga Springs, NY, May 3-7 (2009).

[31] B.M. Kochunas, "A Hybrid Parallel Agorthm for the 3-D Method of Characteristics Solution of the Boltzmann Transport Equation on High Performance Compute Clusters," PhD Dissertation, University of Michigan (2013).

[32] E.W. Larsen, P.F. Zweifel, "On the spectrum of the linear transport operator," *J. Math. Phys.*, **15**, 1987-1997 (1974).

[33] J.C. Lee, *Nuclear Reactor Physics and Engineering*, Wiley (2020).

[34] J. Lehner, G.M. Wing, "An Unsymmetric Operator Arising in the Theory of Neutron Diffusion" *Commun. Pure Appl. Math*, textbfIX, 487-497 (1956).

[35] R.J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, Society for Industrial and Applied Mathematics (2007).

[36] R.K. McCardell, D.I. Herborn, J.E. Houghtaling, "Reactivity Accident Test Results and Analyses for the SPERT III E-Core—a Small, Oxide-fueled, Pressurized-water Reactor," AEC Research and Development Report IDO-17281 (1969).

[37] J. Mika, "The Initial-Value Problem for a Slab," *J. Math. Phys.*, **7** (5), 833-843 (1966).

[38] MPACT Team "MPACT Theory Manual," *Version 2.2.0* (2016).

[39] M. Nelkin, "Asymptotic Solutions of the Transport Equation for Thermal Neutrons," *Physica*, **29**, 261-273 (1963).

[40] K.O. Ott, D.A. Meneley, "Accuracy of the Quasistatic Treatment of Spatial Reactor Kineticss," *Nucl. Sci. Eng.*, **36**, 402-411 (1969).

[41] K.O. Ott, R.J. Neuhold *Introductory Nuclear Reactor Dynamics*, American Nuclear Society (1985).

[42] H. Park, D.A. Knoll, C. K. Newman, "Nonlinear Acceleration of Transport Criticality Problems," *Nuc. Sci. Eng.*, **172**, 52-65 (2012).

[43] F. Perdu, J.M. Loiseaux, A. Billebaud, R. Brissot, D. Heuer, C. Lebrun, E. Liatard, O. Meplan, E. Merle, H. Nifenecker, J. Vollaire, "Prompt reactivity determination in a subcritical assembly through the response to a Dirac pulse," *Prog. Nucl. Energy*, **42 (1)**, 107-120 (2003).

[44] R.M. Potenza, "Quarterly Technical Report, SPERT Project," AEC Research and Development Report IDO-17206 (1966).

[45] J.M. Pounders, J. Boffie, "Analysis of an Adaptive Time Step Scheme for the Transient Diffusion Equation," *Joint International Conference on Mathematics, Supercomputing in Nuclear Applications, and the Monce Carlo Method*, Nashville, TN, April 19-23 (2015).

[46] Y. Saad, M.H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *J. Sci. Comput.*, **7 (3)**, 856-869 (1986).

[47] Y. Saad, "Numerical Methods for Large Eigenvalue Problems," Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics (1992).

[48] D.C. Sahni, "Some New Results Pertaining to Criticality and Time Eigenvalues of One-Speed Neutron Transport Equation," *Prof. Nucl. Energy*, **30 (3)**, 305-320 (1996).

[49] Q. Shen, Y. Wang, D. Jabaay, B. Kochunas, T. Downar, "Transient Analysis of C5G7-TD Benchmark with MPACT," *Ann. Nucl. Energy*, **125**, 107-120 (2019).

[50] K.P. Singh, R.S. Modak, S.B. Degweker, K. Singh, "Iterative schemes for obtaining dominant alpha-modes of the neutron diffusion equation," *Ann. Nucl. Energy*, **36**, 1086-1092 (2009).

[51] K.S. Smith, "Nodal method storage reduction by nonlinear iteration," *Trans. Am. Nucl. Soc.*, **44**, 265 (1983).

[52] T.M. Sutton, B.N Aviles, "Diffusion Theory Methods for Spatial Kinetics," *Prog. Nucl. Energy*, **30 (2)**, 119-182 (1996).

[53] K. Tsujita, T. Endo, A. Yamamoto, "Application of the Multigrid Amplitude Function Method for Time-Dependent Transport Equation Using MOC," *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Sun Valley, Idaho, May 5–9 (2013).

[54] J. Wilert, H. Park, and W. Taitano, "Using Anderson Acceleration to Accelerate the Convergence of Neutron Transport Calculations with Anisotropic Scattering," *Nucl. Sci. Eng.*, **181**, 342-350 (2015).

[55] A. Wysocki, A. Gerlach, B. Kochunas, R. Salko, "Initial Implementation of Transient VERA-CS," CASL Technical Report CASL-U-2017-1303-000 (2017).

[56] B.C. Yee, E.W. Larsen, B. Kochunas "An analytical derivation of transport-corrected P0 cross sections and diffusion coefficients," *Proc. PHYSOR 2016*, Sun Valley, Idaho, May 1–5 (2016).

[57] A. Zhu, Y. Xu, T. Downar, "A Multilevel Quasi-Static Kinetics Method for Pin-Resolved Transport Transient Reactor Analysis," *Nucl. Sci. Eng.*, **182**, 435-451 (2016).