Modeling Feedback Effects of Transient Nuclear Systems Using Monte Carlo

by

Miriam A. Kreher

B.S., University of Pittsburgh (2016)

Submitted to the Department of Nuclear Science and Engineering
and Center for Computational Science and Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational Nuclear Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Miriam A. Kreher
Department of Nuclear Science and Engineering
and Center for Computational Science and Engineering
May 22, 2022

Certified by: Benoit Forget
Korea Electric Power Professor of Nuclear Engineering
Head and Professor of Nuclear Science and Engineering

Accepted by: Ju Li
Battelle Energy Alliance Professor in Nuclear Engineering
Professor of Materials Science and Engineering
Chair, Committee on Graduate Theses

Accepted by: Nicolas Hadjiconstantinou
Professor of Mechanical Engineering
Co-Director, Center for Computational Science and Engineering

Modeling Feedback Effects of Transient Nuclear Systems Using Monte Carlo

by

Miriam A. Kreher

Submitted to the Department of Nuclear Science and Engineering
and Center for Computational Science and Engineering
on May 22, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational Nuclear Science and Engineering

Abstract

Monte Carlo neutron transport is the gold standard for accurate neutronics simulation of nuclear reactors in steady-state because each term of the neutron transport equation can be directly tallied using continuous-energy cross sections rather than needing to make approximations in energy, angle, or geometry. However, the time dependent equation includes time derivatives of flux and delayed neutron precursors which are difficult to tally. While it is straightforward to explicitly model delayed neutron precursors, and thus solve the time dependent problem in Direct Monte Carlo, this is such a costly approach that the practical length of transient calculations is limited to about 1 second. In order to solve longer problems, a high-order/low-order approach was adopted that uses the omega method to approximate the time derivatives as frequencies. These frequencies are spatially distributed and provided by a low-order Time Dependent Coarse Mesh Finite Difference diffusion solver. While this scheme has been previously applied to prescribed transients, thermal feedback is now incorporated to provide a fully self-propagating Monte Carlo transient multiphysics solver which can be applied to transients of several seconds long.

Several recently developed techniques are used in the implementation of the proposed coupling approaches. Firstly, underrelaxed Monte Carlo, which is a steady-state technique that stabilizes the search for temperature distributions, is applied to find initial conditions. Secondly, tally derivatives are a Monte Carlo perturbation technique that can identify how a tally will change with respect to a small change in the system. Test problems of varying complexity are carried out in flow-initiated transients to show the versatility of these methods.

Overall, this multi-level, multiphysics, transient solver provides a bridge between high fidelity Monte Carlo neutronics and the fast multi-group diffusion methods that are currently used in safety analysis.

Thesis Supervisor: Benoit Forget
Title: Korea Electric Power Professor of Nuclear Engineering
Head and Professor of Nuclear Science and Engineering

Acknowledgments

I would like to thank everyone at MIT who supported me during graduate school, making it an overall fun and exciting time in my life. A special thank you to Professors Benoit Forget and Kord Smith as well as to all the members of the Computational Reactor Physics Group for pushing me to become a better scientist and engineer. Thank you to the OpenMC community for your many questions and answers throughout the years.

Thank you to my undergraduate professors at the University of Pittsburgh who got me here and have continued to help me regularly, Professor Dan Cole and Dr. David Griesheimer.

To my husband, Seth Kreher, I am grateful that you became a part of my life at just the right moment and for your constant help and understanding.

I am dedicating this thesis to my grandfather, Arthur Samuel Rathbun. I am thrilled to have developed such a close connection to his life's work.

This research has been funded by the Department of Energy Computational Science Graduate Fellowship under grant DE-FG02-97ER25308. This research was also supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy (DOE) Office of Science and the National Nuclear Security Administration. This work made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

Contents

1 Introduction **27**

1.1 Motivation 27

1.2 Nuclear Transients 28

1.3 Objectives and Structure of this Thesis 29

2 Steady-State Coupling **33**

2.1 Background 33

2.2 Iterative instability 36

2.3 Code setup 38

2.3.1 Thermal hydraulics 38

2.3.2 Power updates 40

2.4 Numerical results 41

2.4.1 Fully converged Monte Carlo coupling 41

2.4.2 Reduced-batch Monte Carlo 45

2.4.3 Ramp-up Monte Carlo 51

2.5 Summary 53

3 Time Dependent Analysis **55**

3.1 Background 55

3.2 Dynamic Monte Carlo 59

3.3 High-Order/Low-Order Methods 63

3.3.1 Methods 64

3.3.2 Results 79

3.4 Omega Method for Monte Carlo . . . 96
3.4.1 Results . . . 99
3.4.2 Conclusion . . . 106

4 Thermal Feedback Coupling **109**
4.1 Transient Thermal Feedback Model . . . 109
4.1.1 Null transient of the thermal feedback model . . . 111
4.1.2 Inlet change transient of the thermal feedback model . . . 114
4.1.3 TD- TH/CMFD coupling . . . 118
4.2 Tally Derivatives . . . 121
4.2.1 Tally derivative example (single group) . . . 122
4.2.2 Cross section derivative example (2-group) . . . 124
4.2.3 Derivatives for TD-CMFD . . . 129

5 Transient Implementation **135**
5.1 Fuel Rod Geometry . . . 135
5.1.1 Null transient considerations . . . 136
5.1.2 Inlet temperature transient . . . 140
5.2 Uniform Assembly Geometry . . . 170
5.2.1 Null transient considerations . . . 176
5.2.2 Uniform inlet temperature transient . . . 177
5.2.3 Radial effects with inlet temperature transient . . . 182

6 Conclusions **191**
6.1 Summary of thesis contributions . . . 191
6.1.1 Time dependent . . . 192
6.1.2 Multiphysics . . . 193
6.1.3 Multi-scale . . . 194
6.2 Future Work . . . 195
6.2.1 Reactivity-driven transients . . . 195
6.2.2 Further use of tally derivatives . . . 196

6.2.3 High fidelity thermal hydraulics 196

A Description of the 2D LRA Transient Benchmark 199

B Detailed derivation of the analytic precursor solution for the frequency transform method 201

C Detailed Description of Flux Factorization in the OpenMC Implementation of the Omega Method 205

C.1 Frequency Transform of the Flux and Precursor Time-Derivatives . . 208

C.2 Approximation of the Flux and Precursor Frequencies 209

C.3 Frequency Treatment in Monte Carlo 214

C.4 Delayed Neutron Precursor Integration 215

C.5 Computing Coarse Mesh Diffusion Coefficients 215

List of Figures

2-1 Illustration of cells (i) and cell interfaces (i±1/2). 39

2-2 Shannon entropy as reported in OpenMC after each batch of neutrons for a fuel rod geometry with an entropy mesh of 30 axial cells. After 100 batches, the entropy is within the margin of the remaining calculation, which justifies the choice of 100 inactive batches. 42

2-3 Doppler feedback instabilities in a fuel rod using progressively worsening fuel thermal conductivities: case 1 (top left), case 2 (top right), case 3 (bottom left), and case 4 (bottom right). Values are absolute value of max relative change in linear power across all cells (cells are each 12.192 cm long). 43

2-4 Axial power profiles for the Doppler feedback instabilities in a fuel rod with thermal conductivity associated with case 4. 43

2-5 Numerically damped Doppler feedback instabilities in a fuel rod with the thermal conductivity of case 4. Values are absolute value of max relative change in linear power. 44

2-6 Axial power profiles for the numerically damped Doppler feedback instabilities. 44

2-7 Doppler feedback instabilities in a fuel rod with the thermal conductivity of case 4 with progressively smaller number of batches in each cycle: 1000 (top left), 500 (top right), 300 (bottom left), and 100 (bottom right). Values are absolute value of max relative change in linear power. 46

2-8 Axial power profiles for the Doppler feedback instabilities in a fuel rod with the thermal conductivity of case 4 with progressively smaller number of batches in each cycle: 1000 (top left), 500 (top right), 300 (bottom left), and 100 (bottom right). In each plot, cycles 5-8 are shown. 47

2-9 Axial power profiles for the linear power in a fuel rod with the thermal conductivity of case 4 comparing the 8th cycle from Figure 2-6 and 2-8. 48

2-10 Axial power profiles for the linear power in a fuel rod with the thermal conductivity of case 4 comparing the 8th cycle with and without use of inactive batches in the 100-batch reduced-batch Monte Carlo technique. 49

2-11 Doppler feedback instabilities in a fuel rod with the thermal conductivity of case 4 comparing the 8th cycle with (left) and without (right) inactive batches in the 100-batch reduced-batch Monte Carlo technique. 49

2-12 Axial power profiles in a fuel rod with the thermal conductivity of case 4 comparing the 8th cycle of 100-batch reduced-batch Monte Carlo and several cycles of single-batch Monte Carlo. 50

2-13 Axial power profiles for ramp-up coupling with the thermal conductivity of case 4 with the following batch sequence: 10, 50, 100, 200, 300, 400, 500 before hitting the fuel temperature fluctuation criteria of less than 10 K. 51

2-14 Ramp-up Monte Carlo damping of Doppler feedback instabilities. Values are absolute value of max relative change in linear power. 52

3-1 Best case FOM with DNP production of 0.001, Prompt/DNP of 1, and Emission fraction of 100-to-1. 62

3-2 Original case FOM with DNP production of 1, Prompt/DNP of 1, and Emission fraction of 1-to-1. 62

3-3 Improvement in FOM between best case and original case. 63

3-4 2D LRA geometry. 64

3-5 Example power evolution during the 2D LRA transient on a 1 cm mesh. 65

3-6 Assembly power densities normalized to 1 W/cc at steady-state, error with respect to results in Smith [1]. 65
3-7 Time stepping scheme for transient HOLO methods. 69
3-8 Example reactivity evolution during the 2D LRA transient on a 1 cm mesh. 71
3-9 Summary of HOLO methods and their relationships. 79
3-10 Power profile shows that the fixed shape method is inadequate. . . . 81
3-11 Zoom-in to the first peak (Fixed shape method is not visible in this view). 81
3-12 k-balance values over time. 82
3-13 k-balance deviation from 1 in pcm over time. 82
3-14 Prompt frequencies $[s^{-1}]$ as calculated by the omega method. 83
3-15 Precursor frequencies $[s^{-1}]$ for each delayed neutron group in the first cell of the geometry as calculated by the omega method. 83
3-16 Spatial distribution of delayed frequencies $[s^{-1}]$ for precursor group 2 at different points in time, as calculated by the omega method. 84
3-17 Normalized assembly power densities at the peak for adiabatic versus omega. In green and red, errors that are lower and higher for omega method, respectively. 85
3-18 Assembly temperatures at the end-of-transient for adiabatic versus omega. In green and red, errors that are lower and higher for omega method, respectively. 86
3-19 Power profile, zoomed-in. 87
3-20 Absolute error on power with respect to reference, in time, zoomed to the second peak. 88
3-21 k-balance values over time. 88
3-22 Normalized assembly power densities at the peak for modified versus hybrid alpha. In green and red, errors that are lower or higher for hybrid, respectively. 89

3-23 Assembly temperatures at the end-of-transient for modified versus hybrid alpha. In green and red, errors that are lower or higher for hybrid, respectively. 89

3-24 Comparing prompt frequencies in the FTM to the amplitude derivative term in the IQS. 92

3-25 Zoomed-in power profile. 92

3-26 Normalized assembly power densities at the peak for IQS versus FTM. In green and red, errors that are lower and higher for FTM, respectively. 93

3-27 Assembly temperatures at the end-of-transient for IQS versus FTM. In green and red, errors that are lower and higher for FTM, respectively. 94

3-28 Communication diagram between the high-order Monte Carlo solver and the low-order TD-CMFD solver. 99

3-29 Normalized core power density of the C5G7 geometry under a prescribed moderator density transient for 2 seconds. 102

3-30 The core reactivity of the C5G7 geometry under a prescribed moderator density transient for 2 seconds. 102

3-31 Relative difference (%) in pin powers between the omega method and adiabatic approximation at 1 second. 103

3-32 Spatial distribution of the precursor frequency terms in the C5G7 geometry at 1 second for each of the 8 delayed precursor groups (starting at group 1 in the top left-hand corner). The delayed frequency term ($\frac{\lambda}{\lambda+\omega^{D,i}}$) is dimensionless. 103

3-33 Evolution of the flux frequency terms in time ($\frac{\omega^{P,g}}{v}$). Flux frequencies are a spatially flat. 104

3-34 Evolution of the precursor frequency terms in the top left corner pincell in time ($\frac{\lambda}{\lambda+\omega^{D,i}}$). 104

3-35 End-of-transient power for omega and adiabatic methods in a fuel rod after a 2 second moderator density transient compared to the null transient. 105

3-36 Axial power difference between the omega and adiabatic methods in a fuel rod after a 2 second transient. 106

4-1 Steady-state axial temperature distributions. 111

4-2 The steady-state calculation of MC/TH precedes the time dependent scheme to provide input temperatures for the geometry. 112

4-3 Null transient behavior of the time dependent coolant (left) and fuel (right) temperature equations, as compared to steady-state temperatures using the power profile that generated the temperature distribution. 113

4-4 Null transient behavior of the time dependent coolant temperature equations, as compared to steady-state temperatures using the power profile tallied after the temperatures are imposed. 113

4-5 Coolant temperature solution of the inlet temperature rise with dt=6.25x10^{-4} s. 115

4-6 Absolute difference in axial temperature from the dt=6.25x10^{-4} s solution at 0.5s. 116

4-7 Log scale difference in axial temperature from the dt=6.25x10^{-4} s solution at 0.5s (using absolute values). 116

4-8 L-infinity norm of difference from the dt=6.25x10^{-4} s solution. 117

4-9 Fuel temperature solution of the zeroed power transient for 1 second with dt=6.25x10^{-4}s. 117

4-10 Time stepping illustration of the transient procedure with TD- TH/CMFD. 118

4-11 Flowchart of solution procedure. 119

4-12 Time stepping illustration of the TD- TH/CMFD procedure with the tally derivatives. 122

4-13 Flowchart of modified solution procedure with tally derivatives. . . . 130

4-14 A representative example of fuel temperature derivatives for a fission rate tally in a 1-D fuel rod geometry. The x-axis units are reactions per source particle per degree Kelvin. 131

4-15 A representative example of water density derivatives for a fission rate tally for select cells in a 1-D fuel rod geometry. The x-axis units are reactions per source particle per g/cc. 131

4-16 A representative example of water density derivatives for the thermal group for a fission rate tally for all cells in a 1-D fuel rod geometry. The x-axis units are reactions per source particle per g/cc. 132

4-17 A representative example of water density derivatives for the fast group for a fission rate tally for all cells in a 1-D fuel rod geometry. The x-axis units are reactions per source particle per g/cc. 132

4-18 A representative example of water density derivatives for a fission rate tally in a 1-D fuel rod geometry as summed by approximation in Equation (4.15). The x-axis units are reactions per source particle per g/cc. 134

5-1 Fluctuations in normalized power density [W/cc] during a null transient. The left-hand figure uses 50,000 particles per batch; the right-hand figure uses 800,000 particles per batch. 138

5-2 Fluctuations in normalized power density [W/cc] during a null transient. The particle count represents the number of particles per batch. 138

5-3 Fluctuations in normalized power density [W/cc] during a prescribed null transient. The particle count represents the number of particles per batch. 139

5-4 Time dependent progression of the coolant temperature during the inlet temperature transient. 140

5-5 Time dependent progression of the fuel temperature during the inlet temperature transient. 141

5-6 Time dependent normalized power density [W/cc] during an inlet temperature transient with varying outer time step sizes. 142

5-7 Evolution of the total cross section during the first outer time step under two types of weighting compared to a reference solution. Fast cross section 1/3 from the bottom of the rod (top left), fast cross section 2/3 from the bottom of the rod (top right), thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right). 143

5-8 Evolution of the thermal absorption cross section (defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$) during the first outer time step under two types of weighting compared to a reference solution. Thermal cross section 1/3 from the bottom of the rod (left), and thermal cross section 2/3 from the bottom of the rod (right). 143

5-9 Evolution of the downscattering cross section during the first outer time step under two types of weighting compared to a reference solution. Location 1/3 from the bottom of the rod (left), and location 2/3 from the bottom of the rod (right). 144

5-10 Evolution of the total cross section during the inner time steps under three types of weighting. Fast cross section 1/3 from the bottom of the rod (top left), fast cross section 2/3 from the bottom of the rod (top right), thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right). 145

5-11 Evolution of the thermal absorption cross section (defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$) during the inner time steps under three types of weighting. Thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right).146

5-12 Evolution of the downscattering cross section during the inner time steps under three types of weighting. Location 1/3 from the bottom of the rod (left), and location 2/3 from the bottom of the rod (right). . 146

5-13 Time dependent normalized power density [W/cc] during an inlet temperature transient with mixed time and density weighting. 147

5-14 Time dependent normalized power density [W/cc] during an inlet temperature transient with varying iterative strategies. 150

5-15 Iterative convergence during the four time steps without tally derivatives in the zeroth iteration. 151

5-16 Iterative convergence during all four time steps with tally derivatives in the zeroth iteration. 153

5-17 Evolution of the fast total cross section 1/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation. 154

5-18 Evolution of the fast total cross section 2/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation. 155

5-19 Evolution of the thermal total cross section 1/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation. 156

5-20 Evolution of the thermal total cross section 2/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation. 157

5-21 Evolution of the thermal absorption cross section 1/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation. 158

5-22 Evolution of the thermal absorption cross section 2/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation. 159

5-23 Evolution of the downscattering cross section 1/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation. 160

5-24 Evolution of the downscattering cross section 2/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation. 161

5-25 The thermal total cross section 1/3 from the bottom of the rod during the inner time steps in an interpolation scheme that leverages tally derivative extrapolation during all iterations. 163

5-26 Low accuracy power in an interpolation scheme that leverages tally derivative extrapolation during all iterations. 163

5-27 Power in a two-way tally derivative extrapolation scheme. 165

5-28 Evolution of the total cross section during the first outer time step compared to a reference solution. Fast cross section 1/3 from the bottom of the rod (top left), fast cross section 2/3 from the bottom of the rod (top right), thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right). 165

5-29 Evolution of the thermal absorption cross section (defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$) during the first outer time step compared to a reference solution. Thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right). 166

5-30 Evolution of the downscattering cross section during the first outer time step compared to a reference solution. Location 1/3 from the bottom of the rod (left), and location 2/3 from the bottom of the rod (right). 166

5-31 Evolution of the fission tally derivative with respect to fuel temperature at every outer time step. The x-axis units are reactions per source particle per degree Kelvin. 167

5-32 Evolution of the fission tally derivative with respect to water density at every outer time step. The x-axis units are reactions per source particle per g/cc. 168

5-33 17x17 assembly geometry from the x-y view, coloring by material. Since temperatures can be assigned to cells (as opposed to materials), all fuel, cladding, and gap materials are the same material definition across the geometry. However, densities must be assigned to materials, so water materials are different in every cell. 171

5-34 Four rods of the geometry from the y-z view, coloring by material, showcasing the individual water materials at every axial cell. 171

5-35 Four rods of the geometry from the y-z view, coloring by cell, showcasing the individual cells of every material at every axial cell. 172

5-36 Evolution of key convergence figures during the ramp-up Monte Carlo procedure. On the left: tally convergence and k-eigenvalue convergence increase with each cycle. On the right: the root mean square change in power between consecutive cycles, with the final change being of 0.5%. 173

5-37 Evolution of the power difference from the assembly-average for select cycles: 20 batches, 40 batches, 60 batches, 80 batches, 100 batches, and 500 batches (starting at the top left). 174
5-38 Water temperature distribution in axial view of the assembly geometry. 174
5-39 Fuel temperature distribution in axial view of the assembly geometry. 175
5-40 Change in normalized power density [W/cc] during a null transient on the assembly geometry. 176
5-41 Absolute change in water temperature in Kelvin between t=1s and t=0s among all 289 fuel rods during the null transient. 177
5-42 Comparison of normalized power density change between identical transients applied to the fuel rod geometry and the assembly geometry. . 179
5-43 Change in water temperature profile in the assembly geometry between t=0s (left) and t=2s (right). 180
5-44 Change in water temperature profile in one rod of the assembly geometry between t=0s (left) and t=2s (right). 180
5-45 Change in precursor frequency terms ($\frac{\lambda}{\lambda+\omega^{D,b}}$) for group 1 delayed neutron precursors in the assembly geometry between t=0.5s (left) and t=1s (right). 181
5-46 Time dependent progression of the coolant temperature during the inlet temperature transient. 183
5-47 Assembly area affected by the inlet temperature change transient: 64 rods in the top left hand corner of the assembly. 183
5-48 Change in water temperature profile in the assembly geometry between t=0s (left), t=0.5s (middle), and t=1.0s (right) in axial view during a quarter-assembly inlet temperature change. 184
5-49 Radial shift in power, normalized to unity, at an axial cut at 1/3 the height from the bottom of the assembly for several time points of the transient. 185
5-50 Radial shift in power, normalized to unity, at an axial cut at 1/2 the height of the assembly for several time points of the transient. 186

5-51 Radial shift in power, normalized to unity, at an axial cut at 2/3 the height from the bottom of the assembly for several time points of the transient. 187
5-52 Axial power distribution, normalized to unity at t=0. 187
5-53 Change in precursor frequency terms ($\frac{\lambda}{\lambda+\omega^{D,b}}$) for group 1 delayed neutron precursors in the assembly geometry between t=0.5s (left) and t=1s (right) in axial view. 188
5-54 Change in precursor frequency terms ($\frac{\lambda}{\lambda+\omega^{D,b}}$) for groups 1-6 in the assembly geometry at t=1s in radial view at an axial cut at 1/2 the height of the assembly. 189
5-55 Change in normalized assembly power density during the quarter assembly inlet temperature change. 189

A-1 2D LRA geometry. 200

C-1 Plot of initial shape at the first outer time step. 211
C-2 Plot of initial amplitude at the first inner time step. 211

List of Tables

2.1 Comparing absolute fuel temperature changes between successive iterations for the numerically damped Doppler feedback instabilities. . . 45

2.2 Comparing absolute fuel temperature changes between successive iterations for the 100-batch solution of reduced-batch Monte Carlo. . . . 46

2.3 Comparing absolute fuel temperature changes between successive iterations for ramp-up Monte Carlo. 52

2.4 Comparing cost and convergence across coupling schemes.
*Max relative change in linear power at the last cycle
**Max absolute change in fuel temperature at the last cycle 53

3.1 Sensitivity of the transient results on time step size: comparison with a fine-time solution. 68

3.2 Sensitivity of the adiabatic method to outer time step size. 84

3.3 Sensitivity of the omega method to outer time step size. 84

3.4 Alpha eigenvalue results, obtained with 0.1s outer time steps. 87

3.5 Sensitivity of the coarse time integration method to outer time step size. 90

3.6 Sensitivity of the "stripped" coarse time integration method to outer time step size. 91

3.7 Time differencing results, obtained with 0.01s outer time steps. 91

3.8 Impact of frequencies on peak power root mean square error (RMS). . 95

4.1 Tally derivative values for each state. 123

4.2 Predicting k-eff with tally derivatives from state 1 123

4.3 Predicting k-eff with tally derivatives from state 2 124

4.4 Predicting k-eff with tally derivatives from state 3 124

4.5 Predicted cross sections in infinite cylinder for a water density of 0.70g/cc based on cross section derivatives tallied and calculated at 0.75g/cc. *absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$ 126

4.6 Predicted cross sections in infinite cylinder for a water density of 0.75g/cc based on cross section derivatives tallied and calculated at 0.70g/cc. *absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$ 126

4.7 Predicted cross sections in infinite cylinder for a fuel temperature of 800 K based on cross section derivatives tallied and calculated at 900 K. *absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$ 127

4.8 Predicted cross sections in infinite cylinder for a fuel temperature of 900 K based on cross section derivatives tallied and calculated at 800 K. *absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$ 127

4.9 Impact of cross section prediction error on total reactivity change prediction. *absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$ 128

5.1 Residual error of the calculated power during the transient after every iteration with 2 imposed iterations. 149

5.2 Residual error of the calculated power during the transient after every iteration with 3 imposed iterations. 149

5.3 Residual error of the calculated power during the transient after every iteration with 4 imposed iterations. 149

5.4 Residual error of the calculated power during the transient after every iteration with an iterative convergence criteria of 0.01. 149

5.5 Residual error of the calculated power during the transient after every iteration with an iterative convergence criteria of 0.005. 150

5.6 Runtime quantification of active cycles with and without tally derivatives based on data saved by OpenMC statepoint files after running on approximately 200 CPUs. 169

Chapter 1

Introduction

1.1 Motivation

As a safe and reliable source of carbon-free electricity, nuclear power has an important role to play in the world's clean power goals. To emphasize the safety of nuclear power, experiments and simulations go hand-in-hand: experiments provide the nuclear data, radiation effects, thermal hydraulic measurements, etc, while simulation is a tool to predict and test things that are too expensive to measure directly in the core.

As computer power has increased dramatically over the past few decades, the opportunities for relying on simulation have grown. In fact, in the nuclear weapons complex where testing is prohibited by law, a heavy emphasis on simulation has emerged, coupled with non-nuclear testing of weapon components. Such a reliance on simulation is possible thanks to high fidelity simulation which is in turn supported by high performance computing (HPC) or supercomputers.

High fidelity modeling and simulation remains a contemporary challenge of the nuclear industry. For example, using the state-of-the-art Monte Carlo method for neutron transport, it can take a CPU-hour to finely resolve a single fuel pellet of uranium. An entire fuel rod can take a CPU-day; an assembly of fuel rods takes a CPU-year; a full-core calculation can take a CPU-decade. The exact times depend heavily on the desired tallies and their convergence. In fact, active work is being conducted to bring those numbers down. In 2010, the Monte Carlo code MC21

demonstrated a steady-state simulation of full-core, 3D, pin-resolved power converged to 1% standard deviation in 18 hours on 400 cores (less than one CPU-year) [2, 3]. Similar high fidelity resolution is more difficult in transient scenarios.

Even more challenging is multiphysics coupling which accounts for multiple physical phenomena that interact together. The coupled nature of neutron flux and the properties of water within the core in Light Water Reactors constitutes an important effect on safety and operations. In particular, the temperature of the fuel and the density of the water provide thermal feedback to neutron flux that inhibits reactor power from growing out of control. Modeling coupled physics at steady-state is becoming possible with greater fidelity, however, it can incur a large computational cost. This is all the more true in transients because the costly coupling calculations need to account for time dependence.

1.2 Nuclear Transients

Nuclear transients cover a large range of timescales. The shortest ones, such as in fast burst reactors, operate over microseconds, while Light Water Reactor rod movements can last between milliseconds to several seconds. Such transients might include startup physics testing, power control, or emergency shut downs. Dynamic rod worth measurements last a few minutes, safety analysis transients last several minutes, and other transients, such as poison buildup and decay, or fuel depletion, last for hours. The computational cost is highest in transients of milliseconds to several minutes, where the impact of delayed neutrons must be modeled carefully, as opposed to shorter transients in which they can be ignored, and longer transients which are slow enough for prompt and delayed neutrons to be modeled in secular equilibrium. Cross section changes due to temperature dependent neutron data are also nearly instantaneous which adds to the large expense for transients of a few seconds long. Furthermore, simulations of time dependent reactor behavior face the computational challenge of resolving both space and time during reactivity or flow changes.

These transient scenarios, where time dependent reactor power is modeled with thermal feedback are a major component of safety analysis studies. Monte Carlo neutron transport is an important, high fidelity validation tool which requires further research to extend to transients on the order of tens of seconds. The goal of this thesis is to provide this capability.

The major feedback effects considered in this thesis are fuel temperature Doppler feedback and moderator temperature/density feedback in Light Water Reactors. The temperature of the fuel is a source of negative thermal feedback because higher temperatures cause the absorption resonances in uranium to broaden, resulting in an increase in absorption in the resonance region and a subsequent decrease in thermal fission. The moderator temperature is of interest mostly for how it alters moderator density. In general, when moderator density decreases (as moderator temperature increases), the moderator-to-fuel ratio decreases which results in less moderation and therefore less thermal fission. However, the initial moderator-to-fuel ratio plays a significant role in this feedback mechanism, and Light Water Reactors are designed to be slightly under-moderated to ensure that a decrease in moderator density remains a negative feedback effect.

1.3 Objectives and Structure of this Thesis

This thesis showcases an efficient implementation of a self-propagating thermal feedback transient code in Monte Carlo. A self-propagating transient is defined here as needing only external events to be prescribed (such as an initiating change in flow parameters). This is more challenging than a fully prescribed transient where the evolution of material properties are known and imposed on the system. The implementation of any self-propagating scheme requires iteration to converge on the properties of the next time step. Iterating over Monte Carlo is typically cost-prohibitive because of the high computational cost of each simulation. This thesis accomplishes the iteration efficiently by using a High-Order/Low-Order (HOLO) scheme. The high-order method is Monte Carlo, using a time dependent transport equation. The low-order

method is Time Dependent Coarse Mesh Finite Difference (TD-CMFD) which is a diffusion solver that includes time dependent terms. Together, the high-order and low-order methods communicate to produce a consistent solution over the time of the transient. A thermal fluids solver is added into this scheme to provide the thermal feedback.

In Chapter 2, a survey of steady-state coupling tools is given to lay the groundwork for several state-of-the-art Monte Carlo techniques. Without a stabilization (or relaxation) technique, the steady-state iteration between neutronics and thermal hydraulics tends to be unstable. Several techniques are presented to stabilize this iteration, specifically taking into account the unique features of Monte Carlo.

In Chapter 3, a comprehensive review of transient methods is presented which leads to the use of the omega method, a HOLO eigenvalue implementation of the frequency transform, as the logical choice for this thesis. A deterministic implementation on the 2D LRA benchmark and a Monte Carlo implementation of the C5G7 benchmark geometry both emphasize the value added by the omega method for prescribed transients.

In Chapter 4, the addition of time dependent thermal hydraulics to transient HOLO Monte Carlo creates a multiphysics, multi-scale solver and constitutes a major contribution of this thesis. Various coupling strategies aim to optimize the balance between the accuracy and cost of Monte Carlo. Among these coupling approaches, tally derivatives are tested. Tally derivatives are a perturbation technique that estimates how much a Monte Carlo tally will change with respect to a change in the underlying properties (temperature, material density, or nuclide density for instance). In this work, tally derivatives are implemented to estimate changes in multigroup cross sections.

In Chapter 5, the methods detailed in Chapter 4 are tested and analyzed on a fuel rod geometry and an assembly geometry. These examples showcase the full coupling of time dependent Monte Carlo, time dependent coarse mesh finite difference, and time dependent thermal hydraulics with the efficiency improvements of tally derivatives. Analysis of cross section evolution and frequencies during the transient highlight the

action of the multiphysics calculation.

Altogether, this work pushes the boundaries of what can be accomplished given the computational constraints of Monte Carlo neutron transport. While the objective of this thesis is not a high-fidelity multiphysics solution of coupled transients, the aim is to prove that such multiphysics coupling can be accomplished efficiently while taking advantage of the many benefits of Monte Carlo neutronics. This assumes that the thermal hydraulic component of the feedback calculation is not the computational bottleneck of the simulation. Therefore, the thermal hydraulic calculations in this thesis are simplified compared to the state-of-the-art computational fluid dynamics methods of the research literature. Nonetheless, the novelty of this work rests in the use of a transient flow calculation with a transient Monte Carlo solver. Furthermore, the use of tally derivatives represents a first-of-its-kind implementation of reactivity predictions in time and space which are shown to add efficiency to the coupling scheme.

Chapter 2

Steady-State Coupling

2.1 Background

.

Large-scale calculations of reactor cores are by-and-large conducted by diffusion solvers. Although a simplification of the neutron transport equation, the diffusion approximation with its underlying data processing has been fine-tuned over the past several decades to be a robust option for solving Light Water Reactor neutronics and is the most tractable option for full-core analysis. Today, standard industrial codes like SIMULATE5 use three-dimensional, two-group diffusion theory coupled with thermal hydraulic feedback in Light Water Reactor analysis [4]. The steady-state diffusion equation is:

$$-\nabla \cdot D_g(r)\nabla\Phi_g(r)+\Sigma_{tg}(r)\Phi_g(r) = \sum_{g'}\left[\left(\frac{\chi_g}{k_{eff}}\nu_{g'}\Sigma_{fg'}(r) + \Sigma_{sg'\to g}(r)\right)\Phi_{g'}(r)\right], \quad (2.1)$$

where g is energy group, r is spatial position, D is the diffusion coefficient, Φ is the scalar neutron flux, χ is the fission yield, k_{eff} is the reactor multiplication factor, ν is the mean number of neutrons per fission, Σ_f is the macroscopic fission cross section, Σ_t is the macroscopic total cross section, and Σ_s is the macroscopic scattering cross section.

Homogenization equivalence is essential to the accuracy of diffusion solvers. In

general, all deterministic methods rely on lattice physics codes for good cross section generation. These lattice calculations "analyze axial segments of fuel assemblies, referred to as lattices, to determine the detailed spatial and spectral distribution of neutrons and photons across the segment. Once the flux distribution is known, the cross sections can be condensed and homogenized into the structure needed" [5]. Equivalence factors are a key aspect of maintaining reaction rates and leakage rates after a lattice calculation, as are discontinuity factors that preserve nodal balance [5, 6, 7, 8].

The transport equation, as opposed to the simplified diffusion equation, can be solved for full-core calculations using various approaches. The Method of Characteristics (MOC) is a leading approach for reactor analysis, although it still comes at a high computational expense. Multi-group data is used, but no spatial homogenization is needed for MOC. Most codes implement 2D planar MOC calculations in the radial direction and a 1D diffusion or transport calculation in the axial direction because there is little heterogeneity axially in Light Water Reactors [9, 10, 11, 12, 13]. With this approximation, MOC can be practically used, although Tramm, Gunow, and Gaston have developed improvements in fully 3D MOC full-core transport [14, 15, 16].

Commonly, a Coarse Mesh Finite Difference diffusion solver (CMFD) can be included in deterministic calculations for acceleration. Both nodal diffusion and MOC have been successfully accelerated using CMFD [17, 18, 19, 15, 20]. Equation (2.2) describes the CMFD relationship between a surface current and the flux in neighboring cells, based on the approximation of Fick's Law, using finite difference and the linear diffusion coupling term $\tilde{D}$, where J is the surface current, i is the cell number, and x is the x-dimension.

$$J^g_{i\pm1/2} = \frac{-2D^g_{i\pm1}D^g_i}{D^g_{i\pm1}\Delta^x_i + D^g_i\Delta^x_{i\pm1}}(\pm\Phi^g_{i\pm1} \mp \Phi^g_i) = -\tilde{D}^g_{i\pm1/2}(\pm\Phi^g_{i\pm1} \mp \Phi^g_i) \qquad (2.2)$$

Due to these approximations, the currents used in this diffusion solver are not guaranteed to match the higher order method that it is accelerating. Therefore, a non-linear diffusion coupling term, $\hat{D}$ is introduced in CMFD to force equivalence

with the higher order currents, denoted J_{HO}:

$$J^g_{HO,i\pm1/2} = -\tilde{D}^g_{i\pm1/2}(\pm\Phi^g_{HO,i\pm1} \mp \Phi^g_{HO,i}) - \hat{D}^g_{i\pm1/2}(\Phi^g_{HO,i\pm1} + \Phi^g_{HO,i}) \tag{2.3}$$

$$\implies \hat{D}^g_{i\pm1/2} = \frac{-\tilde{D}^g_{i\pm1/2}(\pm\Phi^g_{HO,i\pm1} \mp \Phi^g_{HO,i}) - J^g_{HO,i\pm1/2}}{(\Phi^g_{HO,i\pm1} + \Phi^g_{HO,i})} \tag{2.4}$$

As an acceleration method, CMFD is a type of High-Order/Low-Order method (HOLO). A HOLO solver pairs a high fidelity solver with a more rudimentary one to accelerate the solution.

In contrast to deterministic methods, Monte Carlo solutions to the neutron transport equation do not make approximations in angle, energy, or space. Every neutron in a Monte Carlo simulation is explicitly tracked and undergoes a random walk. All interactions and their outcomes are sampled from continuous-energy cross sections. In this way, every term in the neutron transport equation can be sampled with very high accuracy given a large number of simulated neutrons. The high fidelity aspect of Monte Carlo has made it an essential validation tool for the deterministic methods used in reactor licensing and steady-state safety calculations despite its high computational cost.

Doppler broadening, a physical feedback response of nuclear data to temperature changes in the system, requires temperature dependent cross sections in order to be accurately modeled. There are several possible approaches to temperature dependent data, as summarized in [21]: pseudo-materials that mix nuclides of different temperatures [22], kernel reconstruction [23], on-the-fly Doppler broadening [24], target motion sampling [25, 26], and windowed multipole [27]. The windowed multipole method (WMP) is used in this work. WMP leverages the strengths of the multipole formalism, a "physically and mathematically equivalent formulation of the resonance parameters found in nuclear data evaluations" for given poles [28], while low-order polynomials describe surrounding cross section resonances in the resolved resonance region in a temperature independent way. Then, Doppler broadening can be done analytically during neutron tracking, resulting in a lower memory calculation. For

the unresolved resonance and fast energy ranges, cross section data is loaded for a single temperature in OpenMC.

Monte Carlo can also be accelerated with CMFD. One early example of this uses multi-group Monte Carlo in a full-core calculation [29]. However, this hybrid approach is prone to instabilities [30], especially when multiphysics feedback is included, as described in detail in [31, 21].

2.2 Iterative instability

The simulation of Doppler broadening can be unstable as described in [32], however, that publication contains errors with respect to the software implementation of temperature updates. Therefore, this thesis provides an updated analysis that correctly uses OpenMC to study multiphysics coupling. Straightforward Picard iteration that solves neutronics (power) and thermal hydraulics (temperatures) sequentially is prone to convergence difficulties as the power and temperature distributions affect each other. Previous literature has shown local Doppler feedback can be a major source of iterative instability [33, 34, 31, 21]. In a parametric stability study, Gill et al. perform a series of computational experiments on a single Pressurized Water Reactor fuel rod with a range of coolant mass fluxes and clad resistivity values [33]. Nearly all of these experiments show power and fuel temperature oscillations, irrespective of the flow rate. This suggests that Doppler broadening is the major issue at play in the instabilities observed, rather than coolant density. A proposed solution to this issue is damping the simulation by means of underrelaxation, which can be defined numerically or physically. Numerical underrelaxation consists of including previous solutions in each update calculation by means of an underrelaxation factor, w:

$$x^{n+1} = (1 - w)x^n + wx^{n+1/2} \tag{2.5}$$

where x^{n+1} is the new solution update, for example flux, power, or temperature. If x^n is the previous power solution before the thermal hydraulics code has its update, then $x^{n+1/2}$ is after the thermal hydraulics code had its update. Thus x^{n+1} takes into

account several previous solutions, not just the most immediate, and w is the weighting factor. The outcome is that the changes between solution updates is slowed down since older solutions are being input into each update. Without large fluctuations in each step, the simulation is more likely to be stable. The factor w is arbitrary as it can be difficult to estimate the ideal amount of underrelaxation required in a system. In [31], a factor of 0.4 was employed to underrelax a coupled Monte Carlo and thermal hydraulics calculation accelerated by CMFD.

While this form of underrelaxation is most commonly applied to thermal hydraulics only, Dufek and Gudowski have developed a stochastic approximation based on the Robbins-Monro algorithm to be used with Monte Carlo coupling problems [35]. This method reduces the impact of feedback as the calculation approaches the final solution. It was demonstrated numerically to obtain an ideal convergence rate and is similar in concept to applying underrelaxation on the power, as carried out in [33], with a vanishing damping factor.

The same principle can be applied "physically" by smoothing the physics in each update with information from previous iterations. In the case of Monte Carlo, this can be done by running fewer batches of neutrons between temperature updates. Consequently, unconverged neutronics are used in determining each new temperature update. In practice, this increases the frequency of temperature updates and does not allow the neutronics to fully converge under an incorrect temperature distribution. Rather, each batch is simulated under a new temperature distribution with the corresponding temperature dependent cross sections. New batches use the fission sites of the previous batch as the initial source, so the overall simulation is converging toward the final solution while the desired biasing is being achieved. Since this strategy eliminates the need for full Monte Carlo runs between temperature updates, it introduces damping and acceleration at the same time. It is admittedly difficult to quantify how much underrelaxation has been performed. However, determining the proper numerical factor w is equally difficult as the effect of underrelaxation is not linear. Therefore, the extent of physical underrelaxation in Monte Carlo must be determined with trial-and-error, just as it is for numerical underrelaxation.

In the following sections, a steady-state fuel rod is simulated. The instabilities of typical neutronics/TH coupling are reproduced using fully converged neutronics information. Then, an underrelaxation of the system is implemented by varying the number of batches of Monte Carlo particles between thermal hydraulic updates.

2.3 Code setup

These calculations consist of a single, 365.76 cm LWR fuel rod model in OpenMC constructive solid geometry. The 1.25984 cm pitch of the fuel is filled with water and surrounded by reflective boundary conditions, while the top and bottom of the fuel rod have vacuum boundary conditions (no water reflector or end plugs are included). The rod model includes a helium gap and zirconium cladding. An enthalpy-based solver, described below, is used to determine temperatures and coolant density. Thermal expansion is ignored. The coupling is external, with data being passed between OpenMC and the thermal solver.

2.3.1 Thermal hydraulics

The temperature and density are calculated based on the enthalpy added to the coolant along a fuel rod, beginning with the inlet coolant at the base of the rod, as shown in equations (2.6), (2.7), (2.8), where $q'_i L$ is power tallied from Monte Carlo in cell i (the rate of heat generation, noted $\dot{q}$ in [36]), h is enthalpy, G is mass flux, A is flow area, T is temperature of the water, and IAPWS97 is the steam table Python library. An illustration of cells and cell interfaces is shown in Figure 2-1.

$$h_{i+1/2} = h_{i-1/2} + \frac{q'_i L}{G \cdot A} \tag{2.6}$$

$$T_{i+1/2} = \text{IAPWS97}(\text{Pressure}, h_{i+1/2}).T \tag{2.7}$$

$$T_i = \frac{T_{i+1/2} + T_{i-1/2}}{2} \tag{2.8}$$

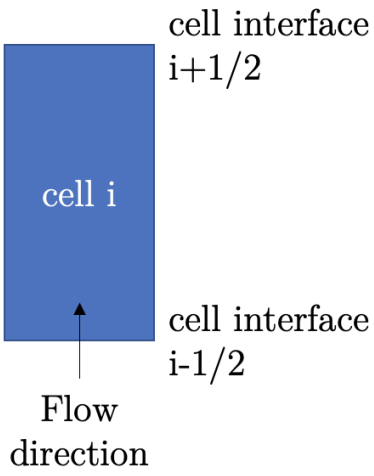

Figure 2-1: Illustration of cells (i) and cell interfaces (i±1/2).

Pressurized water reactor conditions are used, and a purely convective heat transfer coefficient is used for the coolant at the cladding surface (no nucleate boiling is taken into account), a modified Dittus-Boelter correlation given by equation (13.24) in [37] where Re is the Reynolds number, Pr is the Prandtl number, k is the liquid thermal conductivity, and D_e is the equivalent diameter:

$$\text{HTC} = 0.023 Re^{0.8} Pr^{0.4} \frac{k}{D_e} \tag{2.9}$$

Then, a heat transfer resistance model is used to calculate the temperatures on the inner cladding and gap surfaces (axial conduction effects are neglected):

$$T_{\text{fuel\_surface},i} = T_{\text{clad\_out},i} + \frac{q'_i}{2\pi}\left[\frac{1}{k_{\text{clad}}} ln\left(\frac{r_{\text{clad\_out}}}{r_{\text{clad\_in}}}\right) + \frac{1}{r_{\text{gap}} h_{\text{gap}}}\right] \tag{2.10}$$

where r is a radius, k_{clad} is the cladding conductivity, and h_{gap} is the conductance of the gap. Thermal resistance was calculated with a gap width of 0.00787 cm and conductance of 31 kW/m^2/K [38], cladding width of 0.05715 cm and conductivity of 16.08 W/m/K [39], and a fuel radius of 0.39218 cm with a temperature dependent fuel conductivity [40] where b is burnup in MWdays/kgU:

$$k_{f,i} = 1/(0.1148 + 0.0035b + 2.47510 \times 10^{-4}(1 - 0.0033b)T_{f,i}) + 0.0132e^{0.00188T_{f,i}} \tag{2.11}$$

$T_{f,avg,i}$ is the fuel temperature averaged axially over the mesh cell i [38]:

$$T_{f,avg,i} = T_{\text{fuel\_surface},i} + \frac{q'_i}{2\pi} \frac{1}{4k_{\text{fuel}}} \tag{2.12}$$

The Monte Carlo simulations in this thesis use a radially flat fuel temperature, while the fuel pellet temperature has a quadratic radial profile. Therefore, an average "Doppler fuel temperature" is calculated for the Monte Carlo simulation with a BE2 approximation: "The Doppler reactivity effects are linear with the perturbation in the square root of the fuel temperature. Therefore, k-effective values for any fuel temperature can be well approximated using the [...] flat temperature data. Performing this procedure, the Doppler reactivity worth can be evaluated for the different effective Doppler temperature definitions [...] namely 'BE2.'" [41]. In this work, the BE2 approximation is used with a value of $\omega = 0.92$:

$$T_{f,i} = \omega \times T_{f,avg,i} + (1 - \omega) \times T_{\text{fuel\_surface},i} \tag{2.13}$$

For the Monte Carlo simulation, all temperatures are cell-averaged and applied to the OpenMC cells. Moderator densities are also cell-averaged and applied to the OpenMC materials that fill the water cells.

2.3.2 Power updates

The power distribution is calculated with an OpenMC fuel rod model compliant with the BEAVRS benchmark, excluding spacer grids [42, 43]. The mesh applied to the geometry has 30 axial cells and is the same mesh for the neutronics and thermal hydraulics. The corresponding temperature dependent data is generated from windowed multipole representation as described in 2.1. The recoverable fission power is tallied by assuming that all power is deposited directly into the fuel. This tally is scaled at every iteration between neutronics and thermal hydraulics such that the fuel rod produces a constant 67 kW, the average power produced per rod in a typical 3411 MWth PWR [44].

2.4 Numerical results

Using these tools, stability and convergence of several physically underrelaxed coupling schemes are compared. The figures in this section represent the maximal relative power change over a number of coupling cycles. A successful scheme should minimize this infinity norm while converging to the correct solution.

2.4.1 Fully converged Monte Carlo coupling

Firstly, a "regular" coupling scheme is performed whereby a fully converged Monte Carlo calculation (100 inactive batches, 1,000 active batches with 100,000 particles per batch) is performed before each temperature update cycle. After 100 inactive batches, the Shannon entropy of the system is stabilized within the margin of the remaining calculation as seen in Figure 2-2. After all 1100 batches, the eigenvalue is converged to within 9 pcm. In this work, a batch refers to the number of neutron histories simulated between Monte Carlo evaluations of the multiplication factor, whereas a cycle refers to the iteration between a Monte Carlo calculation and the external temperature update. In a cycle, power data is passed after the OpenMC calculation to the thermal solver, and then the temperatures are applied to the OpenMC input files before the next OpenMC calculation. The total number of particles in each temperature update cycle is 100 million. The simulation starts with an initial Monte Carlo power calculation using a flat temperature distribution (with fuel at 900 K and all other materials at 566 K). The resulting power distribution is used to calculate the temperature distribution for the first cycle. It is also possible to start with an initial temperature calculation using an assumed power distribution. In fact, this is potentially more efficient because even assuming a flat power distribution, the inlet and outlet temperatures are known based on heat balance.

The instabilities are shown to be related to changes in fuel thermal conductivity. This is expected as a result of the Doppler feedback which is heightened under thermal conductivity degradation, as seen for instance in irradiated fuel [45, 46]. For example, using equation (2.11) and changing the variable b (for burnup) to 0 (case

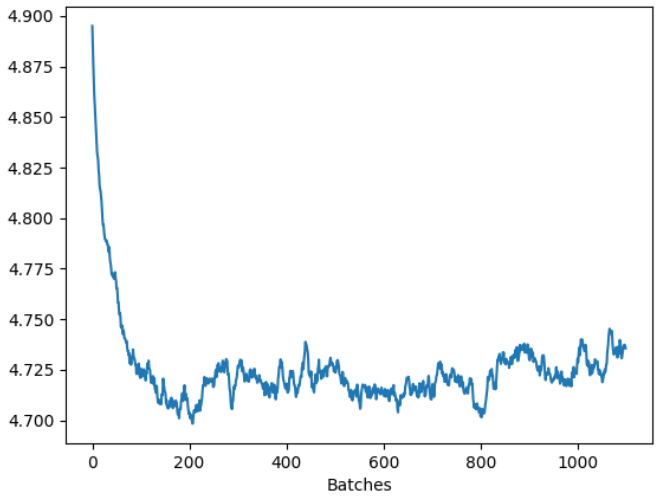

Figure 2-2: Shannon entropy as reported in OpenMC after each batch of neutrons for a fuel rod geometry with an entropy mesh of 30 axial cells. After 100 batches, the entropy is within the margin of the remaining calculation, which justifies the choice of 100 inactive batches.

1), 25 (case 2), 50 (case 3), and 75 (case 4) MWdays/kgU at 900K, the fuel thermal conductivities are 3.03, 2.53, 2.17, and 1.91 W/m/K, respectively. With these four conductivity conditions simulated in Figure 2-3 but no change to the fuel composition, the smallest fuel thermal conductivity shows the most substantial oscillations: after 8 costly repetitions of fully converged Monte Carlo neutronics calculations, the power continues to experience 20% change in the power solution (the final value of the max relative change in linear power is 0.22). Figure 2-4 shows how the axial power profiles oscillate between cycles as the coupling scheme struggles to converge.

As an example of how these oscillations can be damped, numerical underrelaxation of Equation (2.5) is applied to case 4 using a damping factor of 0.5. This is highly successful with the final value of the max relative change in linear power of 0.01. Based on Figure 2-5, the solution appears converged after the 4th cycle which can be further confirmed by looking at the plots of axial power distribution in Figure 2-6 or the absolute differences in fuel temperature in Table 2.1. Based on the results in this table, a convergence test based on a 10 Kelvin fluctuation criteria for the fuel temperature can be devised. Given this test, the damped simulation would have

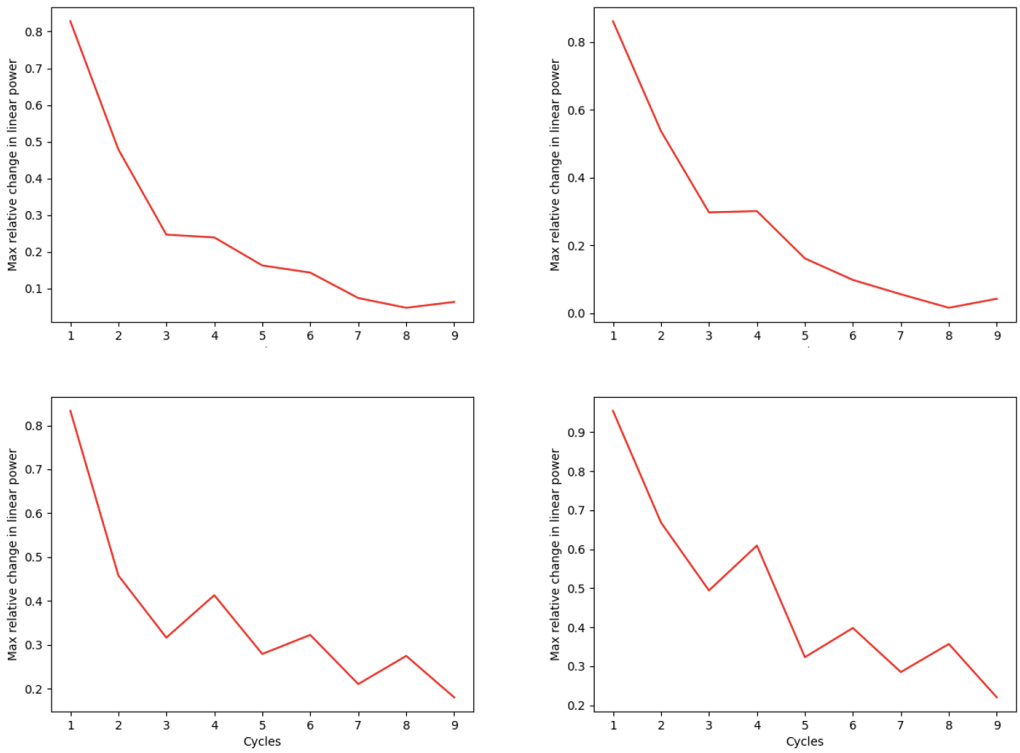

Figure 2-3: Doppler feedback instabilities in a fuel rod using progressively worsening fuel thermal conductivities: case 1 (top left), case 2 (top right), case 3 (bottom left), and case 4 (bottom right). Values are absolute value of max relative change in linear power across all cells (cells are each 12.192 cm long).

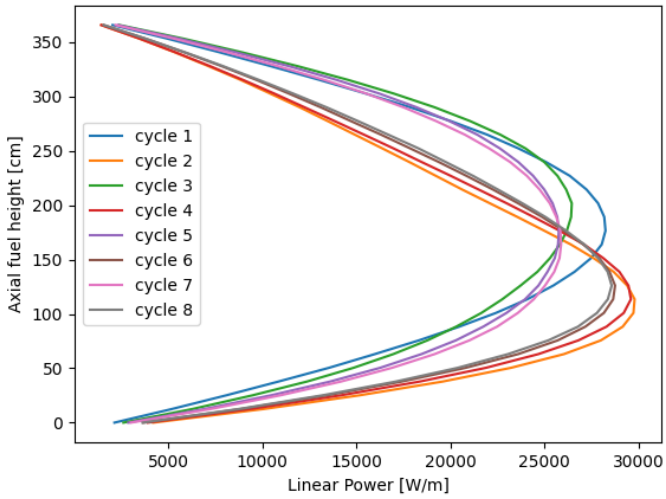

Figure 2-4: Axial power profiles for the Doppler feedback instabilities in a fuel rod with thermal conductivity associated with case 4.

cut off at 4 cycles, with a maximum relative linear power fluctuation of 0.01 and a maximum fuel temperature change of 7.03 K. This criteria also quantifies the level of statistical fluctuation that can still be expected from a converged power/temperature solution.

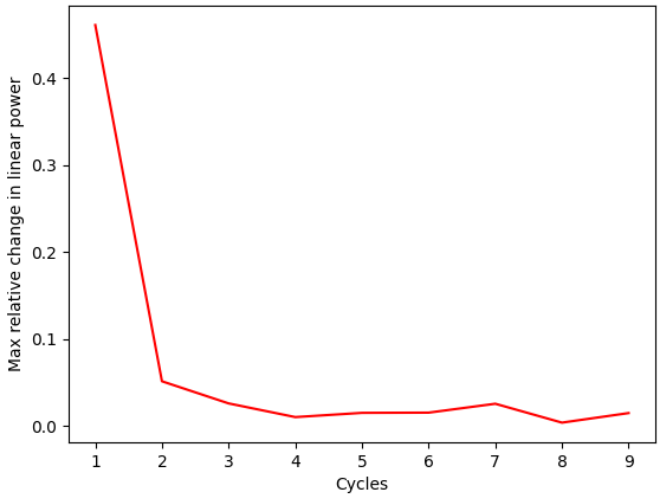

Figure 2-5: Numerically damped Doppler feedback instabilities in a fuel rod with the thermal conductivity of case 4. Values are absolute value of max relative change in linear power.

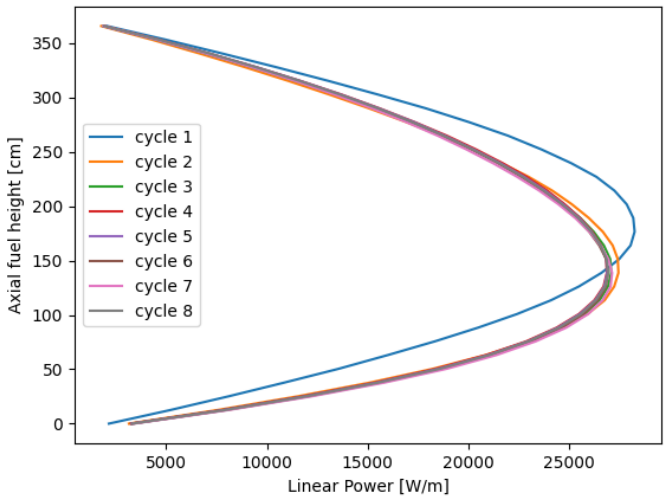

Figure 2-6: Axial power profiles for the numerically damped Doppler feedback instabilities.

Cycle	Max absolute change in fuel temperature across all cells [K]
2	125.02
3	11.97
4	7.03
5	3.18
6	5.05
7	6.50
8	9.55

Table 2.1: Comparing absolute fuel temperature changes between successive iterations for the numerically damped Doppler feedback instabilities.

While the successful damping of Doppler oscillations can be shown with numerical underrelaxation, it is often disadvantageous to run several fully converged Monte Carlo calculations to find only one steady-state temperature and power distribution. Therefore, more sophisticated and less costly methods are explored next.

2.4.2 Reduced-batch Monte Carlo

The goal of numerical underrelaxation is a "contamination" of the new solution with information from a previous solution. This slows the progression of the overall solution by preventing oscillations between widely varying intermediate solutions. In Monte Carlo neutronics, a practical way to slow the progression of the solution is to cut off the convergence before stationarity of the neutron distribution has been reached. Put differently, the neutron distribution is not permitted to fully converge under the wrong temperature distribution, and the temperatures are updated as the fission source is being converged. Several cases are run with this approach, reducing the number of batches per Monte Carlo calculation from 1000 batches used in Figure 2-3. Figures 2-7 and 2-8 show that as the number of batches is reduced, the oscillations appear to dampen.

Reduced-batch Monte Carlo is computationally very advantageous, as the cost for each cycle can be lowered by an order of magnitude compared to the full 1000-batch solution while appearing to significantly reduce axial oscillations. However, a side effect of this underrelaxation technique is the significantly reduced smoothness

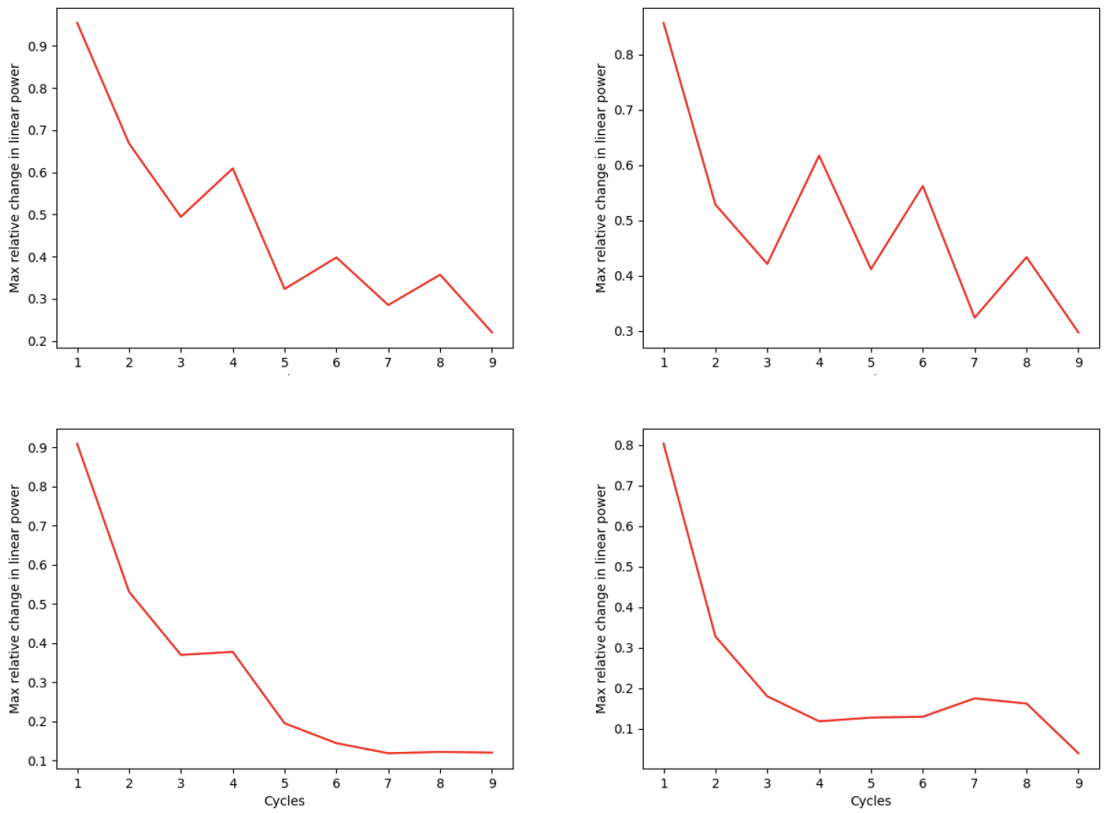

Figure 2-7: Doppler feedback instabilities in a fuel rod with the thermal conductivity of case 4 with progressively smaller number of batches in each cycle: 1000 (top left), 500 (top right), 300 (bottom left), and 100 (bottom right). Values are absolute value of max relative change in linear power.

Cycle	Max absolute change in fuel temperature across all cells [K]
2	210.42
3	145.29
4	101.13
5	48.74
6	47.72
7	62.23
8	49.78

Table 2.2: Comparing absolute fuel temperature changes between successive iterations for the 100-batch solution of reduced-batch Monte Carlo.

of the convergence compared to numerical damping. As seen in Figure 2-7, the linear power convergence of the 100-batch solution stagnates around 0.2, which is an order of magnitude higher than seen in Figure 2-5. At cycle 8, the changes in fuel temperature

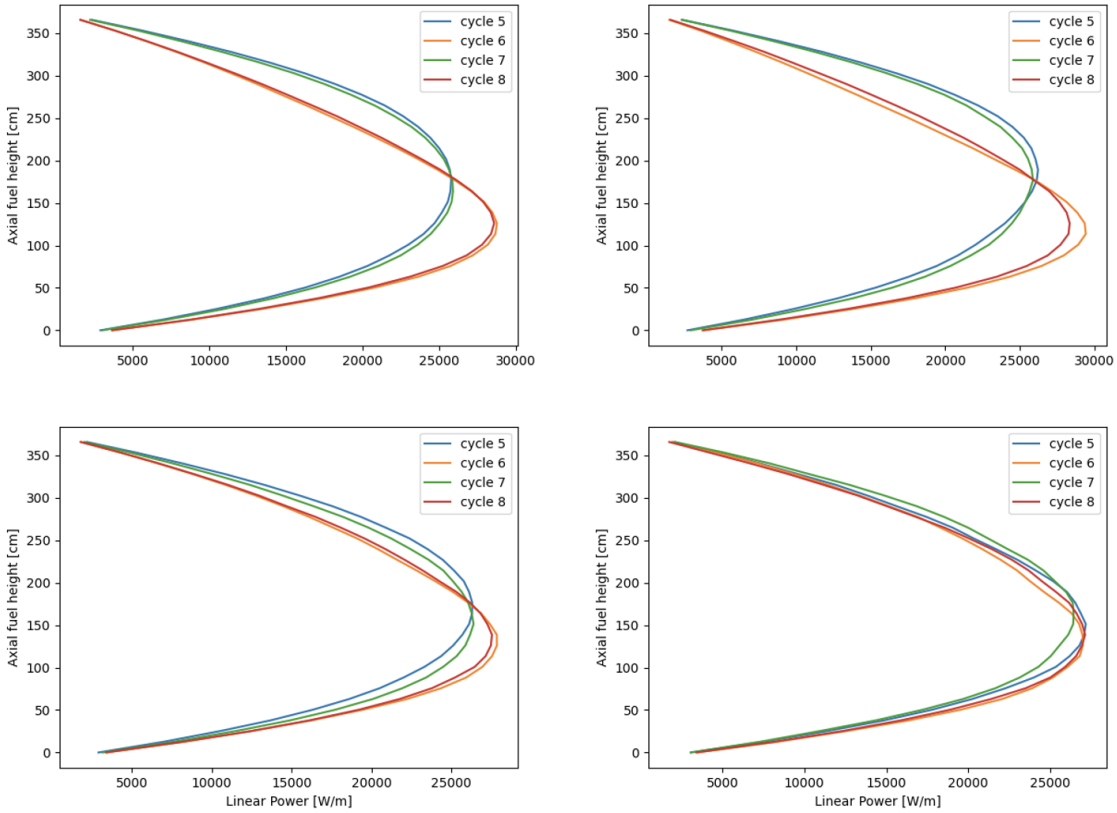

Figure 2-8: Axial power profiles for the Doppler feedback instabilities in a fuel rod with the thermal conductivity of case 4 with progressively smaller number of batches in each cycle: 1000 (top left), 500 (top right), 300 (bottom left), and 100 (bottom right). In each plot, cycles 5-8 are shown.

are of 50 Kelvin, as seen in Table 2.2 which is higher than the 10 Kelvin metric established from numerical damping. It is unclear if this is a result of insufficient convergence of the oscillatory behavior or the lack of statistical convergence arising from the reduced number of neutrons simulated at each cycle. Figure 2-9 compares the result from the 8th cycle of 100-batch reduced-batch damping and numerical damping. They overlap, which may indicate that the oscillatory convergence of the 100-batch underrelaxation is successful, but Figure 2-8 emphasizes that the solution is not the smooth shape that is expected of a fully statistically converged power distribution.

Further computational savings can be achieved by using the previous cycle's source distribution as a starting point. One way to implement this is through OpenMC's C-

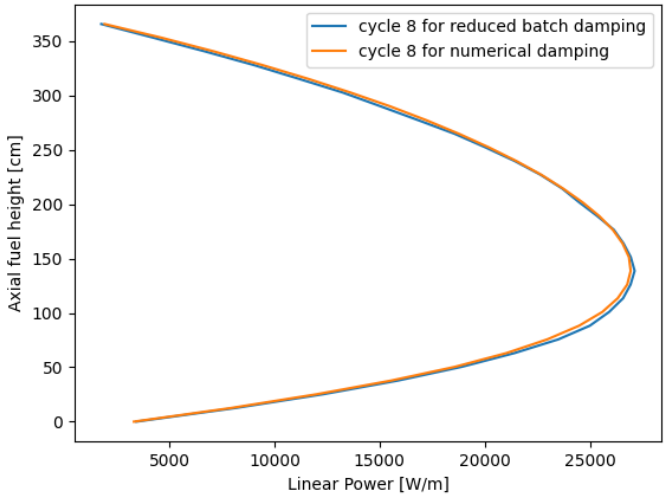

Figure 2-9: Axial power profiles for the linear power in a fuel rod with the thermal conductivity of case 4 comparing the 8th cycle from Figure 2-6 and 2-8.

API which allows for in-memory modification of material properties. The power tally can be flushed without interrupting the calculation, allowing the source distribution to be automatically used when the tally is re-started. Running the reduced-batch Monte Carlo technique with 100 active batches per cycle would proceed as:

1. Run the first 100 batches, then flush the tallies (these are the inactive batches of cycle 1).
2. Run the next 100 batches (these are the actives batches of cycle 1), then use the resulting tallies to calculate new temperatures. Change material properties in-memory. Flush the tallies.
3. Repeat steps 1 and 2 for all cycles.

Using the previous cycle's source either from a saved source file or the C-API behaves the same way and increases the convergence of the result as seen by comparing Figure 2-11 with the 100-batch solution of Figure 2-7. Another gain in computational efficiency can be achieved if the inactive batches are not used. In fact, the notion of damping the oscillations implies a numerical contamination of each simulation with information from the previous one. Therefore, removing the inactive batches allows

the new cycle to retain some information from the previous cycle by starting the new tally at the old source distribution, rather than first letting the neutrons settle under the new temperature distribution. Figure 2-10 shows that the same final result can be achieved with or without the inactive batches (within the noise of the underconverged 100-batch method), which makes a case for the computational savings of omitting them during the reduced-batch Monte Carlo underrelaxation process.

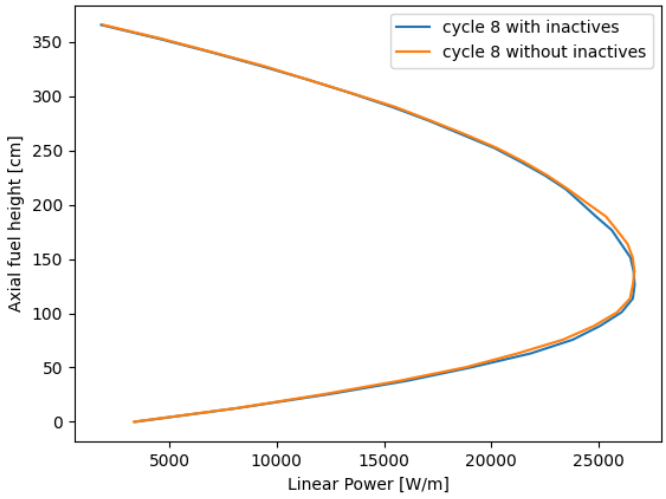

Figure 2-10: Axial power profiles for the linear power in a fuel rod with the thermal conductivity of case 4 comparing the 8th cycle with and without use of inactive batches in the 100-batch reduced-batch Monte Carlo technique.

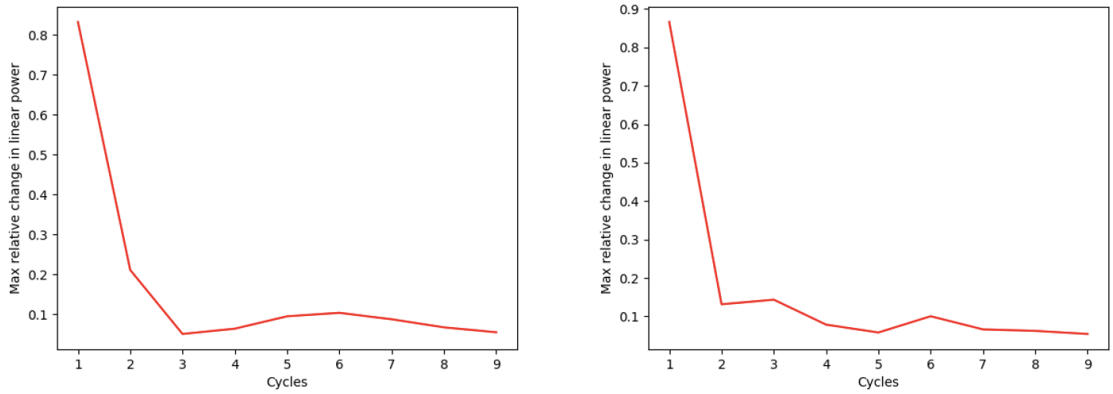

Figure 2-11: Doppler feedback instabilities in a fuel rod with the thermal conductivity of case 4 comparing the 8th cycle with (left) and without (right) inactive batches in the 100-batch reduced-batch Monte Carlo technique.

Single-batch Monte Carlo (SBMC) attempts to push reduced-batch Monte Carlo (RBMC) underrelaxation to the smallest possible batch size: only 1 batch of Monte Carlo neutrons is run between temperature updates. As expected, Figure 2-12 shows the large fluctuations that accompany such under-converged cycles: although there are no numerical instabilities, the statistical fluctuations are significant. It is possible that SBMC is advantageous in the early cycles, but looses its potential when used for all 1000 cycles, since the final solution is worse than the 100-batch RBMC without computational savings. An alternative implementation might average the tallies across a number of SBMC cycles to build a less noisy solution. A middle-ground approach may be a ramp-up Monte Carlo (RUMC), where each cycle's batches increase steadily, achieving a damped, converged solution.

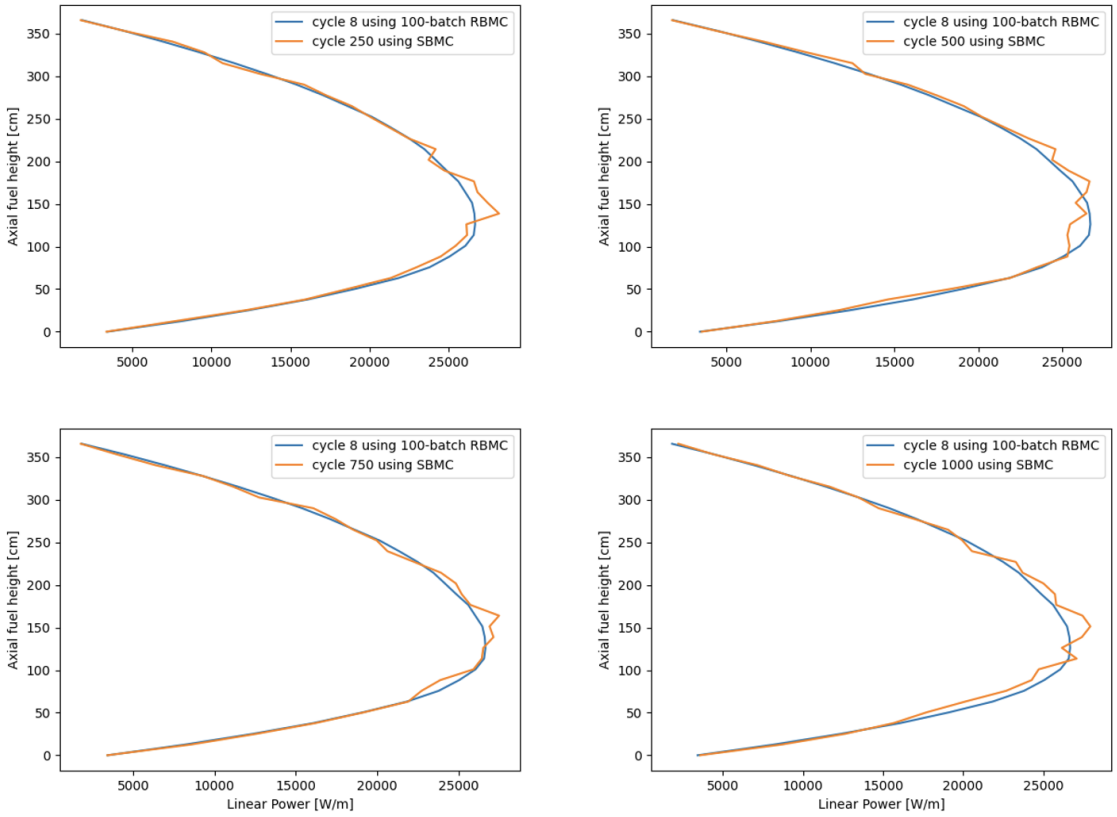

Figure 2-12: Axial power profiles in a fuel rod with the thermal conductivity of case 4 comparing the 8th cycle of 100-batch reduced-batch Monte Carlo and several cycles of single-batch Monte Carlo.

2.4.3 Ramp-up Monte Carlo

While single-batch Monte Carlo as a damping technique has merit for efficiently damping oscillatory behavior, its implementation throughout the entire coupling scheme is insufficient. As a result, a ramp-up technique is tested. Ramping up the number of particles in a Monte Carlo calculation has previously been used as an acceleration technique for source convergence in criticality calculations [47, 48, 49, 50].

This ramp-up starts with reduced-batch Monte Carlo and progressively increases the number of batches between tally flushes. Figures 2-13 and 2-14 show a possible ramp-up procedure of up to 1000 batches per cycle: 10, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000. A cut-off criteria stops the process once fuel temperature changes fall below 10 K in keeping with the metric determined above from numerical damping. The criteria is met after the 500-batch cycle, resulting in 7 cycles total, shown in Table 2.3.

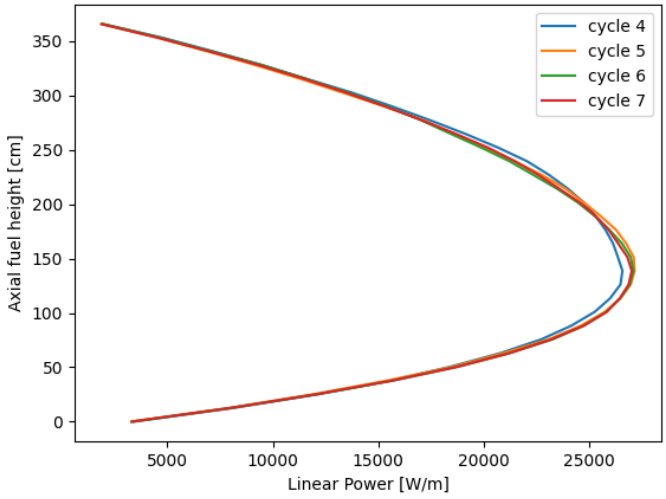

Figure 2-13: Axial power profiles for ramp-up coupling with the thermal conductivity of case 4 with the following batch sequence: 10, 50, 100, 200, 300, 400, 500 before hitting the fuel temperature fluctuation criteria of less than 10 K.

Table 2.4 summarizes the run parameters of the coupling strategies presented in this chapter. Neither RBMC nor SBMC achieve the desired fuel temperature fluctuation of 10 K, as their convergence appears to stagnate even after a large number

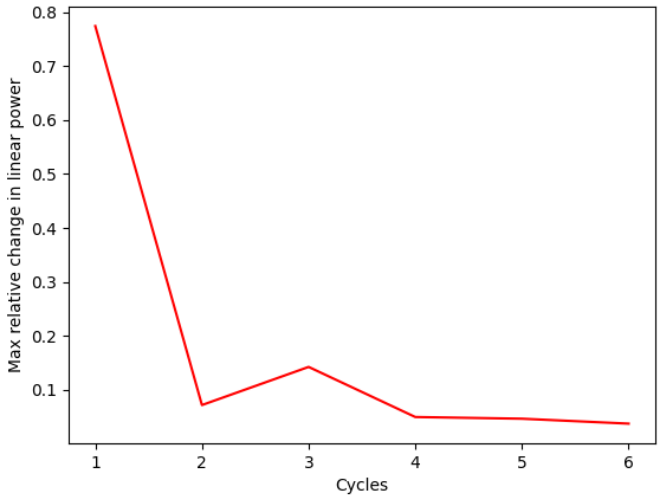

Figure 2-14: Ramp-up Monte Carlo damping of Doppler feedback instabilities. Values are absolute value of max relative change in linear power.

Cycle	Max absolute change in fuel temperature across all cells [K]
2	212.87
4	57.01
5	19.96
6	10.65
7	8.15

Table 2.3: Comparing absolute fuel temperature changes between successive iterations for ramp-up Monte Carlo.

of cycles. The power convergence of SBMC is much less tight without a much lower cost. Furthermore, from the above plots of RBMC and SBMC power profiles, their statistical underconvergence constitutes a major drawback. Finally, RUMC achieves a similar power convergence to RBMC with a fuel temperature convergence that is in line with numerical damping, suggesting that it is a more efficient way to dampen the problem using Monte Carlo. The ramp-up sequence for RUMC is flexible, so further experimentation may be able to optimize it. This approach closely resembles the use of the Robbins Monroe algorithm [35], where the damping factor becomes smaller at every cycle.

	Numerical Damping	100-batch RBMC (no inactives)	SBMC	RUMC
particles/batch	100,000	100,000	100,000	100,000
active batches	1000	100	1	1560
inactive batches	100	0	0	0
cycles	4	10	1000	7
particles/cycle	110 M	10 M	100,000	ramps up
total particles	440 M	100 M	100 M	156 M
power convergence*	0.01	0.05	0.15	0.04
10 K temperature test**	passes	fails	fails	passes

Table 2.4: Comparing cost and convergence across coupling schemes.
*Max relative change in linear power at the last cycle
**Max absolute change in fuel temperature at the last cycle

2.5 Summary

Multiphysics coupling presents a unique set of challenges in the solution of steady-state conditions for nuclear reactor cores. Numerical instabilities stem from the need to know power to determine temperature, and vice versa while the changes in neutron cross-sections create non-linear behavior. In this chapter, numerical underrelaxation has been shown to be effective for this type of multiphysics problem. Numerical underrelaxation is not necessarily cost prohibitive depending on the type of solvers used for neutronics and thermal hydraulics. However, the use of Monte Carlo puts its feasibility into question. This chapter showcases several implementations of efficient underrelaxation when Monte Carlo is involved. By leveraging batch-wise properties of Monte Carlo stochastic calculations, a successful damping technique is achieved when the number of batches between temperature updates is reduced compared to a full scale problem. For reduced-batch Monte Carlo and single-batch Monte Carlo, alternative tallying schemes (such as tallying over several cycles) need to be explored to surpass the convergence plateau associated with the statistical underconvergence of low particle counts. Ramp-up Monte Carlo was shown to be successful whereby the batches per cycle ramp up throughout the convergence process. Further work is necessary to determine ideal ramp sequences, which may include adaptive batching based on the convergence trends. These techniques can be used to determine the

initial conditions of a transient scenario. In the following chapters, the steady-state power and temperature distributions selected by ramp-up Monte Carlo will be used to kick off a time dependent scheme that holds steady-state and solves transients.

Chapter 3

Time Dependent Analysis

3.1 Background

Transient problems present an additional layer of complexity as they require solving time dependent neutron transport equations. Many of the same issues exist as for steady-state, with the added challenge of time discretization and the need to track delayed neutron precursors (DNPs). Unlike in steady-state, where prompt and delayed neutrons are considered to be in secular equilibrium, a transient is often characterized by the time delay incurred by neutrons emitted by delayed neutron precursors. In fact, if it were not for these delayed neutrons that reduce the reactor period, power reactors would be impossible to operate with mechanical controls. In this chapter, an overview of time dependent techniques is given which leads to the selection of the omega method as the most efficient and easiest to implement for Monte Carlo transients of multiple seconds duration.

Transient simulation began with the basic 0D point kinetics approximation. The Point Kinetics Equations, shown below, do not include any spatial resolution which makes them sufficient only for simple problems.

$$\frac{d}{dt}\Phi(t) = \frac{\rho(t) - \beta}{\Lambda(t)}\Phi(t) + \sum_i \lambda_i C_i(t), \tag{3.1}$$

$$\frac{d}{dt}C_i(t) = \frac{\beta_b}{\Lambda(t)}\Phi(t) - \lambda_i C_i(t) \tag{3.2}$$

where ρ is reactivity and Λ is prompt neutron lifetime. Today's industry codes rely on nodal diffusion in their kinetics solvers [51, 52, 53]. The transient diffusion equation has to account for the time derivatives and the delayed neutron sources. Therefore, it becomes a coupled set of equations between neutron diffusion and delayed neutron precursor concentrations:

$$\begin{aligned} &\frac{1}{v_g(r,t)}\frac{\partial}{\partial t}\Phi_g(r,t) - \nabla \cdot D_g(r,t)\nabla\Phi_g(r,t) + \Sigma_{tg}(r,t)\Phi_g(r,t) = \\ &\sum_{g'}\left[\left((1-\beta)\frac{\chi_g^p}{k_{eff}}\nu\Sigma_{fg'}(r,t) + \Sigma_{sg\to g'}(r,t)\right)\Phi_{g'}(r)\right] + \sum_i \chi_g^d \lambda_i C_i(r,t), \end{aligned} \tag{3.3}$$

$$\frac{\partial}{\partial t}C_i(r,t) = \beta_i \sum_{g'}\left[\frac{1}{k_{eff}}\nu\Sigma_{fg'}(r,t)\Phi_{g'}(r)\right] - \lambda_i C_i(r,t), \tag{3.4}$$

where i is the precursor group, t is time, v is neutron speed, β is the delayed neutron fraction, λ is the decay constant, C is the delayed neutron precursor concentration, and χ has a prompt (p) and delayed (d) component.

The solution of the transient equation requires an approximation to the time derivatives. In SIMULATE3-K, the frequency transform method [54] is used, which will be discussed in detail in this thesis. For the multiphysics coupling of transient diffusion with thermal hydraulics, an iterative scheme converges both the core thermal hydraulics and the neutron flux distribution at every time step. Similarly, the code PARCS uses an exponential transformation in addition to a discretization approximation of the time derivative called the theta method [55]. In such industry codes, cross section generation is exclusively computed by lattice calculations.

Modern deterministic research codes adopt a similar approach but can include more high fidelity methods at the expense of computational cost. For example, in the Multiphysics Object-Oriented Simulation Environment (MOOSE) [56], a number of physics applications are designed to interface specifically for solving coupled multiphysics problems. The MOOSE-based neutronics code Griffin relies on Monte Carlo generated cross sections, which can be generated at multiple states of the reactor, and

interpolates between these states. Furthermore, Griffin can use very detailed meshes because it is based on the finite element method. The Griffin workhorse is still diffusion, but first order S_N transport is an alternative higher fidelity option for smaller geometries. Griffin can then interface with a thermal hydraulic module, a fuel performance module, or other coupled physics modules to determine the feedback effects of the system.

Method of Characteristics (MOC) codes, as described in Chapter 2, are also capable of transient simulation [57, 58, 59, 60, 61, 62] by solving the time dependent MOC fixed-source equations on the MOC fine mesh. This is higher fidelity than diffusion calculations, but at a large computational expense, and therefore remains in the realm of research. Recent work has been applied to dramatically speeding up transient MOC by creating an MOC-equivalent diffusion problem which can be quickly solved using a special matrix decomposition. This work could lead to transient MOC-based real time simulators in specific scenarios [63].

Another high fidelity, high cost approach is transient Monte Carlo. Transient simulations in Monte Carlo have largely focused on the Dynamic Monte Carlo method [64, 65, 66], whereby the neutron histories that are tracked for steady-state calculations are supplemented with delayed neutron precursors. The precursors are explicitly tracked and then stochastically made to decay and emit a neutron in the process. There is such a large variation in precursor decay timescales that this method is very computationally costly. Furthermore, at any given time, there are thousands more precursors in a problem than neutrons, making the task of tracking them explicitly quite daunting. Many variance reduction techniques accompany this method to make it more practical [67]. Variance reduction is a manipulation of importance-weighting for individual particles whereby statistical variance is reduced for the same computational cost. Provably unbiased variance reduction techniques have been developed at Los Alamos National Laboratory for charged particle simulation [68, 69], and similar techniques can be applied to delayed neutron precursors [70]. While costly to implement for large reactors, the Dynamic Monte Carlo method is much more convenient for short transients in small geometries such as weapons calculations [71]. Multi-

physics coupling in Dynamic Monte Carlo has been successfully achieved in Monte Carlo codes Serpent [64] and Tripoli [66, 72] but remains costly. In practice, transient simulations by Dynamic Monte Carlo are typically limited to the order of 1 second.

An alternative to using very costly high fidelity techniques for transients is to employ high fidelity calculations at outer time steps and lower order time dependent calculations at inner time steps. In such a high-order/low-order (HOLO) scheme, the high-order method is used periodically to resolve the details of the flux, including shape. These outer time steps can be very coarse which provides large computational cost savings. The low-order method advances the problem in time by approximating the flux shape and focusing instead on the amplitude. HOLO methods have also been called the Multigrid Amplitude Function method (MAF) [73]. A recently published comparison of HOLO methods performs a rigorous one-to-one comparison of popular HOLO methods for transient problems [74]. In general, the separation into high-order and low-order methods is enabled by the separation of flux into shape (ψ) and amplitude (Φ) functions:

$$\Psi(r, E, \Omega, t) = \psi(r, E, \Omega, t)\Phi(r, E, \Omega, t) \tag{3.5}$$

where Ω is the angular dependence and E is the energy dependence. The nature of the methods that solve for the high- and low- orders influences what variables these functions depend on. In deterministic methods, it is common to couple diffusion (as high-order) and point kinetics (as low-order). These equations assume that the amplitude is only a time dependent scalar, resulting in:

$$\Psi(r, E, t) = \psi(r, E, t)\Phi(t). \tag{3.6}$$

However, higher fidelity deterministic HOLO implementations exist, such as transient Method of Characteristics [73]. In any HOLO implementation, the high-order method provides the information needed to calculate the key PKE parameters, ρ and Λ based on cross sections, flux, and adjoint flux. The main question is then how to approximate the time derivative terms of the transient equations. HOLO schemes

have been shown to be successful with Monte Carlo in order to extend the time-scales for practical transients, as has previously been done by using Dynamic Monte Carlo to tally parameters which can be used in a time dependent quasi-static predictor-corrector time stepping scheme [75] or by the frequency transform method [76, 77].

In this chapter, an example of improving the efficiency of the Dynamic Monte Carlo method will lead to the need for HOLO schemes in Monte Carlo. An overview of HOLO schemes for transient analysis will be presented to introduce the many possible approaches. Then, focusing more specifically on Monte Carlo, a promising transient technique will be shown in action: the omega method.

3.2 Dynamic Monte Carlo

Monte Carlo methods track neutron histories to obtain a steady-state solution to the neutron transport equation. It is important to simulate enough histories with enough separate batches of neutrons to obtain good statistics and to run the histories long enough for an equilibrium to be achieved. These histories are naturally amenable to time dependent tracking as well since the change in population size and distribution between generations can be sampled at little added cost. The largest computational cost comes from the simulation of delayed neutron precursors which are generated after every fission event. Their location in the fuel and their half-lives must be tracked in order for their subsequent emission of a delayed neutron to accurately impact the system. However, their accumulation rate is so high that their memory burden makes it practically intractable to model them explicitly. While neutrons live on the order of 10^{-4} seconds, delayed neutron precursors can live for many seconds before emitting delayed neutrons. Whereas neutrons can be quickly discarded from memory after undergoing their reactions, the relatively long life of a DNP causes them to build up in the system – up to thousands of times more than neutrons – and use large memory resources to store locations until they decay.

Dynamic Monte Carlo methods often include population control and variance reduction. Both require the manipulation of importance weighting to reach the same

average, unbiased result while forcing particles to behave in a desired way. This results in lower variance without increased computational cost. Sjenitzer [64] describes a population comb for delayed neutron precursors that is fundamental to making Dynamic Monte Carlo feasible. To illustrate the impact of variance reduction in Dynamic Monte Carlo, a few tests are presented. The geometry is a critical unreflected Godiva, a sphere of a uranium metal. The test procedure is:

1. Find the fundamental mode by picking an arbitrary source burst at t=0. Run in time-dependent mode with prompt neutrons only until they settle into the fundamental mode.

2. Run a single time step with DNP production turned on to determine their spatial distribution.

3. With the steady-state prompt neutron and DNP distributions, run the steady-state Godiva problem to compute a spatial flux distribution and a figure of merit in each spatial cell.

The following figure of merit (FOM) is evaluated for several permutations on the algorithm that impacts the DNP importance weighting. The goal is to determine which algorithm parameters improve the solution of the flux.

$$\text{FOM} = \frac{1}{\text{Var} \cdot T} \tag{3.7}$$

where Var is the variance of the results, and T is the compute time of the calculation. Therefore, precision and cost are both accounted for. This figure of merit was evaluated for flux, a spatially dependent variable, rather than global power as was used in early work on the Dynamic Monte Carlo method.

This procedure is run in the Monte Carlo Application ToolKit (MCATK) developed at Los Alamos National Laboratory [78] with explicit delayed neutron modeling. Permutations specific to how delayed neutron precursors are formed and emit delayed neutrons are tested. The following three parameters are varied:

1. DNP production ratio: the number of delayed neutron precursors produced at every fission. Test values include DNP production ratios of 1, 0.1, 0.01, and 0.001.

2. Prompt-to-DNP ratio: the ratio of the target number of prompt neutrons to the target number of delayed neutron precursors, where the target number is maintained by the population control algorithm. Tests include prompt-to-DNP ratios of 0.1, 1, 10, 100, and 1000.

3. Emission fraction: the fraction of delayed neutron precursors that emit a neutron at each time step. Tests include emission fractions of 0.001, 0.01, 0.1, 0.5, 0.8, and 1.

The figures showcase the spatially dependent improvements made to the figure of merit through the permutations. The x-coordinates are the distance from the center of the Godiva geometry. Figure 3-1 is the FOM for the "best case," that is, the case with the most improvement compared to the original case. The FOM of the original case is shown in Figure 3-2. Figure 3-3 is related to Figures 3-1 and 3-2 by taking the ratio of the best case to the original case to show the improvement factor. The cases that had the best FOM also had the most similar prompt and delayed neutron weights. This leads to the conclusion that methods that have more similar delayed and prompt neutron weights carry less variance and are therefore more likely to be the most efficient. These included DNP production ratios as low as 0.001 and the best prompt-to-DNP ratio of 100, the combination of which was an improvement in the FOM of a factor of 2 close to the center of the sphere and up to 2.3 at the exterior, as shown in Figure 3-3. Emission fractions were also useful, but less efficient than the other parameters, offering only a factor of 2 improvement across the geometry. As expected, it is more efficient to cut down on the number of DNPs created than storing DNPs that only emit delayed neutrons a fraction of the time. However, delayed neutron production should not be cut too severely. Overly aggressive weight manipulation results in too few delayed neutrons created at each time step and limits, or sometimes reverses the improvements seen in the FOM.

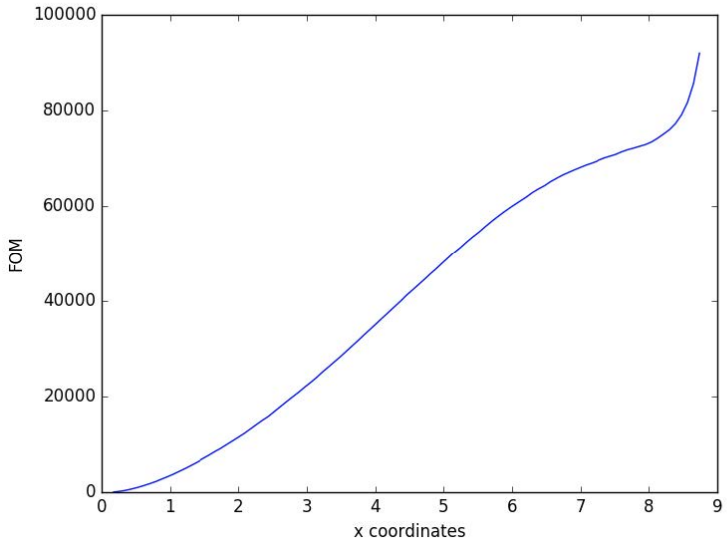

Figure 3-1: Best case FOM with DNP production of 0.001, Prompt/DNP of 1, and Emission fraction of 100-to-1.

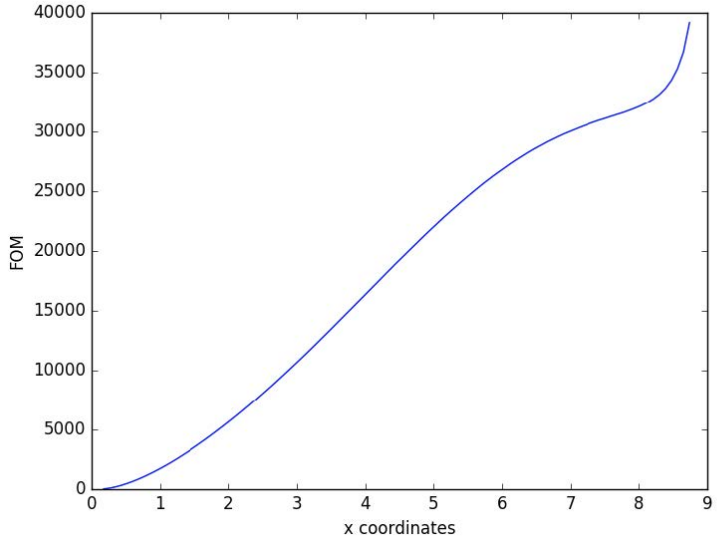

Figure 3-2: Original case FOM with DNP production of 1, Prompt/DNP of 1, and Emission fraction of 1-to-1.

Such weight manipulation and variance reduction is essential to the computational feasibility of Dynamic Monte Carlo. However, even with these techniques, the run times are impractical for transients on the order of several seconds. As an alternative to using overly expensive high fidelity solvers, a high-order/low-order scheme can be

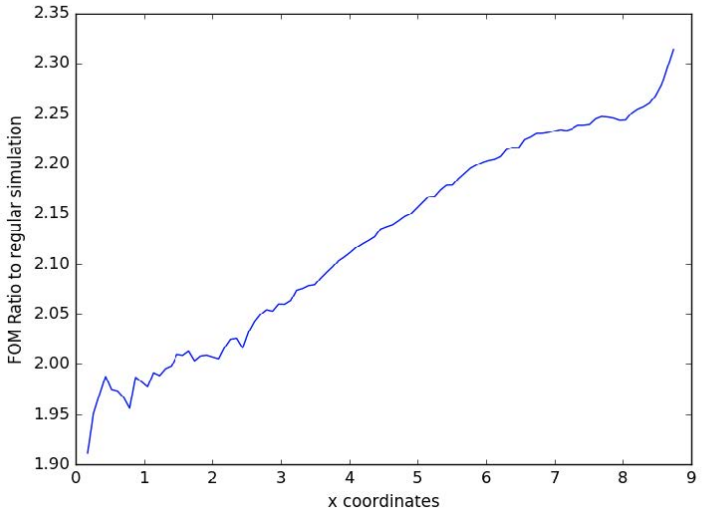

Figure 3-3: Improvement in FOM between best case and original case.

used to leverage high fidelity treatment while using a low-order approximation as an acceleration tool. In the following section, the most common transient HOLO methods are presented and quantitatively compared against each other.

3.3 High-Order/Low-Order Methods

HOLO methods pair a high fidelity solver with a more rudimentary one to accelerate the solution. Sometimes, HOLO methods are used at steady-state to resolve fine mesh problems in a multigrid scheme [19, 79, 80, 81]. In time dependent problems, the high-order method is used periodically to resolve the details of the flux, including shape, while the low-order method advances the problem in time by approximating the flux shape and focusing instead on the amplitude.

The methods in this comparative study include several eigenvalue methods: a fixed shape method for illustration only, the adiabatic method which is commonly used, the omega method which is an improvement on the adiabatic method, and the alpha eigenvalue method. Several time differencing methods are also compared: the frequency transform method, a coarse time integration (that is, a linear first order differencing with large time steps), and the improved quasi-static method. The goal

is to directly compare the accuracy of these methods on a common problem and establish clearly how these methods are related to each other. As such, the solvers chosen can be simple. The high-order solver selected is 2D diffusion, and the low-order solver is point kinetics.

3.3.1 Methods

Reference Solution

The comparison problem is the 2D LRA transient benchmark of a BWR quarter-core in which a control rod is withdrawn [82]. The geometry is a BWR quarter-core with assemblies that are 15x15 cm. The control rod is withdrawn in the region noted R in Figure 3-4 [83]. As the control rod is removed, there is thermal feedback in the system which causes the power spike to stabilize, as illustrated in Figure 3-5. See the full description of this numerical benchmark in Appendix A, along with an important note about values that differ from the original benchmark specification.

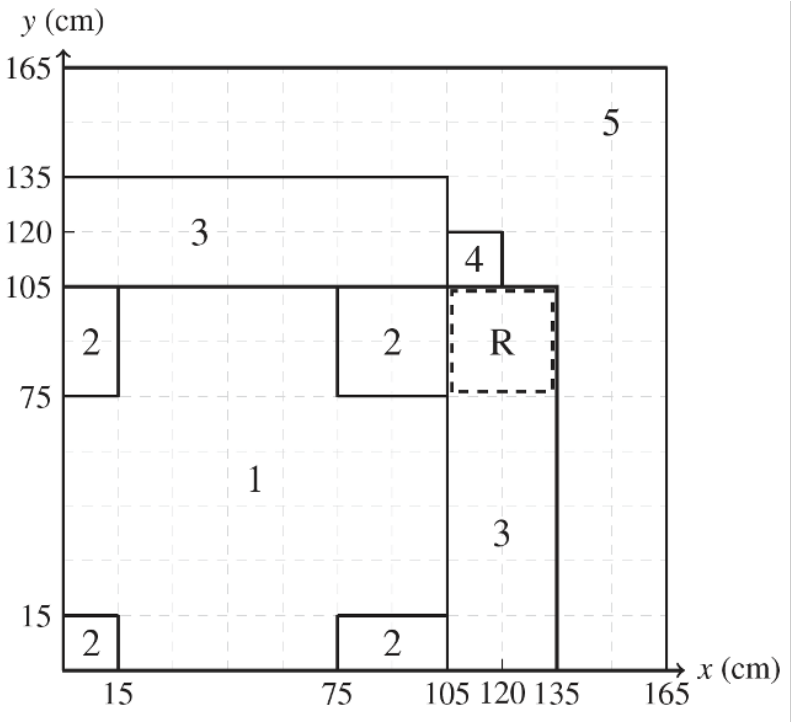

Figure 3-4: 2D LRA geometry.

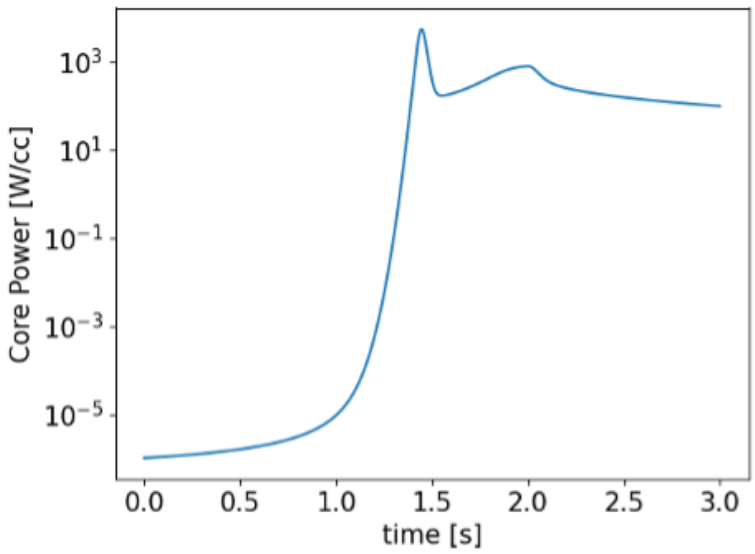

Figure 3-5: Example power evolution during the 2D LRA transient on a 1 cm mesh.

The steady-state problem must be solved to obtain initial values. Our steady-state 2-group diffusion calculation was found to be spatially converged with a uniform 1 cm mesh. Our results are compared to computational results in the literature [1]. Our steady-state eigenvalue was within 4 pcm of the reference, and the maximum error in normalized assembly-averaged power was below 1% relative error as shown in Figure 3-6. These results demonstrate that our code accurately depicts the geometry and steady-state solution of the benchmark.

							1.3280 -0.82	
						2.1610 -0.11	1.6210 -0.39	0.8465 -0.96
					1.8520 0.13	2.0510 0.02	1.6790 -0.23	0.9716 -0.60
	ref. value % rel. err.			0.8643 0.03	1.1520 -0.10	1.3390 -0.19	1.4220 -0.12	0.9325 -0.53
			0.5524 0.20	0.6782 0.13	0.8432 0.09	1.0220 0.00	1.2210 0.03	0.8530 -0.46
		0.4240 0.36	0.4921 0.27	0.6181 0.23	0.7826 0.19	0.9667 0.08	1.1730 0.09	0.8268 -0.36
	0.3995 0.53	0.4067 0.44	0.4904 0.36	0.6705 0.29	0.9398 0.19	1.1510 0.06	1.2810 0.20	0.8672 -0.27
0.6122 0.86	0.4402 0.54	0.4130 0.52	0.5118 0.44	0.7902 0.28	1.3860 0.47	1.6610 0.42	1.4810 0.16	0.9242 -0.21

Figure 3-6: Assembly power densities normalized to 1 W/cc at steady-state, error with respect to results in Smith [1].

The 2-group time dependent diffusion equations, coupled with the precursor balance equations, are discretized in space and time and solved using high-order/low-order methods. There is little information in the literature about transient benchmark results for fine spatial mesh solutions of the 2D LRA benchmark. Most transient reference solutions utilize coarse-mesh nodal methods, mostly on 15 cm meshes. While nodal methods can finely resolve flux, the benchmark's prescribed feedback and cell details are assembly-averaged throughout the transient which does not make a sufficiently detailed reference solution compared to finite difference. Therefore, the reference for the transient problem here is a temporally refined run of our fully implicit solver [84]. This reference uses an implicit time scheme with first order temporal differencing. Since the transient cross sections are prescribed, it is possible to update the cross sections before solving for Ψ^{n+1}. However, since the cross section change induced by Doppler feedback relies on the new temperature (see Appendix A), and therefore the new fission reaction rate, there is a lag. It is minor enough to avoid needing iterations given small time steps (0.01s or smaller).

Some problems are conducive to using higher order differencing methods such as BD2 (backward differencing of the second order, otherwise known as Crank-Nicolson). However, proof of higher order convergence is limited to cases with constant matrix properties. In a transient problem, where the matrix changes at every time step, higher order convergence is rarely observed. BD2 still tends be highly accurate with fine time steps. Given that the goal of this work is to look at longer time steps in a HOLO scheme, a BD1 approach is used:

$$\left(\frac{df}{dt}\right)^{n+1} = \frac{1}{\Delta t}(f^{n+1} - f^{n}) + O(\Delta t). \tag{3.8}$$

The diffusion equation, in mesh-centered finite difference with cell-averaged properties, discretized in time in an implicit scheme is shown in Equation (3.9) with Equation (3.10). The constant k_{crit} represents the value of k-eff in the initial critical state of the reactor. Since reactor models are not necessarily exactly critical, the transient equations must scale by k_{crit} to guarantee that a steady-state solution can

be maintained if no perturbations are made.

$$\frac{1}{v_g}\frac{\Psi_g^{n+1}-\Psi_g^n}{\Delta t} = \nabla\cdot D_g^{n+1}\nabla\Psi_g^{n+1} - [\Sigma_{r,g}^{n+1}]\Psi_g^{n+1} + \sum_{g'\neq g}\Sigma_{s,g'\to g}^{n+1}\Psi_{g'}^{n+1}$$
$$+(1-\beta_{tot})\frac{\chi_g^P}{k_{crit}}\sum_{g'}\nu_{g'}\Sigma_{f,g'}^{n+1}\Psi_{g'}^{n+1} \tag{3.9}$$
$$+\sum_i \chi_g^D\lambda_i C_i^{n+1}$$

with:

$$\frac{dC_i(t)}{dt} = -\lambda_i C_i(t) + \frac{\beta_i}{k_{crit}}\sum_{g'}\nu_{g'}\Sigma_{f,g'}(t)\Psi_{g'}(t) \tag{3.10}$$

By assuming that the flux shape changes linearly over the time step, the precursor equation has an analytic solution which removes much of the error associated with the time discretization of the precursor concentrations. This derivation is detailed in Appendix B for the frequency transform method. To apply those equations to the implicit method, set $\omega = 0$ to obtain equation (3.11) which matches Bandini [54] equation (5.7):

$$C_i(t^{n+1}) = C_i^n e^{-\lambda_i\Delta t}$$
$$+\frac{\beta_i}{k_{crit}\lambda_i}\sum_{g'}\nu_{g'}\Sigma_{fg'}\left[\psi_{g'}^n\left(-e^{-\lambda_i\Delta t}+\frac{1-e^{-\lambda_i\Delta t}}{\lambda_i\Delta t}\right)+\psi_{g'}^{n+1}\left(1-\frac{1-e^{-\lambda_i\Delta t}}{\lambda_i\Delta t}\right)\right] \tag{3.11}$$

Substituting into the diffusion equations (3.9) and (3.10) and assuming 2 energy groups, the group 1 equation is:

$$-\nabla\cdot D_1\nabla\Psi_1^{n+1} + \left[\Sigma_{r1}+\frac{1}{v_1\Delta t}\right]\Psi_1^{n+1}$$
$$-\left[\frac{(1-\beta_{tot})}{k_{crit}}+\sum_i\lambda_i\frac{\beta_i}{k_{crit}\lambda_i}\left(1-\frac{1-e^{-\lambda_i\Delta t}}{\lambda_i\Delta t}\right)\right][\nu_1\Sigma_{f1}\Psi_1^{n+1}+\nu_2\Sigma_{f2}\Psi_2^{n+1}]$$
$$=\sum_i\lambda_i\frac{\beta_i}{k_{crit}\lambda_i}\left(-e^{-\lambda_i\Delta t}+\frac{1-e^{-\lambda_i\Delta t}}{\lambda_i\Delta t}\right)[\nu_1\Sigma_{f1}\Psi_1^{n+1}+\nu_2\Sigma_{f2}\Psi_2^{n+1}] \tag{3.12}$$
$$+\sum_i\lambda_i C_i^n e^{-\lambda_i\Delta t}+\frac{\Psi_1^n}{v_1\Delta t}$$

and the group 2 equation is:

$$-\nabla \cdot D_2 \nabla \Psi_2^{n+1} + \big[\Sigma_{a2} + \frac{1}{v_2 \Delta t}\big] \Psi_2^{n+1} - \Sigma_{s1\to 2} \Psi_2^{n+1} = \frac{\Psi_2^n}{v_2 \Delta t} \tag{3.13}$$

Which can be re-written as:

$$\begin{cases} -\nabla \cdot D_1^{n+1} \nabla \Psi_1^{n+1} + \widehat{\Sigma_{r1}^{n+1}} \Psi_1^{n+1} - \sum_{g'} \widehat{\Sigma_{fg'}^{n+1}} \Psi_{g'}^{n+1} = S_1 \\ -\nabla \cdot D_2^{n+1} \nabla \Psi_2^{n+1} + \widehat{\Sigma_{r2}^{n+1}} \Psi_2^{n+1} - \Sigma_{s1\to 2}^{n+1} \Psi_1^{n+1} = S_2 \end{cases} \tag{3.14}$$

where the modified cross sections account for the additional terms including the time discretization terms and the analytic precursor terms. This form resembles steady-state equations, allowing us to solve each step in the transient problem similarly to a flux matrix inversion in fission source iteration. A Gauss Seidel iteration scheme is used to do so with very tight convergence criteria of 1×10^{-10} on flux.

The transient problem on a 1 cm mesh was evaluated for temporal convergence using 1×10^{-5}, 1×10^{-4}, and 1×10^{-3} second time steps. The sensitivity of the results to the time step size are presented in Table 3.1. The run with the finest time steps is used as the reference in this work. These results are evidence of a linear convergence rate. The times and power values reported in Table 3.1 are the global comparison metrics that will be used throughout this section to compare numerical methods against each other. A spatial comparison is also performed on the normalized assembly-averaged power at the power peak and the assembly-averaged temperatures at 3s, see section 3.3.2.

Table 3.1: Sensitivity of the transient results on time step size: comparison with a fine-time solution.

	1e-5s solution	1e-4s (% rel. diff.)	1e-3s (% rel. diff.)
Time to first peak [s]	1.44545	0.04	0.45
Power at first peak [W/cc]	5448.7	0.03	0.43
Power at second peak [W/cc]	792.80	0.05	0.16
Power at t=3s [W/cc]	98.06	0.02	0.12

Description of Point Kinetics as the Low-Order Method

In HOLO schemes, the high-order method is carried out over outer time steps and the low-order method over inner time steps (in blue and green on 3-7, respectively).

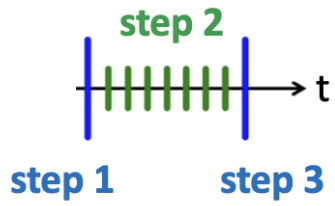

Figure 3-7: Time stepping scheme for transient HOLO methods.

It is assumed that the flux shape is slowly changing so it can be updated at the infrequent outer steps, while the amplitude is quickly changing so it is updated often on the inner steps. Decomposing flux into shape and amplitude components:

$$\Psi(r, E, t) = \psi(r, E, t)\phi(t) \tag{3.15}$$

where Ψ is the full flux, ψ is the shape, and ϕ is the amplitude. The Point Kinetics Equations (PKE) assume that the amplitude is a time dependent scalar $\phi(t)$, resulting in no spatial resolution. However, the shape, which can be updated at outer time steps, can be used to calculate PKE parameters. In this work, two outer time steps are solved, and the resulting shape functions are interpolated linearly for increased accuracy and some spatial resolution on the inner time steps. In terms of Figure 3-7, the procedure is:

- Step 1: initial shape calculation
- Step 2: amplitude calculation, assuming constant shape
- Step 3: next-step shape calculation
- Step 4: repeat Step 2: amplitude calculation, using interpolated shape

In fact, this interpolation could be repeated in a loop until a converged result is achieved, but this is not implemented in order to compare methods fairly with a constant number of outer step iterations.

There are many ways to compute PKE parameters. In this work, they are derived from the 2-group diffusion equations, simplified to assume zero leakage. The flux shape ψ is from the most recent shape calculation or interpolated between outer steps, while the amplitude ϕ is from the previous inner time step. Additionally, an adjoint flux weighting (ψ^*) is used. In the eigenvalue methods below, the adjoint is updated every time the flux shape is updated, so the adjoint used is the adjoint of the new shape, as done in Wu [85]. This differs from some implementations where only the beginning-of-life adjoint is used, which is a more arbitrary weighting factor [86, 87]. However, in the time differencing methods, the beginning-of-life adjoint weighting is used to avoid including an eigenvalue solver in addition to a time differencing solver. Thus, the reactivity insertion at each inner step is computed with interpolated flux and adjoint shape as:

$$\rho(t) = \frac{\begin{array}{c}\int dr \psi_1^*(r,t)\{-\Sigma_{r1}(r,t)\psi_1(r,t)\phi(t) + \sum_{g'=1}^{2} \nu_{g'}\Sigma_{fg'}\psi_{g'}(r,t)\phi(t)/k_{crit}\} \\ + \int dr \psi_2^*(r,t)\{-\Sigma_{a2}(r,t)\psi_2(r,t)\phi(t) + \Sigma_{s1\to 2}(r,t)\psi_2(r,t)\phi(t)\}\end{array}}{\int dr \psi_1^*(r,t) \sum_{g'=1}^{2} \nu_{g'}\Sigma_{fg'}\psi_{g'}(r,t)/k_{crit}} \tag{3.16}$$

and the prompt neutron lifetime as:

$$\Lambda(t) = \frac{\int dr \frac{1}{v_1}\psi_1(r,t)\psi_1^*(r,t) + \frac{1}{v_2}\psi_2(r,t)\psi_2^*(r,t)}{\int dr \psi_1^*(r,t) \sum_{g'=1}^{2} \nu_{g'}\Sigma_{fg'}(r,t)\psi_{g'}(r,t)/k_{crit}} \tag{3.17}$$

These terms are then inserted in the PKE:

$$\frac{d}{dt}\phi(t) = \frac{\rho(t) - \sum_i \beta_i}{\Lambda(t)}\phi(t) + \sum_i \lambda_i C_i(t) \tag{3.18}$$

$$\frac{d}{dt}C_i(t) = \frac{\beta_i}{\Lambda(t)}\phi(t) - \lambda_i C_i(t) \tag{3.19}$$

Defined in this way, the PKE are then solved using a high order Pade approximation exponential matrix function in the scipy linalg library in Python [88]. Figure 3-8 illustrates the reactivity change during the 2D LRA transient.

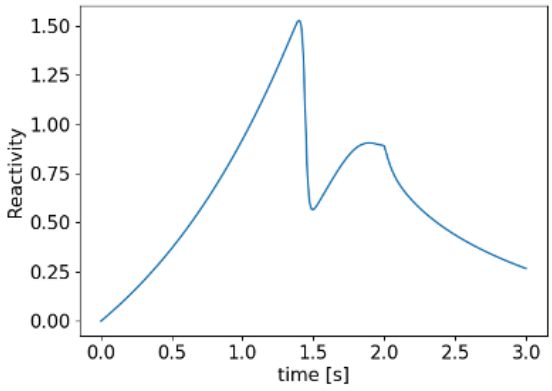

Figure 3-8: Example reactivity evolution during the 2D LRA transient on a 1 cm mesh.

With 2-group 2D diffusion as the high-order method, and PKEs as the low-order method, the list of HOLO methods used in this comparison study will now be presented. Two broad categories are distinguished: eigenvalue methods and time differencing methods. All methods are carried out on a variation of the time stepping scheme of Figure 3-7. This summary of methods focuses on their differences, mostly revolving around approximations to time derivatives in the high-order method.

Eigenvalue Methods

The following methods employ some form of eigenvalue solver to calculate the flux shape updates. The PKE are the main driver of the transient, occurring at every inner time step to update the power, calculate all feedback parameters, and compute a spatial distribution of precursor concentrations using the interpolation of the flux shape between outer time steps.

The fixed shape method is a very rudimentary method whereby the beginning-of-life flux shape is used throughout the PKE transient calculation without any updates. This method illustrates the importance of periodic shape function updates. The sole shape calculation is governed by the following k-eigenvalue equation, solved by power

iteration, with all time dependent terms at time step 0, hence the superscript. The time derivatives are absent because the equation is at steady-state.

$$-\nabla\cdot D_g^0\nabla\psi_g^0+[\Sigma_{rg}^0]\psi_g^0-\sum_{g'\neq g}\Sigma_{sg'\to g}^0\psi_{g'}^0=\frac{1}{k}\left[\frac{(1-\beta_{tot})}{k_{crit}}\chi_g^P+\sum_i\frac{\chi_g^D}{k_{crit}}\frac{\beta_i\lambda_i}{\lambda_i}\right]\sum_{g'}\nu_{g'}\Sigma_{fg'}^0\psi_{g'}^0 \tag{3.20}$$

$$\implies -\nabla\cdot D_g^0\nabla\psi_g^0+[\Sigma_{rg}^0]\psi_g^0-\sum_{g'\neq g}\Sigma_{sg'\to g}^0\psi_{g'}^0=\frac{\chi_g^{eq}}{k}\sum_{g'}\nu_{g'}\Sigma_{fg'}^0\psi_{g'}^0 \tag{3.21}$$

In the adiabatic method, the transient is assumed to be slow enough to be approximated by instantaneous eigenstates [54]. As such, the time derivatives for flux and precursor concentrations are zeroed out, much like the fixed shape method. In fact, the only difference between the fixed shape method and the adiabatic method is how often equation (3.21) is solved. In the adiabatic method, equation (3.21) changes slightly to reflect values at outer time step $n+1$. In between these shape updates, the PKE advance the transient using interpolated shape information.

$$-\nabla\cdot D_g^{n+1}\nabla\psi_g^{n+1}+[\Sigma_{rg}^{n+1}]\psi_g^{n+1}-\sum_{g'\neq g}\Sigma_{sg'\to g}^{n+1}\psi_{g'}^{n+1}=\frac{1}{k}\sum_{g'}\nu_{g'}\Sigma_{fg'}^{n+1}\psi_{g'}^{n+1} \tag{3.22}$$

In the omega method, the shape calculation includes some time dependence. The amplitude function is approximated as an exponential:

$$\Psi(r,E,t)=\psi(r,E,t)e^{\omega^P(r,E,t)\cdot t} \tag{3.23}$$

The frequency ω^P is computed as needed by the PKE when returning to the diffusion solver:

$$e^{\omega^P\cdot\Delta t}\phi_{t_{m-1}}=\phi_{t_m} \tag{3.24}$$

$$\implies \omega^P=\frac{1}{\Delta t}log\big(\frac{\phi_{t_m}}{\phi_{t_{m-1}}}\big) \tag{3.25}$$

where m is the inner time step, that is, the PKE time step. Given how small these time steps are, the frequencies are considered instantaneous and therefore their time dependence can be neglected (not because they are constant, but because they are instantaneous). With this in mind, equation (3.23) is differentiated with respect to

time:

$$\frac{\partial \Psi}{\partial t} = \frac{\partial \psi}{\partial t} e^{\omega^P \cdot t} + \psi \cdot \omega^P e^{\omega^P \cdot t} \tag{3.26}$$

If the frequencies are well chosen for accuracy, the shape function is slowly varying relative to the amplitude function. In fact, since the frequencies are computed using the last inner time step before the shape calculation, they are considered quite accurate. Therefore, the derivative of the shape is approximated to be zero [54, 89]. This leads to:

$$\frac{\partial \Psi}{\partial t} = \psi \cdot \omega^P e^{\omega^P \cdot t} \tag{3.27}$$

$$\implies \frac{1}{v}\frac{\partial \Psi}{\partial t} = \frac{1}{v}\omega^P \Psi \tag{3.28}$$

Likewise, the precursor concentrations can be frequency transformed to yield:

$$\frac{\partial C}{\partial t} = \omega^D \cdot C \tag{3.29}$$

where

$$e^{\omega^D \cdot \Delta t} C_{t_{m-1}} = C_{t_m} \tag{3.30}$$

$$\implies \omega^D = \frac{1}{\Delta t} log\big(\frac{C_{t_m}}{C_{t_{m-1}}}\big) \tag{3.31}$$

So $\omega^{P,g}$ is the prompt neutron frequency for each energy group g, and $\omega^{D,i}$ is the delayed neutron frequency for each precursor group i. Both have units of inverse time and can be phase-space dependent, limited only by the precision of the low-order method. In this work, only a single prompt frequency is employed for both energy groups because single-group PKE equations are used. The prompt frequencies are approximated as space independent which has been shown to have little impact on the accuracy of frequency-transformed problems [76]. There is, however, a lot to be gained from spatially distributed delayed frequencies. Therefore, the interpolated shape between outer time steps is used along with the amplitude change calculated by the PKE to numerically integrate the precursor distribution at each inner time step. This step allows us to estimate the precursor distribution at every moment of the transient and provide spatially distributed delayed precursors to the k-eigenvalue

solver at the shape updates:

$$\begin{aligned} &-\nabla \cdot D_g^{n+1} \nabla \psi_g^{n+1} + \left[\Sigma_{rg}^{n+1} + \frac{\omega^P(t)}{v_g} \right] \psi_g^{n+1} - \sum_{g' \neq g} \Sigma_{sg' \to g}^{n+1} \psi_{g'}^{n+1} \\ &= \frac{1}{k} \left[\frac{(1-\beta_{tot})}{k_{crit}} \chi_g^P + \sum_i \frac{\chi_g^D}{k_{crit}} \frac{\beta_i \lambda_i}{\lambda_i + \omega^{Di}(r,t)} \right] \sum_{g'} \nu_{g'} \Sigma_{fg'}^{n+1} \psi_{g'}^{n+1} \end{aligned} \tag{3.32}$$

This form of the equation shows how the prompt frequency can be added to the removal cross section, akin to a time-absorption factor as seen in α-eigenvalue literature. If all the frequencies are exact, the resulting value of k should be 1. The computed value of k and its difference from 1 are used to quantify the accuracy of the omega method implementation. As will be seen in the results, the k-balance of the omega method strays only a few pcm from 1 during the LRA transient with our selected time steps. If the frequencies are all set to 0, the method collapses to the adiabatic method.

The alpha-eigenvalue method is of a slightly different nature than the adiabatic and omega methods but retains similarities. The alpha-eigenvalue method approximates the amplitude function as an exponential so that equation (3.23) becomes:

$$\Psi(r, E, t) = \psi(r, E, t) e^{\alpha(t) \cdot t} \tag{3.33}$$

α is very similar to ω^P, having units of inverse time and acting as a time-absorption term. Instead of calculating α from the PKE and passing it to the shape update, α is calculated during the shape update. A classic alpha-eigenvalue method is described in this work, as well as several modifications that can potentially increase accuracy in reactor-relevant transients.

Note that the formulation of the classic alpha-eigenvalue neglects the role of delayed neutrons, therefore requiring prompt-only values of ν^P, χ^P in the fission term [90]. This classic alpha-eigenvalue method is implemented in this work with

$\nu^P = \nu(1-\beta)$.

$$-\nabla\cdot D_g^{n+1}\nabla\psi_g^{n+1}+\left[\Sigma_{rg}^{n+1}+\frac{\alpha(t)}{v_g}\right]\psi_g^{n+1}-\sum_{g'\neq g}\Sigma_{sg'\to g}^{n+1}\psi_{g'}^{n+1}=\frac{\chi^P}{k}\sum_{g'}\nu_{g'}\Sigma_{fg'}^{n+1}\psi_{g'}^{n+1} \quad (3.34)$$

The critical search method [91], as implemented by Ortega, can be used for the classic α-eigenvalue solver. This method has been shown to be unstable for very subcritical problems but works well otherwise. Using successive guesses for α, values for k are found. The [α,k] pairs are used to extrapolate to the true value of α such that k=1. This method can be costly due to repeated k-eigenvalue calculations (especially when the problem is close to critical) and the need to have a very tightly converged k-eigenvalue. This k-eigenvalue can be used as a balance term to determine the accuracy of the α values like in the omega method. Note, however, that the classic alpha method ignores delayed neutrons, and therefore, it cannot be considered accurate, even if the k-eigenvalue is exactly 1.

In the presence of delayed neutrons, a modification must be made to the classic alpha-eigenvalue formulation. Here, α is included as a precursor frequency and solve for α using the same algorithm as above. Remember that k is forced to be 1 as described in the critical search method. A derivation of equation (3.35) has been published by Josey [92].

$$\begin{aligned}&-\nabla\cdot D_g^{n+1}\nabla\psi_g^{n+1}+\left[\Sigma_{rg}^{n+1}+\frac{\alpha(t)}{v_g}\right]\psi_g^{n+1}-\sum_{g'\neq g}\Sigma_{sg'\to g}^{n+1}\psi_{g'}^{n+1}\\&=\frac{1}{k}\left[\frac{(1-\beta_{tot})}{k_{crit}}\chi_g^P+\sum_i\frac{\chi_g^D}{k_{crit}}\frac{\beta_i\lambda_i}{\lambda_i+\alpha(t)}\right]\sum_{g'}\nu_{g'}\Sigma_{fg'}^{n+1}\psi_{g'}^{n+1}\end{aligned} \quad (3.35)$$

In this case, the second appearance of α is a simplified version of the precursor frequencies in the omega method. It is not delayed group dependent, nor is it spatially distributed.

A novel modification of the α-eigenvalue method, proposed in this work, includes the omega method precursor frequencies into the equation. These $\omega^{D,i}$ values are spatially distributed and calculated from the PKE as input into the eigenvalue shape

calculation, just like in the omega method. Then, the α-eigenvalue calculation is performed with the critical search method. Again, k is forced to be 1.

$$\begin{aligned} &-\nabla \cdot D_g^{n+1} \nabla \psi_g^{n+1} + \left[\Sigma_{rg}^{n+1} + \frac{\alpha(t)}{v_g} \right] \psi_g^{n+1} - \sum_{g' \neq g} \Sigma_{sg' \to g}^{n+1} \psi_{g'}^{n+1} \\ &= \frac{1}{k} \left[\frac{(1-\beta_{tot})}{k_{crit}} \chi_g^P + \sum_i \frac{\chi_g^D}{k_{crit}} \frac{\beta_i \lambda_i}{\lambda_i + \omega^{Di}(r,t)} \right] \sum_{g'} \nu_{g'} \Sigma_{fg'}^{n+1} \psi_{g'}^{n+1} \end{aligned} \tag{3.36}$$

If the spatial distribution and group-wise detail of the precursor concentration change is important in a problem, this hybrid α-ω method should improve accuracy.

Time Differencing Methods

While eigenvalue methods can incorporate time dependent terms, the shape function update remains a static calculation. On the other hand, time differencing methods explicitly solve for the flux shape between outer time steps. Additionally, time differencing can allow for an analytic solution to the precursor equation. The following methods most closely resemble the implicit reference solution, with the exception that the time steps are larger, using the PKE to "fill the gaps" to save computational expense.

The coarse time integration method is achieved by adding a low-order solver to the implicit method and widening the time steps. Thus, the outer time steps drive the transient, and the role of the low-order method is to estimate the feedback effects at step n+1. Additionally, the outer time steps are repeated twice to improve the linear interpolation of the shape used by the PKE during the inner time steps.

The improved quasi-static method (IQS) decomposes the flux into shape and amplitude. The IQS performs a time differencing of both the shape and the amplitude function [93, 94], as opposed to the original quasi-static formulation which ignores the time-dependence of the shape [95, 54]. In IQS, the outer time steps drive the transient, and the low-order method estimates the feedback effects at step n+1. The IQS does not make assumptions about the amplitude function but calculates the time derivative of the amplitude and the integral of the precursors numerically. This

amplitude derivative term can be taken over the entire outer time step (an average derivative between n and n+1) or the latest inner time steps (an "instantaneous" derivative at n+1).

$$\begin{aligned} -\nabla \cdot D_g^{n+1} \nabla \psi_g^{n+1} + \left[\Sigma_{rg}^{n+1} + \frac{(\frac{\partial \phi_g}{\partial t})^{n+1}}{v_g \phi_g^{n+1}} + \frac{1}{v_g \Delta t} \right] \psi_g^{n+1} - \sum_{g' \neq g} \Sigma_{sg' \to g}^{n+1} \psi_{g'}^{n+1} \\ - \frac{(1-\beta_{tot})\chi_g^P}{k_{crit}} \sum_{g'} \nu_{g'} \Sigma_{fg'}^{n+1} \psi_{g'}^{n+1} = \frac{\psi_g^n}{v_g \Delta t} - \frac{1}{\phi_g^{n+1}} \sum_i \chi_g^D \lambda_i C_i^{n+1} \end{aligned} \tag{3.37}$$

In our implementation of the IQS, instantaneous time derivatives are used. Additionally, the value of C_i^{n+1} is computed by a numerical integration of the precursor equation over the PKE. This is a very flexible approach, but it does inhibit the use of an analytic solution to the precursors. Nonetheless, the numerical integration can be very accurate if used with small inner time steps and a sophisticated integration technique. In this work, a rectangular Riemann summing integration is employed.

The frequency transform method (FTM) is very similar to the omega method because the time derivatives of the flux are frequency transformed in an identical way: $\Psi = \psi e^{\omega t}$. The precursor equation is not frequency transformed because it can be solved directly. Assuming the flux shape changes linearly over the time step (in fact, linear shape interpolation is used in all HOLO methods in this work),

$$\frac{\partial C_i(t)}{\partial t} = -\lambda_i C_i(t) + e^{\omega t} \frac{\beta_i}{k_{crit}} \sum_{g'} \nu_{g'} \Sigma_{fg'} \psi_{g'}(t) \tag{3.38}$$

$$\psi(t) = \psi^n + \frac{t - t^n}{\Delta t} [\psi^{n+1} - \psi^n] \tag{3.39}$$

yields the analytic solution:

$$C_i^{n+1} = C_i^n e^{-\lambda_i \Delta t} + \frac{\beta_i e^{\omega \Delta t}}{k_{crit}(\lambda_i + \omega)} \sum_{g'} \nu_{g'} \Sigma_{fg'} \left[\begin{aligned} \psi_{g'}^n \left(- e^{-(\lambda_i + \omega)\Delta t} + \frac{1 - e^{-(\lambda_i + \omega)\Delta t}}{(\lambda_i + \omega)\Delta t} \right) \\ + \psi_{g'}^{n+1} \left(1 - \frac{1 - e^{-(\lambda_i + \omega)\Delta t}}{(\lambda_i + \omega)\Delta t} \right) \end{aligned} \right] \tag{3.40}$$

This form of the equation matches the SIMULATE-3K manual [51]. For full derivation

details, see Appendix B.

The diffusion equation, in an implicit differencing scheme with frequency transformed flux is:

$$\begin{aligned}\frac{1}{v_g}\frac{\psi_g^{n+1}-\psi_g^n}{\Delta t} = \nabla\cdot D_g^{n+1}\nabla\psi_g^{n+1} - \left[\Sigma_{rg}^{n+1}+\frac{\omega}{v_g}\right]\psi_g^{n+1} + \sum_{g'\neq g}\Sigma_{sg'\to g}^{n+1}\psi_{g'}^{n+1} \\ +(1-\beta_{tot})\frac{\chi_g^P}{k_{crit}}\sum_{g'}\nu_{g'}\Sigma_{fg'}^{n+1}\psi_{g'}^{n+1} \\ +e^{-\omega\Delta t}\sum_i \chi_g^D\lambda_i C_i^{n+1}\end{aligned} \tag{3.41}$$

After substituting the analytic precursor solution, simplifying, and re-arranging:

$$\begin{aligned}-\nabla\cdot D_g^{n+1}\nabla\psi_g^{n+1} + \left[\Sigma_{rg}^{n+1}+\frac{\omega}{v_g}+\frac{1}{v_g\Delta t}\right]\psi_g^{n+1} - \sum_{g'\neq g}\Sigma_{sg'\to g}^{n+1}\psi_{g'}^{n+1} \\ -\left[(1-\beta_{tot})\frac{\chi_g^P}{k_{crit}} + \sum_i \chi_g^D\lambda_i\frac{\beta_i}{k_{crit}(\lambda_i+\omega)}\left(1-\frac{1-e^{-(\lambda_i+\omega)\Delta t}}{(\lambda_i+\omega)\Delta t}\right)\right]\sum_{g'}\nu_{g'}\Sigma_{fg'}^{n+1}\psi_{g'}^{n+1} \\ = \sum_i \chi_g^D\lambda_i\frac{\beta_i}{k_{crit}(\lambda_i+\omega)}\left(-e^{-(\lambda_i+\omega)\Delta t}+\frac{1-e^{-(\lambda_i+\omega)\Delta t}}{(\lambda_i+\omega)\Delta t}\right)\sum_{g'}\nu_{g'}\Sigma_{fg'}^{n+1}\psi_{g'}^{n} \\ +\sum_i \chi_g^D\lambda_i C_i^{n+1}e^{-(\lambda_i+\omega)\Delta t} + \frac{\psi_g^n}{v_g\Delta t}\end{aligned} \tag{3.42}$$

To obtain equation (3.11) used in the implicit method, set $\omega = 0$. A similar form to equation (3.14) can be obtained here by defining modified cross sections appropriately, so the solver for the two methods is identical.

$$\begin{cases}-\nabla\cdot D_1^{n+1}\nabla\Psi_1^{n+1} + \widehat{\Sigma_{r1}^{n+1}}\Psi_1^{n+1} - \sum_{g'}\widehat{\Sigma_{fg'}^{n+1}}\Psi_{g'}^{n+1} = S_1 \\ -\nabla\cdot D_2^{n+1}\nabla\Psi_2^{n+1} + \widehat{\Sigma_{r2}^{n+1}}\Psi_2^{n+1} - \Sigma_{s1\to 2}^{n+1}\Psi_1^{n+1} = S_2\end{cases} \tag{3.43}$$

When $\omega = 0$, the method collapses to a coarse time integration method. The frequency transform method is a more specific form of the IQS. In equation (3.37), if $\phi = e^{\omega t}$, $\frac{\partial\phi/\partial t}{v\phi} = \frac{\omega}{v}$ the equation collapses to the frequency transform method. The frequency calculation can be taken over the entire outer time step (an average frequency

between n and n+1) or the latest inner time steps (an "instantaneous" frequency at n+1). The instantaneous frequencies are used in this work.

Figure 3-9 attempts to encapsulate the differences and similarities between methods in a single illustration.

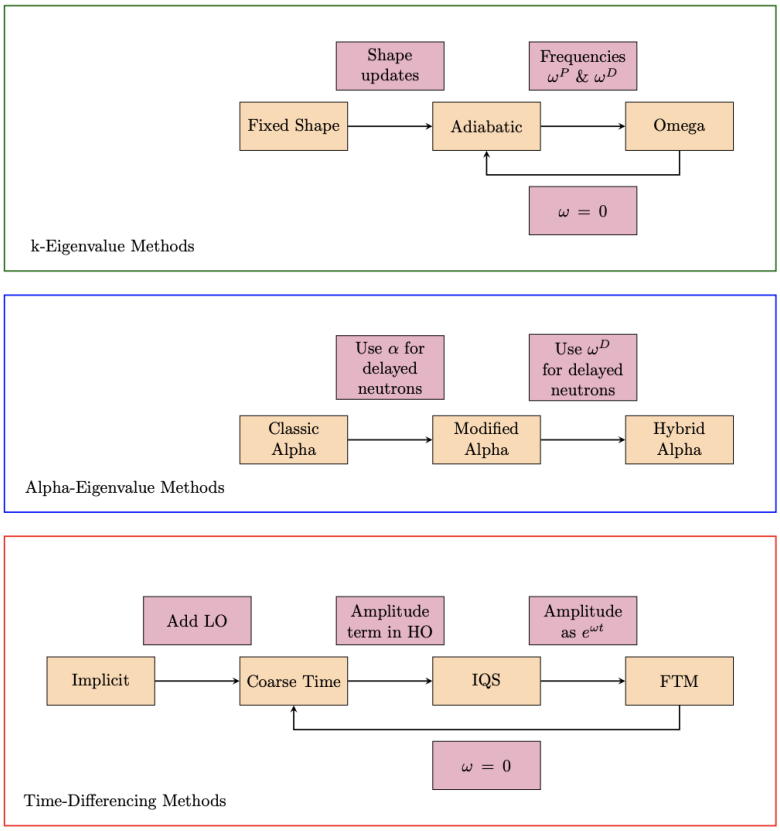

Figure 3-9: Summary of HOLO methods and their relationships.

3.3.2 Results

Results obtained with the methods described are now compared. The error of the eigenvalue methods is expected to be worse than the time differencing methods which is supported by the results. Different eigenvalue methods are compared against each other and likewise for the time differencing methods. The reference solution in all cases is the fully implicit method with 1×10^{-5}s time steps and no low-order method. PKEs use 1×10^{-4}s inner time steps which showed no difference with 1×10^{-5}s or 1×10^{-6}s. A variety of outer time step sizes are tested to observe the convergence

behavior of each method.

All methods use the same 1 cm mesh and the same convergence criteria of 1×10^{-10}s for calculating flux. All methods employ a linear shape interpolation whereby the outer time steps are calculated twice to improve the interpolation of point kinetics parameters. Thanks to this, a full spatial precursor distribution can be computed at every inner time step, regardless of the method. This implies that when a C^{n+1} term is used, it includes full spatial resolution.

Eigenvalue Methods

The fixed shape method is quickly determined to be inaccurate by comparing the transient's power profile in Figure 3-10, where all curves are nearly overlapping except for the fixed shape method. This stresses the importance of flux shape updates. The error of the adiabatic and omega methods is more subtle. Figure 3-11 provides a close-up of the peak to visualize the differences. The values of k-balance in Figure 3-12 are usedto determine the impact of frequencies in the omega method. If the frequencies are accurate, k-balance should be exactly 1. For the adiabatic method on the other hand, the change in k is expected to follow the reactivity change of the problem. Figure 3-12 confirms both, and Figure 3-13 shows the deviation of k-balance from 1 in pcm. Figures 3-14 and 3-15 show the evolution of the prompt and delayed frequencies throughout the transient. Figure 3-16 shows the spatial resolution of the delayed frequencies as the transient evolves. All these behave as expected.

The results in Tables 3.2 and 3.3 highlight that the omega method is better than the adiabatic method for predicting the transient peak. Errors fluctuate, sometimes improving, sometimes degrading with refined time steps. This suggests that a convergence plateau has been reached, likely at 0.1s. The eigenvalue methods introduce an underlying error by assuming that a static update of the shape can characterize the transient. This inherently flawed approach prohibits the solution from converging to the correct answer, even at fine time steps. Results in Figures 3-10 through 3-16 are shown for outer time steps of 0.1s.

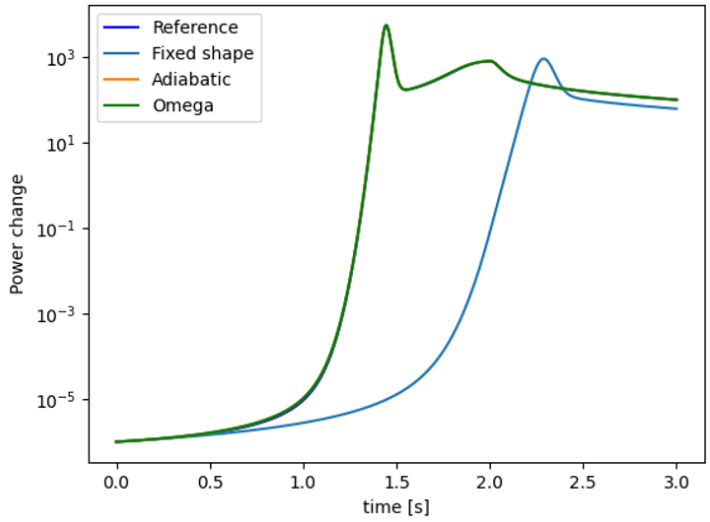

Figure 3-10: Power profile shows that the fixed shape method is inadequate.

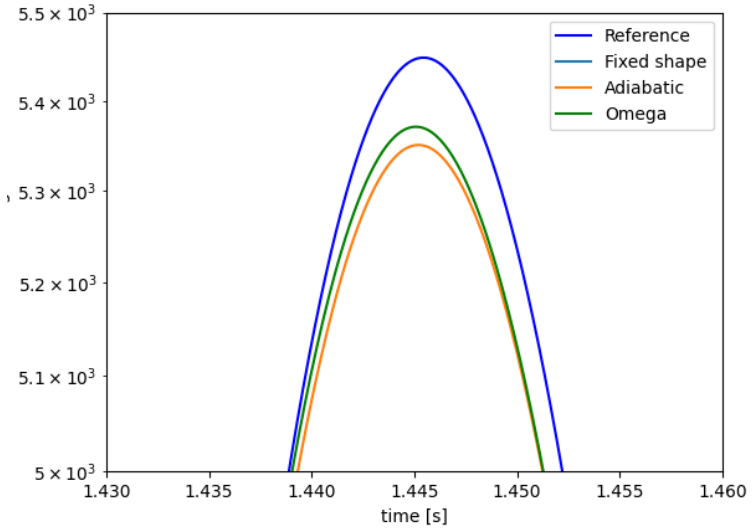

Figure 3-11: Zoom-in to the first peak (Fixed shape method is not visible in this view).

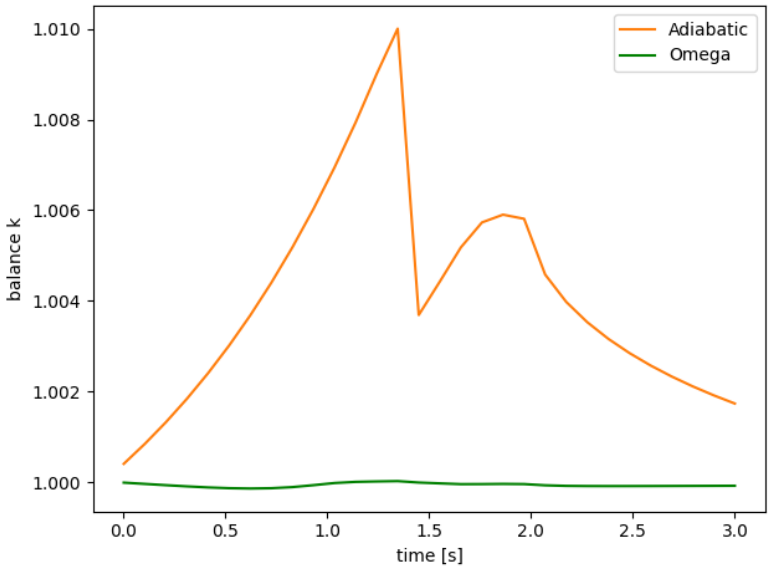

Figure 3-12: k-balance values over time.

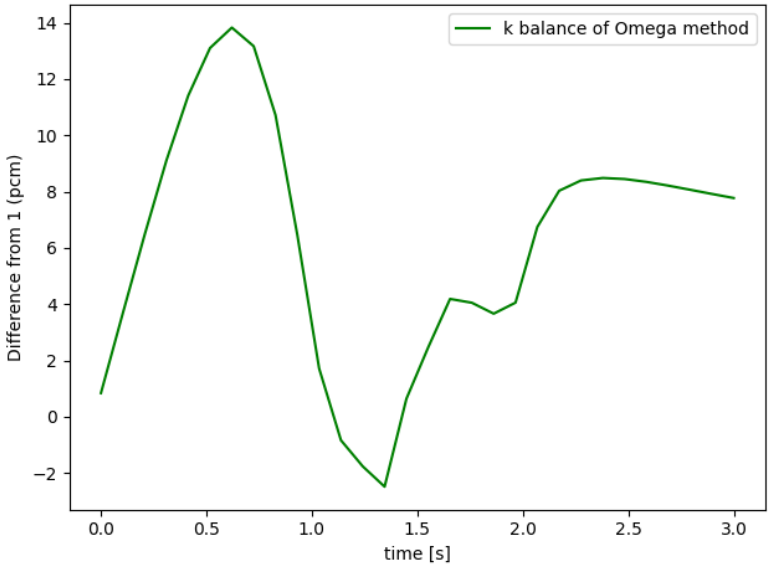

Figure 3-13: k-balance deviation from 1 in pcm over time.

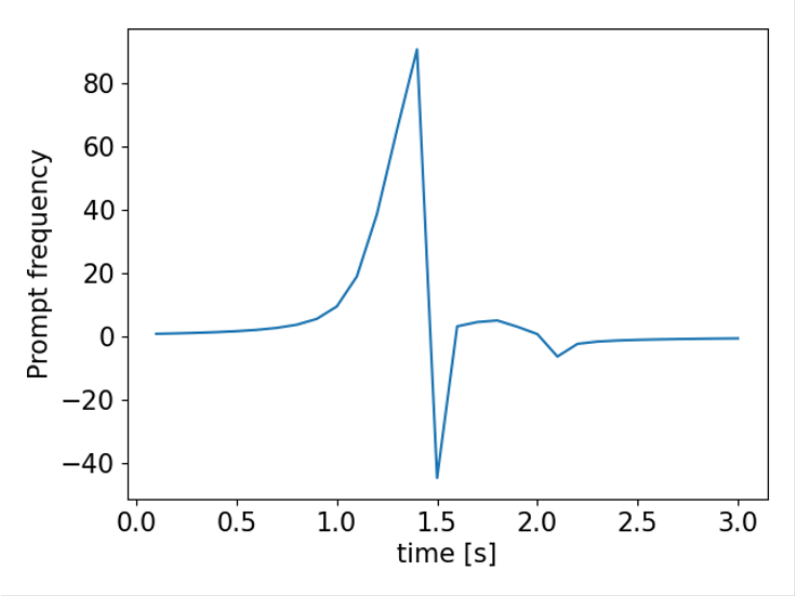

Figure 3-14: Prompt frequencies $[s^{-1}]$ as calculated by the omega method.

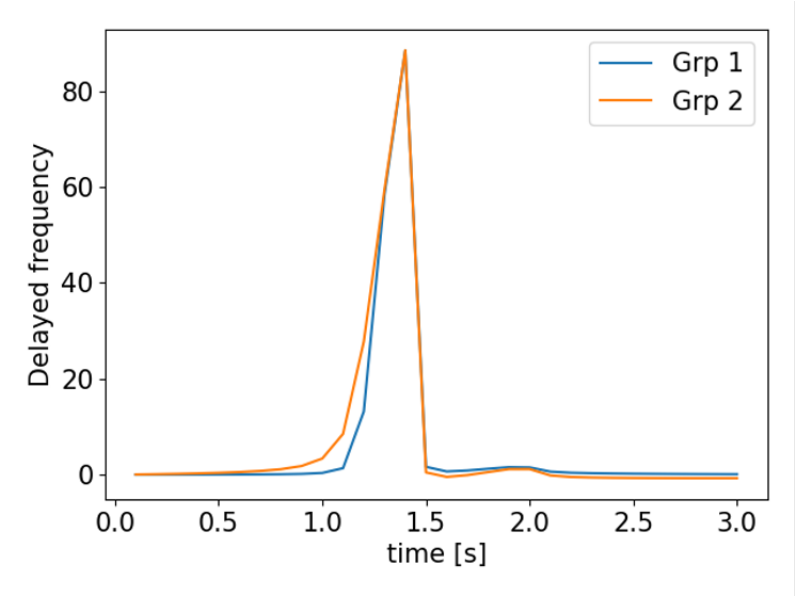

Figure 3-15: Precursor frequencies $[s^{-1}]$ for each delayed neutron group in the first cell of the geometry as calculated by the omega method.

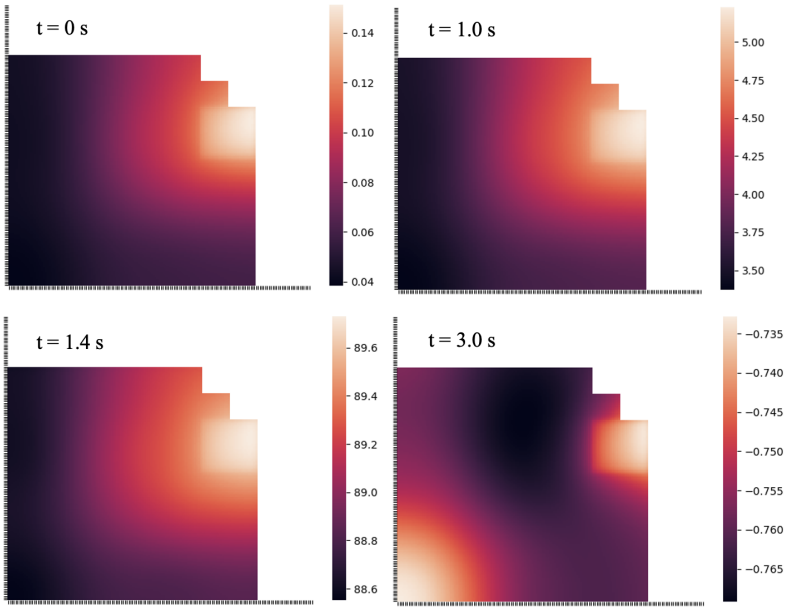

Figure 3-16: Spatial distribution of delayed frequencies $[s^{-1}]$ for precursor group 2 at different points in time, as calculated by the omega method.

Table 3.2: Sensitivity of the adiabatic method to outer time step size.

	Reference	0.01s outer steps solution	0.01s	0.05s	0.1s	0.5s
			% rel. dif. from reference			
Time to first peak [s]	1.44545	1.4453	0.01	0.00	0.02	0.13
Power at first peak [W/cc]	5448.7	5338.6	2.02	1.37	1.80	0.41
Power at second peak [W/cc]	792.80	790.4	0.30	0.26	0.19	1.12
Power at t=3s [W/cc]	98.06	98.79	0.74	0.75	0.69	0.58

Table 3.3: Sensitivity of the omega method to outer time step size.

	Reference	0.01s outer steps solution	0.01s	0.05s	0.1s	0.5s
			% rel. dif. from reference			
Time to first peak [s]	1.44545	1.4452	0.02	0.02	0.02	0.16
Power at first peak [W/cc]	5448.7	5364.44	1.55	1.11	1.43	0.01
Power at second peak [W/cc]	792.80	792.2	0.08	0.06	0.04	1.01
Power at t=3s [W/cc]	98.06	98.90	0.86	0.87	0.84	0.61

The omega method is expected to have better spatial resolution because of the spatially dependent precursor frequencies. Figure 3-17 compares the shape of the assembly powers at the power peak. The omega method is confirmed to outperform the adiabatic method in this respect, with a 22% lower power error root mean square for omega. Figure 3-18 compares the end-of-transient temperatures which illustrates the cumulative effect of the power change during the transient. The region of the control rod movement has lower error with the omega method, but that is not true of the rest of the reactor. This suggests that the exponential frequencies work best in the extreme parts of the transient that can be characterized by an exponential change.

0.57	**0.57**	**0.61**	**0.74**	**0.92**	**1.08**	**1.05**		
-1.11	-0.95	-0.63	-0.24	0.12	0.38	0.54		
-1.02	-0.87	-0.59	-0.26	0.03	0.22	0.29		
0.91	**0.84**	**0.87**	**1.07**	**1.45**	**1.94**	**2.13**	**2.04**	
-1.11	-0.95	-0.61	-0.18	0.23	0.51	0.68	0.74	
-0.97	-0.81	-0.50	-0.13	0.18	0.39	0.49	0.43	
1.02	**0.75**	**0.72**	**0.91**	**1.40**	**2.45**	**3.01**	**2.97**	**1.79**
-1.13	-0.98	-0.61	-0.13	0.31	0.59	0.75	0.80	0.75
-0.99	-0.79	-0.43	-0.03	0.29	0.44	0.55	0.58	0.45
0.85	**0.61**	**0.59**	**0.76**	**1.22**	**2.22**	**2.86**	**3.04**	**1.97**
-1.16	-1.01	-0.64	-0.16	0.28	0.56	0.72	0.76	0.70
-0.99	-0.78	-0.41	-0.01	0.30	0.44	0.55	0.58	0.48
0.49	**0.44**	**0.46**	**0.60**	**0.88**	**1.32**	**1.73**	**2.05**	**1.42**
-1.20	-1.06	-0.73	-0.30	0.09	0.37	0.53	0.57	0.52
-0.92	-0.75	-0.44	-0.08	0.20	0.39	0.49	0.48	0.37
0.32	**0.32**	**0.36**	**0.46**	**0.64**	**0.87**	**1.13**	**1.42**	**1.02**
-1.28	-1.15	-0.87	-0.53	-0.22	0.01	0.15	0.19	0.17
-0.90	-0.76	-0.52	-0.24	-0.01	0.15	0.22	0.19	0.10
0.25	**0.26**	**0.30**	**0.38**	**0.52**	**0.69**	**0.89**	**1.12**	**0.80**
-1.38	-1.27	-1.06	-0.81	-0.60	-0.43	-0.32	-0.27	-0.26
-0.93	-0.82	-0.65	-0.46	-0.32	-0.22	-0.16	-0.19	-0.26
0.26	**0.25**	**0.27**	**0.35**	**0.51**	**0.73**	**0.91**	**1.04**	**0.72**
-1.47	-1.38	-1.22	-1.04	-0.90	-0.79	-0.71	-0.65	-0.64
-1.01	-0.92	-0.79	-0.67	-0.60	-0.57	-0.53	-0.53	-0.58
0.36	**0.27**	**0.27**	**0.36**	**0.57**	**1.02**	**1.24**	**1.12**	**0.71**
-1.50	-1.45	-1.31	-1.15	-1.02	-0.93	-0.87	-0.84	-0.83
-1.11	-1.00	-0.87	-0.78	-0.76	-0.78	-0.76	-0.72	-0.76

ref. value
% rel. err. (adiabatic)
% rel. err. (omega)

Root mean square error	
Adibatic:	0.79
Omega:	0.57

Figure 3-17: Normalized assembly power densities at the peak for adiabatic versus omega. In green and red, errors that are lower and higher for omega method, respectively.

712	**715**	**754**	**854**	**1010**	**1146**	**1136**		
-0.10	-0.05	0.05	0.17	0.31	0.41	0.47		
0.13	0.16	0.21	0.29	0.37	0.43	0.44		
958	**911**	**947**	**1107**	**1412**	**1820**	**2022**	**2009**	
-0.12	-0.06	0.06	0.21	0.37	0.51	0.60	0.65	
0.17	0.20	0.26	0.35	0.45	0.52	0.56	0.56	
1035	**844**	**836**	**989**	**1383**	**2234**	**2757**	**2886**	**1920**
-0.13	-0.06	0.06	0.22	0.40	0.56	0.66	0.73	0.71
0.17	0.20	0.26	0.36	0.47	0.56	0.61	0.64	0.59
916	**746**	**737**	**875**	**1242**	**2061**	**2643**	**2946**	**2068**
-0.12	-0.06	0.05	0.21	0.39	0.55	0.65	0.72	0.70
0.17	0.19	0.25	0.35	0.46	0.55	0.61	0.65	0.61
655	**623**	**646**	**754**	**981**	**1342**	**1695**	**2004**	**1505**
-0.10	-0.05	0.04	0.17	0.32	0.45	0.55	0.61	0.59
0.15	0.17	0.22	0.31	0.41	0.50	0.56	0.58	0.53
533	**538**	**570**	**650**	**789**	**975**	**1196**	**1442**	**1126**
-0.08	-0.05	0.02	0.11	0.21	0.32	0.40	0.45	0.43
0.13	0.15	0.19	0.26	0.33	0.41	0.46	0.48	0.43
487	**494**	**523**	**587**	**692**	**829**	**988**	**1167**	**924**
-0.08	-0.05	-0.01	0.05	0.12	0.18	0.24	0.28	0.26
0.11	0.13	0.16	0.21	0.26	0.31	0.35	0.37	0.33
495	**485**	**504**	**565**	**679**	**845**	**986**	**1088**	**843**
-0.08	-0.06	-0.03	0.00	0.04	0.08	0.11	0.14	0.14
0.11	0.11	0.14	0.17	0.22	0.26	0.28	0.29	0.25
568	**499**	**501**	**565**	**723**	**1054**	**1218**	**1133**	**827**
-0.11	-0.07	-0.05	-0.02	0.01	0.04	0.06	0.08	0.07
0.11	0.11	0.13	0.16	0.20	0.24	0.26	0.26	0.22

ref. value
% rel. err. (adiabatic)
% rel. err. (omega)

Root mean square error	
Adibatic:	0.37
Omega:	0.39

Figure 3-18: Assembly temperatures at the end-of-transient for adiabatic versus omega. In green and red, errors that are lower and higher for omega method, respectively.

Next, alpha eigenvalue methods are compared to the omega method. The classic alpha method is implemented for illustration: although the start of the transient is well simulated in the absence of delayed neutron modeling, the peak is significantly different from the omega method and the other alpha-eigenvalue methods which overlap significantly as seen in Figure 3-19. The fact that it is very close to the reference solution is likely a coincidence. Table 3.4 emphasizes that although the power at the first peak happens to be very accurate, the power at the second peak and at 3s is significantly worse for the classic alpha eigenvalue.

The modified alpha method is a simple way to include delayed neutrons into the classic alpha-eigenvalue method. The hybrid alpha method is more sophisticated and includes spatial resolution. Figures 3-20 and 3-21 show that hybrid alpha is

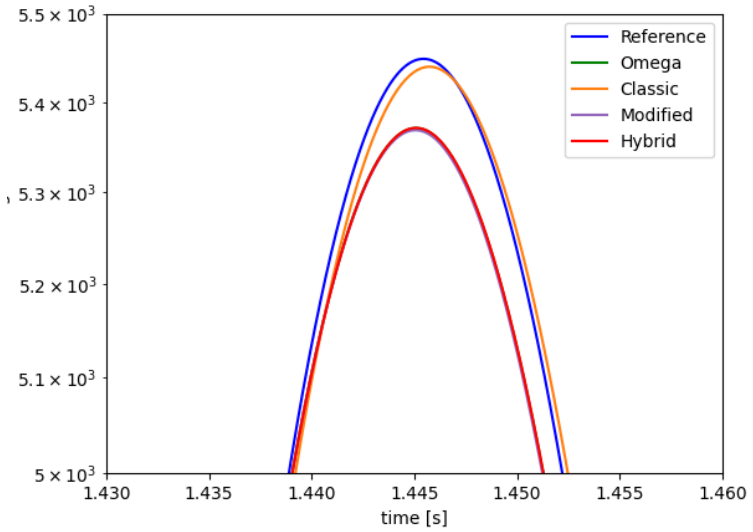

Figure 3-19: Power profile, zoomed-in.

Table 3.4: Alpha eigenvalue results, obtained with 0.1s outer time steps.

	Reference	Classic	Modified	Hybrid
		% rel. dif. from reference		
Time to first peak [s]	1.44545	0.02	0.02	0.02
Power at first peak [W/cc]	5448.7	0.16	1.47	1.42
Power at second peak [W/cc]	792.80	0.98	0.29	0.05
Power at t=3s [W/cc]	98.06	1.78	0.50	0.84

an improvement over modified alpha. Hybrid alpha even appears to outperform the omega method. The only difference between the omega and hybrid alpha methods is the prompt frequency. The omega method calculates a prompt frequency from the inner steps, whereas hybrid alpha uses the critical search algorithm to determine alpha. This suggests that the alpha-eigenvalue search method is more robust than the prompt frequency calculation, although it is significantly more expensive because of the iterative methodology used to evaluate alpha. In Figure 3-22, the power distribution at the peak clearly shows that the hybrid alpha method has better spatial resolution than the modified alpha method, as expected. The power error root mean square is 17% lower for hybrid. In Figure 3-23, the end-of-transient temperatures show a similar trend to adiabatic and omega methods, but less pronounced. In other words, the hybrid alpha method is a larger improvement on the modified alpha than

omega is on adiabatic.

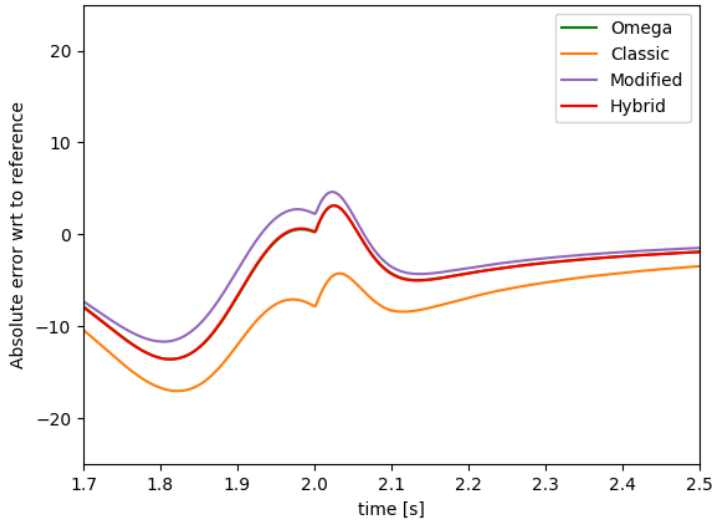

Figure 3-20: Absolute error on power with respect to reference, in time, zoomed to the second peak.

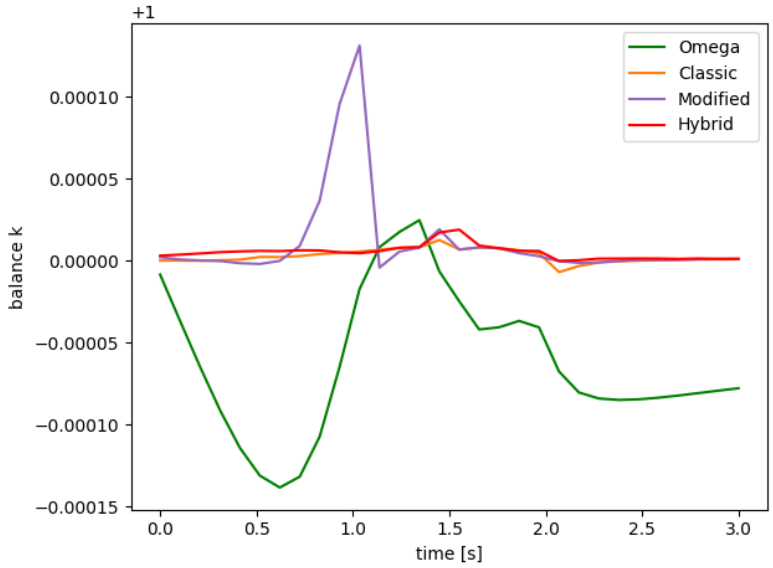

Figure 3-21: k-balance values over time.

0.57	**0.57**	**0.61**	**0.74**	**0.92**	**1.08**	**1.05**		
-1.12	-0.95	-0.63	-0.25	0.09	0.32	0.42		
-1.00	-0.86	-0.58	-0.26	0.02	0.21	0.27		
0.91	**0.84**	**0.87**	**1.07**	**1.45**	**1.94**	**2.13**	**2.04**	
-1.10	-0.93	-0.57	-0.15	0.22	0.47	0.60	0.60	
-0.95	-0.80	-0.49	-0.13	0.18	0.37	0.47	0.41	
1.02	**0.75**	**0.72**	**0.91**	**1.40**	**2.45**	**3.01**	**2.97**	**1.79**
-1.12	-0.93	-0.53	-0.07	0.32	0.53	0.67	0.71	0.62
-0.97	-0.77	-0.41	-0.02	0.29	0.43	0.53	0.56	0.43
0.85	**0.61**	**0.59**	**0.76**	**1.22**	**2.22**	**2.86**	**3.04**	**1.97**
-1.13	-0.94	-0.54	-0.07	0.31	0.52	0.65	0.70	0.62
-0.96	-0.75	-0.39	0.00	0.30	0.43	0.54	0.57	0.46
0.49	**0.44**	**0.46**	**0.60**	**0.88**	**1.32**	**1.73**	**2.05**	**1.42**
-1.13	-0.95	-0.60	-0.18	0.17	0.41	0.54	0.55	0.47
-0.89	-0.72	-0.41	-0.07	0.21	0.38	0.48	0.46	0.35
0.32	**0.32**	**0.36**	**0.46**	**0.64**	**0.87**	**1.13**	**1.42**	**1.02**
-1.16	-1.01	-0.72	-0.39	-0.10	0.10	0.21	0.21	0.15
-0.85	-0.72	-0.48	-0.22	0.00	0.15	0.22	0.19	0.09
0.25	**0.26**	**0.30**	**0.38**	**0.52**	**0.69**	**0.89**	**1.12**	**0.80**
-1.25	-1.13	-0.90	-0.66	-0.46	-0.32	-0.23	-0.22	-0.26
-0.88	-0.78	-0.61	-0.43	-0.29	-0.20	-0.15	-0.18	-0.26
0.26	**0.25**	**0.27**	**0.35**	**0.51**	**0.73**	**0.91**	**1.04**	**0.72**
-1.37	-1.26	-1.07	-0.90	-0.77	-0.70	-0.63	-0.60	-0.62
-0.96	-0.87	-0.74	-0.63	-0.57	-0.55	-0.52	-0.52	-0.58
0.36	**0.27**	**0.27**	**0.36**	**0.57**	**1.02**	**1.24**	**1.12**	**0.71**
-1.44	-1.34	-1.17	-1.02	-0.92	-0.89	-0.84	-0.80	-0.82
-1.06	-0.94	-0.82	-0.75	-0.73	-0.77	-0.74	-0.70	-0.75

ref. value
% rel. err. (modified)
% rel. err. (hybrid)

Root mean square error	
Modified:	0.72
Hybrid:	0.55

Figure 3-22: Normalized assembly power densities at the peak for modified versus hybrid alpha. In green and red, errors that are lower or higher for hybrid, respectively.

712	**715**	**754**	**854**	**1010**	**1146**	**1136**		
-0.35	-0.26	-0.10	0.11	0.32	0.50	0.65		
0.14	0.16	0.21	0.29	0.37	0.43	0.45		
958	**911**	**947**	**1107**	**1412**	**1820**	**2022**	**2009**	
-0.45	-0.35	-0.15	0.10	0.36	0.59	0.76	0.93	
0.17	0.20	0.26	0.36	0.45	0.53	0.57	0.57	
1035	**844**	**836**	**989**	**1383**	**2234**	**2757**	**2886**	**1920**
-0.47	-0.38	-0.18	0.08	0.38	0.66	0.84	0.96	1.02
0.18	0.20	0.26	0.36	0.48	0.57	0.62	0.65	0.60
916	**746**	**737**	**875**	**1242**	**2061**	**2643**	**2946**	**2068**
-0.48	-0.39	-0.21	0.04	0.34	0.62	0.79	0.91	0.95
0.17	0.19	0.25	0.35	0.47	0.56	0.62	0.66	0.62
655	**623**	**646**	**754**	**981**	**1342**	**1695**	**2004**	**1505**
-0.46	-0.38	-0.24	-0.04	0.20	0.42	0.58	0.70	0.73
0.15	0.17	0.23	0.31	0.41	0.50	0.56	0.59	0.54
533	**538**	**570**	**650**	**789**	**975**	**1196**	**1442**	**1126**
-0.44	-0.38	-0.28	-0.14	0.02	0.18	0.31	0.43	0.46
0.13	0.15	0.19	0.26	0.34	0.41	0.46	0.49	0.44
487	**494**	**523**	**587**	**692**	**829**	**988**	**1167**	**924**
-0.44	-0.40	-0.33	-0.24	-0.14	-0.05	0.05	0.15	0.19
0.11	0.13	0.16	0.21	0.27	0.32	0.36	0.38	0.33
495	**485**	**504**	**565**	**679**	**845**	**986**	**1088**	**843**
-0.49	-0.44	-0.38	-0.32	-0.26	-0.20	-0.14	-0.07	-0.01
0.11	0.11	0.14	0.18	0.22	0.26	0.29	0.30	0.25
568	**499**	**501**	**565**	**723**	**1054**	**1218**	**1133**	**827**
-0.58	-0.48	-0.41	-0.36	-0.31	-0.25	-0.20	-0.17	-0.11
0.11	0.11	0.13	0.16	0.20	0.24	0.26	0.26	0.22

ref. value
% rel. err. (modified)
% rel. err. (hybrid)

Root mean square error	
Modified:	0.49
Hybrid:	0.40

Figure 3-23: Assembly temperatures at the end-of-transient for modified versus hybrid alpha. In green and red, errors that are lower or higher for hybrid, respectively.

Time Differencing Methods

The coarse time integration, the improved quasi-static (IQS), and the frequency transform method (FTM) are now compared against each other. Finer time steps are required for stability as discussed earlier. Table 3.5 assesses the temporal convergence of the coarse time differencing method. It does not converge linearly the way the implicit method does. This is due to the effect of PKE and linear shape interpolation between outer time steps. To prove this, Table 3.6 shows a temporal convergence analysis of a "stripped down" version of the coarse time integration. This "strip down" includes removing linear interpolation and using lagging Doppler feedback, as opposed to the PKE-calculated values. By removing these elements, the PKE contributions are deleted, and the method retains only the time-differencing between outer steps. As seen in Table 3.6, the method now approaches linear convergence. Therefore, the addition of a low-order method impacts the temporal convergence properties of the high-order method but provides greater accuracy as the outer time steps are lengthened. Notice that the error of the stripped-down method at 1e-3s time steps is similar to the implicit method. The addition of the low-order method allows the error to be reduced as the time steps get larger. Results in Table 3.7 and the figures in this section are shown for outer time steps of 0.01s.

Table 3.5: Sensitivity of the coarse time integration method to outer time step size.

	Reference	0.001s outer steps solution	0.001s	0.005s	0.01s
			% rel. dif. from reference		
Time to first peak [s]	1.44545	1.4457	0.02	0.01	0.02
Power at first peak [W/cc]	5448.7	5389.9	1.08	1.20	1.55
Power at second peak [W/cc]	792.80	791.2	0.20	0.21	0.24
Power at t=3s [W/cc]	98.06	98.27	0.21	0.21	0.21

Table 3.6: Sensitivity of the "stripped" coarse time integration method to outer time step size.

	Reference	0.001s outer steps solution	0.001s	0.005s	0.01s
			% rel. dif. from reference		
Time to first peak [s]	1.44545	1.440	0.38	2.11	4.53
Power at first peak [W/cc]	5448.7	5425.1	0.43	2.03	3.56
Power at second peak [W/cc]	792.80	794.0	0.15	0.45	0.10
Power at t=3s [W/cc]	98.06	98.01	0.05	0.25	0.50

Table 3.7: Time differencing results, obtained with 0.01s outer time steps.

	Reference	Coarse	IQS	FTM
		% rel. dif. from reference		
Time to first peak [s]	1.44545	0.02	0.14	0.09
Power at first peak [W/cc]	5448.7	1.55	1.58	0.94
Power at second peak [W/cc]	792.80	0.24	0.16	0.32
Power at t=3s [W/cc]	98.06	0.21	0.10	0.13

The FTM employs frequencies, and the IQS employs amplitude derivatives, but both serve a similar purpose to approximate the amplitude change. These amplitude change terms were computed as instantaneous values based on the two most recent inner time steps. Figure 3-24 emphasizes the similarity of these terms, as their values overlap throughout the transient. It is conjectured therefore that the differences between IQS and FTM results are mainly due to the formulation of the precursor equations which is the major difference between the two methods. While IQS performs a numerical precursor integration, the FTM solves for the precursors analytically, as detailed in section 3.3.1. From Figure 3-25 and Table 3.7, the FTM predicts the time and power of the peak better, but the IQS performs better later in the transient.

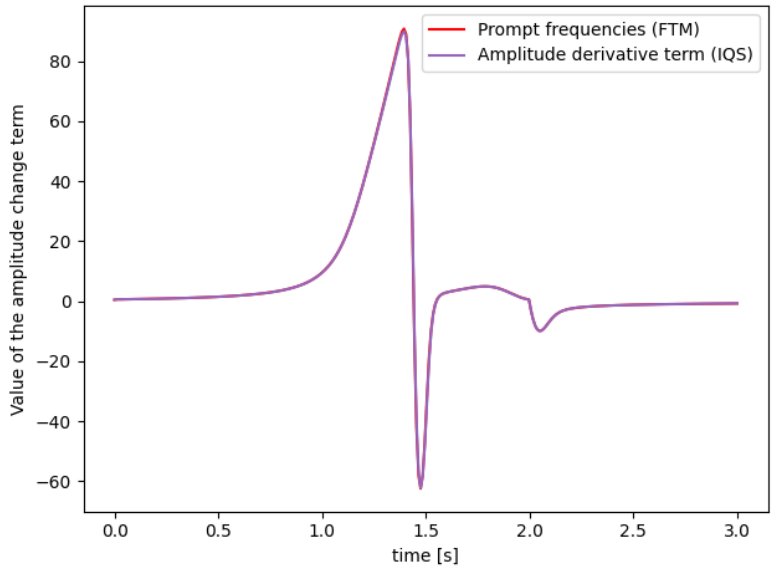

Figure 3-24: Comparing prompt frequencies in the FTM to the amplitude derivative term in the IQS.

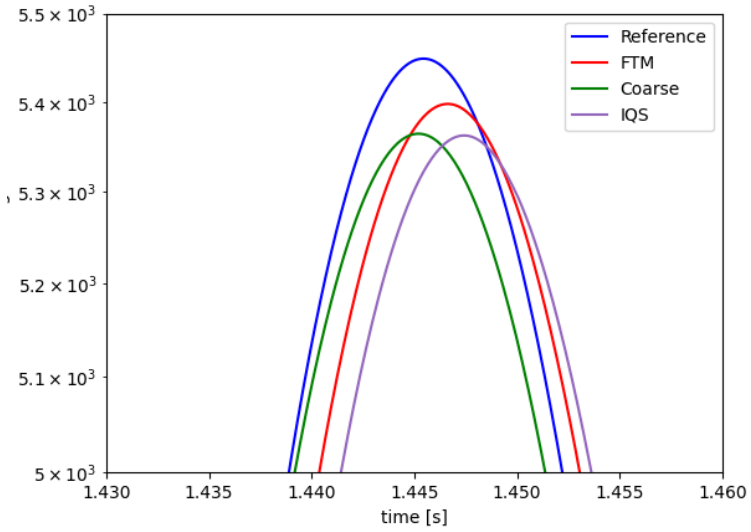

Figure 3-25: Zoomed-in power profile.

Next, the spatial error is analyzed. FTM has a more accurate power distribution at the peak as shown in Figure 3-26 with a root mean square error of 27% lower for FTM. For end of transient temperatures, the methods are more similar as shown in

Figure 3-27. In general, IQS and FTM appear to have a lot in common. The precursor concentration calculation, which is the main difference between the two methods, carries much of the spatial distribution of the problem. Therefore, our conjecture about the use of a numerical precursor integration versus an analytic solution being the primary difference between the two methods is reinforced.

0.57	**0.57**	**0.61**	**0.74**	**0.92**	**1.08**	**1.05**		
0.28	0.22	0.10	-0.04	-0.17	-0.26	-0.35		
0.13	0.10	0.04	-0.02	-0.06	-0.09	-0.11		
0.91	**0.84**	**0.87**	**1.07**	**1.45**	**1.94**	**2.13**	**2.04**	
0.32	0.28	0.17	0.01	-0.13	-0.24	-0.30	-0.38	
0.14	0.11	0.06	-0.01	-0.06	-0.08	-0.09	-0.11	
1.02	**0.75**	**0.72**	**0.91**	**1.40**	**2.45**	**3.01**	**2.97**	**1.79**
0.33	0.35	0.24	0.07	-0.11	-0.25	-0.30	-0.28	-0.33
0.15	0.13	0.07	0.00	-0.05	-0.09	-0.09	-0.07	-0.08
0.85	**0.61**	**0.59**	**0.76**	**1.22**	**2.22**	**2.86**	**3.04**	**1.97**
0.38	0.41	0.31	0.13	-0.06	-0.21	-0.25	-0.23	-0.26
0.16	0.15	0.09	0.02	-0.04	-0.07	-0.08	-0.06	-0.06
0.49	**0.44**	**0.46**	**0.60**	**0.88**	**1.32**	**1.73**	**2.05**	**1.42**
0.52	0.51	0.40	0.22	0.05	-0.07	-0.13	-0.16	-0.21
0.19	0.17	0.11	0.04	-0.01	-0.04	-0.05	-0.05	-0.06
0.32	**0.32**	**0.36**	**0.46**	**0.64**	**0.87**	**1.13**	**1.42**	**1.02**
0.68	0.63	0.51	0.35	0.20	0.09	0.01	-0.06	-0.12
0.23	0.20	0.15	0.08	0.03	0.00	-0.01	-0.03	-0.04
0.25	**0.26**	**0.30**	**0.38**	**0.52**	**0.69**	**0.89**	**1.12**	**0.80**
0.83	0.76	0.63	0.47	0.34	0.23	0.15	0.07	0.00
0.28	0.24	0.19	0.13	0.08	0.05	0.03	0.01	-0.01
0.26	**0.25**	**0.27**	**0.35**	**0.51**	**0.73**	**0.91**	**1.04**	**0.72**
0.91	0.86	0.72	0.56	0.41	0.30	0.23	0.16	0.10
0.32	0.29	0.22	0.16	0.11	0.08	0.06	0.04	0.03
0.36	**0.27**	**0.27**	**0.36**	**0.57**	**1.02**	**1.24**	**1.12**	**0.71**
0.89	0.89	0.77	0.59	0.41	0.26	0.20	0.19	0.14
0.33	0.31	0.24	0.17	0.12	0.08	0.06	0.06	0.05

ref. value
% rel. err. (IQS)
% rel. err. (FTM)

Root mean square error	
IQS:	0.39
FTM:	0.12

Figure 3-26: Normalized assembly power densities at the peak for IQS versus FTM. In green and red, errors that are lower and higher for FTM, respectively.

HOLO Conclusions

In the category of k-eigenvalue methods, the fixed shape method illustrates the strong need for flux shape calculations throughout transient simulations. The error associated with the adiabatic method and the omega method show that the omega method is more accurate thanks to the presence of a prompt frequency and spatially distributed precursor frequencies, increasing the spatial resolution of the transient through the frequency transform which assumes exponential behavior.

712	**715**	**754**	**854**	**1010**	**1146**	**1136**		
0.15	0.14	0.13	0.12	0.11	0.10	0.08		
0.11	0.12	0.14	0.16	0.19	0.22	0.23		
958	**911**	**947**	**1107**	**1412**	**1820**	**2022**	**2009**	
0.18	0.17	0.16	0.14	0.13	0.12	0.11	0.10	
0.13	0.14	0.16	0.19	0.22	0.25	0.27	0.29	
1035	**844**	**836**	**989**	**1383**	**2234**	**2757**	**2886**	**1920**
0.19	0.18	0.16	0.15	0.13	0.12	0.12	0.14	0.13
0.13	0.13	0.15	0.19	0.23	0.26	0.29	0.32	0.32
916	**746**	**737**	**875**	**1242**	**2061**	**2643**	**2946**	**2068**
0.19	0.18	0.17	0.15	0.14	0.13	0.13	0.15	0.14
0.13	0.13	0.14	0.18	0.22	0.26	0.29	0.32	0.32
655	**623**	**646**	**754**	**981**	**1342**	**1695**	**2004**	**1505**
0.18	0.17	0.16	0.16	0.15	0.15	0.15	0.15	0.13
0.11	0.12	0.13	0.17	0.21	0.24	0.27	0.29	0.28
533	**538**	**570**	**650**	**789**	**975**	**1196**	**1442**	**1126**
0.17	0.16	0.16	0.16	0.16	0.16	0.16	0.15	0.13
0.10	0.10	0.12	0.15	0.18	0.21	0.24	0.26	0.24
487	**494**	**523**	**587**	**692**	**829**	**988**	**1167**	**924**
0.16	0.16	0.16	0.16	0.17	0.17	0.17	0.16	0.13
0.09	0.10	0.11	0.13	0.16	0.18	0.20	0.22	0.20
495	**485**	**504**	**565**	**679**	**845**	**986**	**1088**	**843**
0.18	0.17	0.16	0.17	0.17	0.18	0.18	0.17	0.14
0.09	0.09	0.10	0.12	0.15	0.17	0.18	0.19	0.17
568	**499**	**501**	**565**	**723**	**1054**	**1218**	**1133**	**827**
0.21	0.18	0.16	0.17	0.18	0.18	0.18	0.18	0.14
0.11	0.10	0.10	0.12	0.15	0.17	0.18	0.18	0.16

ref. value
% rel. err. (IQS)
% rel. err. (FTM)

Root mean square error	
IQS:	0.16
FTM:	0.21

Figure 3-27: Assembly temperatures at the end-of-transient for IQS versus FTM. In green and red, errors that are lower and higher for FTM, respectively.

For alpha-eigenvalue methods, two alternatives to the classic alpha-eigenvalue were developed and tested. The hybrid alpha mode is much more effective than the modified alpha mode. This thesis recommends using this hybrid mode for all alpha-eigenvalue simulations of reactor transients. The hybrid alpha-eigenvalue method even outperformed the omega method, although at a greater expense.

Finally, for time differencing schemes, the coarse time integration is the least favorable method because it ignores the derivative of the amplitude when updating the flux shape. By adding frequencies, the coarse time integration method becomes the frequency transform method which showed large improvements. The improved quasi-static method and the frequency transform method share many similarities, especially in the behavior of the amplitude derivative terms. In fact, the improved quasi-static method can be considered a generalized form of the frequency transform method. The relative accuracy of the IQS compared to the FTM appears to be strongly problem

dependent: if the power change cannot be characterized as exponential, the more general IQS is recommended thanks to its flexibility. However, during exponential transients, the FTM is recommended for better precursor accuracy and therefore spatial resolution.

Overall, the use of exponential frequencies, especially spatially distributed precursor frequencies, has the largest impact in simulation accuracy for HOLO methods. The root mean square error in power distribution at the peak was consistently about 20% lower when using frequencies, as emphasized in Table 3.8. These frequencies allow for two-way communication between the high- and the low- order solvers.

Table 3.8: Impact of frequencies on peak power root mean square error (RMS).

	RMS error in absence of frequencies	RMS error in presence of frequencies	Difference in RMS error
k-eigenvalue	Adiabatic	Omega	22%
	0.79	0.57	
Alpha-eigenvalue	Modified Alpha	Hybrid Alpha	17%
	0.72	0.55	
Time-differencing	IQS	FTM	27%
	0.39	0.12	

While time-differencing schemes are most accurate, the eigenvalue methods are more adaptable to further applications in Monte Carlo transients. Furthermore, they require fewer outer time steps, significantly reducing the computational cost. The novel hybrid omega/alpha eigenvalue method shows improvement over existing eigenvalue solvers as a high-order method and deserves further consideration despite that in this implementation, the cost of solving for the alpha eigenmode is too costly to recommend over the omega method. For applications in Monte Carlo, this would require a stochastic alpha eigenvalue solver which is not currently available in OpenMC. For this work, the k-eigenvalue methods are the most adaptable. In the following section, the adiabatic and omega methods are implemented in OpenMC. In Chapter 4, all examples use the omega method.

3.4 Omega Method for Monte Carlo

Historically, the point kinetics equations have been widely used as the low-order method in time dependent calculations. As described in Section 3.3.1, point kinetics use the initial shape function of the flux and assume the amplitude is only a time dependent scalar, resulting in poor spatial resolution. The spatial dependence of the flux amplitude can be preserved by using a time dependent Coarse Mesh Finite Difference (TD-CMFD) solver instead of point kinetics. This method solves time dependent diffusion where equivalence factors and cross sections are provided from high fidelity, frequency-transformed Monte Carlo, implemented as the omega method. The diffusion equations can be solved in an Improved Quasi Static time-integrated scheme with an analytic solution to the precursor equations. This formulation was recently published [76] and will be further demonstrated in this section. First, analysis from [76] is recomputed, and further problems are carried out to demonstrate the impact of frequencies in transient problems.

Coupling Monte Carlo and Coarse Mesh Finite Difference is ideal for transients on the order of a few seconds since no nuclide depletion or build-up is included. The two solvers exchange information based on the omega method formulation of the neutron transport equation. The omega method approximates the derivative terms of the transport equation with exponential frequencies as detailed in section 3.3.1. This omega method is also sometimes referred to as the exponential transform or frequency transform method [54, 89]. In a slightly different form, exponential frequencies have also been used as a stiffness confinement method similar to treating prompt and delayed neutrons separately [96, 97].

A frequency is determined for each cell, energy group, and precursor group during the Time Dependent Coarse Mesh Finite Difference calculations. By definition, CMFD is a diffusion solver with an additional non-linear diffusion coefficient term. This term is a correction term which is required for consistency between the high- and low-order methods. In this case, it is used to maintain local consistency between Monte Carlo and diffusion. TD-CMFD receives information from Monte Carlo such

as multi-group cross sections, currents, and diffusion coefficients. With that information, the TD-CMFD solver can carry out the time dependent transient for a number of time steps. As described in [76]: "A MC solve is performed over long time intervals, Δt^s, to compute MGXS on the fine mesh and cell-to-cell currents on the coarse mesh. At the initial time step, the space, angle, and energy integrated MGXS are used to compute the fine mesh powers. The fine mesh powers are then normalized to a user input value and the delayed neutron precursor concentrations are computed. The shape-weighted MGXS on the fine mesh are collapsed down to the coarse mesh and interpolated on time intervals Δt^a. When MGXS and the flux shape are requested at a time point in between MC solves, these parameters are interpolated from the most recent MC solves. The flux amplitude and precursor concentration are then propagated forward over the Δt^a time interval. After both the inner and outer solves, the fine mesh powers are computed and checked for convergence to a user input value. The use of fine mesh cross sections allow for feedback to be conducted on the fine mesh during the inner solves. However, this is seen as a large effort in and of itself and is therefore recommended for future work." In this implementation [76, 77], the TD-CMFD matrix entries include reaction rates in the form of cross sections multiplied by the Monte Carlo flux (shape). Therefore, the TD-CMFD resulting amplitude is actually a measure of how different Monte Carlo and TD-CMFD results are. In prior work, the shape and amplitude provided a way to communicate information from fine mesh Monte Carlo tallies to coarse mesh TD-CMFD [76]. In this work however, the TD-CMFD mesh was selected to match the Monte Carlo tally mesh. This was done to facilitate the time dependent thermal hydraulics feedback coupling to the single channel model of Chapters 4 and 5, mentioned above as "future work" which is undertaken in this thesis. Thus, with proper preservation provided by the non-linear diffusion coefficients enforcing balance in each TD-CMFD mesh, the amplitude function should be unity. However, small deviations are observed due to tally inconsistencies between tracklength and analog tally estimators. As laid out in [31], tracklength estimators are more accurate, but they are not available for every type of tally, namely transport cross sections, consistent scatter matrices, χ, and currents.

The Monte Carlo calculations are eigenvalue calculations, including the first steady-state calculation at t=0. This implies, as quoted from [76], that as "an eigenvalue problem, the coarse mesh amplitude will not correspond to any real power level. Therefore, the initial flux shape will be normalized to a user defined initial core power." Then, the "shape function following any time dependent MC solve is then defined as the normalized flux tallied from the MC solve divided by the amplitude."

Since the Monte Carlo calculations are the most expensive part of the simulation, the TD-CMFD should be carried out for as long as possible, keeping the total number of Monte Carlo calculations small. The calculation returns to Monte Carlo whenever the change in the frequency becomes large or after a fixed number of time steps. At the end of the TD-CMFD time, the instantaneous frequencies are computed and passed to Monte Carlo. They represent the current state of the prompt neutron and delayed precursor populations. These frequencies serve as the approximation for the derivative term in the transport equation and therefore become inputs to Monte Carlo.

The frequency treatment in Monte Carlo is detailed and validated in [76] and has been re-implemented for this thesis in C++ based OpenMC: "Monte Carlo is an event-based method that is formulated such that a reaction is randomly sampled from a probability distribution at each collision. The frequency therefore must be input as an additional absorption (if positive) or production (if negative) reaction term. The absolute value of the frequency shall be added to the total cross section used to compute the distance to collision. During the collision, the absolute value of the frequency is used as a distributed absorber cross section. If the distributed absorber reaction is sampled, the neutron is either killed (if the frequency is positive) or a secondary neutron is born with the same weight, energy, and trajectory as the incident neutron (if the frequency is negative)." For the precursor frequencies, "the frequency-weighted delayed-nu-fission cross section needs to be used in tallying the eigenvalue with track-length and collision estimators. In continuous energy Monte Carlo the precursor decay constant, $\lambda_i(r,t)$, is only known on a nuclide basis. To avoid looping over all nuclides for this tally and because the decay constants for a

given precursor group will be nearly identical for all nuclides, the spatially-integrated precursor decay constant on the fine mesh is used."

In summary, Figure 3-28 shows how the two solvers communicate with each other.

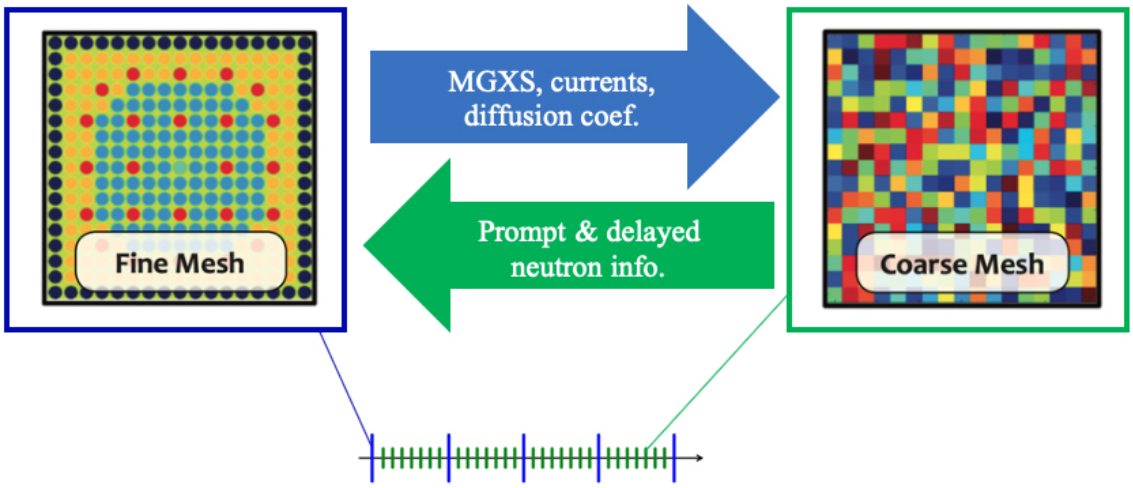

Figure 3-28: Communication diagram between the high-order Monte Carlo solver and the low-order TD-CMFD solver.

For delayed neutron modeling, an alternative to Dynamic Monte Carlo has been introduced to save a significant amount of computation and memory [76]. The precursors are homogenized over small regions, and their concentrations are propagated in time by integrating the precursor balance equation. This assumes the use of solid fuel so precursors are not migrating during the transient. This technique is particularly well-suited for TD-CMFD as the homogenized regions can overlap with the mesh used in diffusion. For the Monte Carlo sampling of delayed neutron parameters, "It is expected that the uncertainty in the delayed neutron fraction, precursor decay constants, and prompt neutron lifetime will be relatively small as they will all use track-length estimators or have very small variability across the core (e.g. the precursor decay constants) [76]."

3.4.1 Results

Example problems are carried out with the omega method as implemented in OpenMC [42]. The first problem is on the C5G7 geometry [98]. C5G7 is a widely simulated bench-

mark due to its heterogeneous geometry and the recent addition of time dependent test problems. The problem modeled here is a modified version of one of the C5G7 benchmarks. It is a prescribed drop in moderator density in the inner most fuel assembly (as opposed to a uniform drop in the whole geometry) with no intrinsic feedback. The transient is fully prescribed, and there is no thermal hydraulic solver in the simulation. The following examples resemble the work in Shaner's thesis [76]. Some differences appear, namely in Figures 3-32, 3-33, and 3-34 because the group structures are different, and different values of v are used in the calculation of the flux frequency terms for these plots. Furthermore, the diffusion solver (used in both Shaner's work and these results) contained a small error in the calculation of non-linear diffusion coefficients that was not found until after this work was carried out. However, since the role of non-linear diffusion coefficients is to force balance between CMFD and Monte Carlo, this does not significantly impact the final results. However, it does affect the values of the non-linear diffusion coefficients. Therefore, the analysis section on diffusion coefficients in [77] is erroneous.

This simulation was computed with 1 million particles per batch, with 100 batches including 40 inactive batches. Outer time steps are 0.5s and inner steps are 0.01s, and outer time steps were forced to iterate twice. In this 2-second transient, the moderator density drops (only in the top right corner) to 80% of its value linearly in the first second. Then it rises back to its original value linearly over the next second. Figures 3-29 and 3-30 show the core power and reactivity change in time, respectively, both for the omega method and the adiabatic method. Figure 3-31 further illustrates the difference between the two methods with the discrepancy in pin powers at 1 second. This difference, which is concentrated in the region of the moderator density change can be explained by the distribution of the precursor frequencies, shown in Figure 3-32. Since the adiabatic method neglects the frequencies, it fails to capture the extent of the transient. Finally, Figures 3-33 and 3-34 illustrate how the flux and precursor frequencies change in time over the course of the transient. Note that flux frequency for each energy group is a core average quantity. In light of the relatively short prompt neutron lifetimes of most reactors, "prompt neutron spatial distribution

will rapidly equilibrate suggesting that the flux frequency can be approximated as spatially flat," although this approximation is not likely to hold for rapid transients like rod ejections [76]. Furthermore, the flux frequencies shown in 3-33 are nearly proportional to velocity, so the frequencies $\omega^{P,g}$ are nearly the same in each group. The inverse velocities range from 4.47×10^{-10} to 3.55×10^{-6} which accounts for the small scale of the flux frequency terms. For the precursor frequencies, the precursor groups with the longest half-lives have the "largest percentage increase in delayed neutron fraction. This increase in the frequency-weighted delayed neutron fraction of nearly five times the steady-state value will significantly soften the flux spectrum in higher energy groups leading to lower fraction of the neutrons produced via fast fission. Furthermore, this will act to resist spatial shifts in the flux for negative reactivity insertions and accelerate shifts in flux for positive reactivity insertions. When the moderator density is increased between t=1-2 seconds, the short half-life groups increase more rapidly and their effective delayed neutron fractions fall below their steady-state values. This results in a scenario at time t=2.0s where the core-integrated frequency-weighted delayed neutron fraction of the long half-life groups are above their steady-state values and the short half-life groups are below their steady-state values. Once the moderator density is back to the initial value, the precursor groups take several half-lives to return to near their steady-state values." [76].

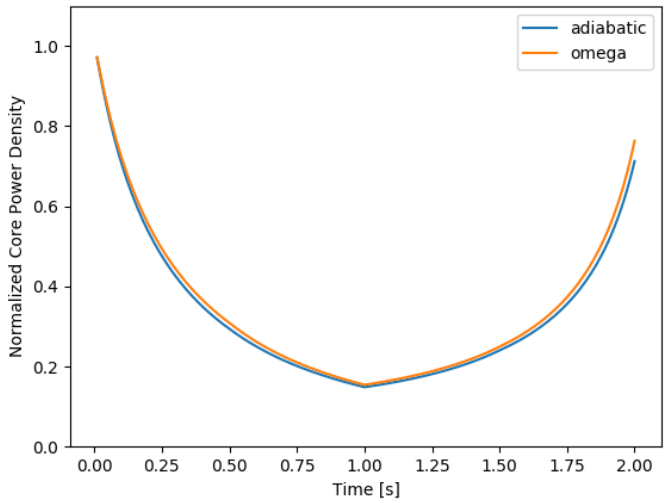

Figure 3-29: Normalized core power density of the C5G7 geometry under a prescribed moderator density transient for 2 seconds.

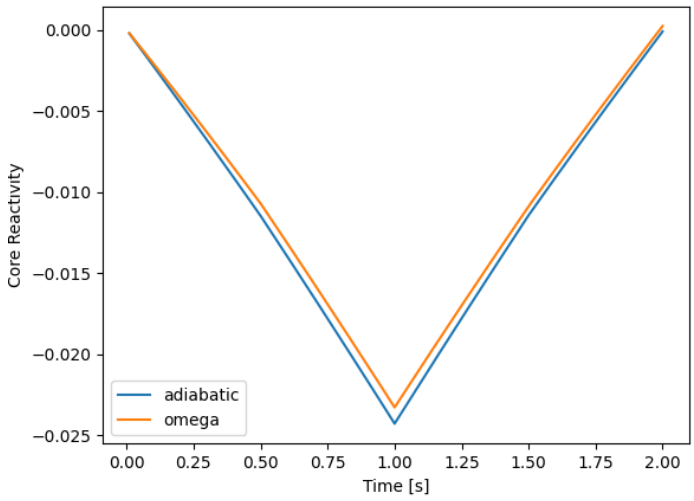

Figure 3-30: The core reactivity of the C5G7 geometry under a prescribed moderator density transient for 2 seconds.

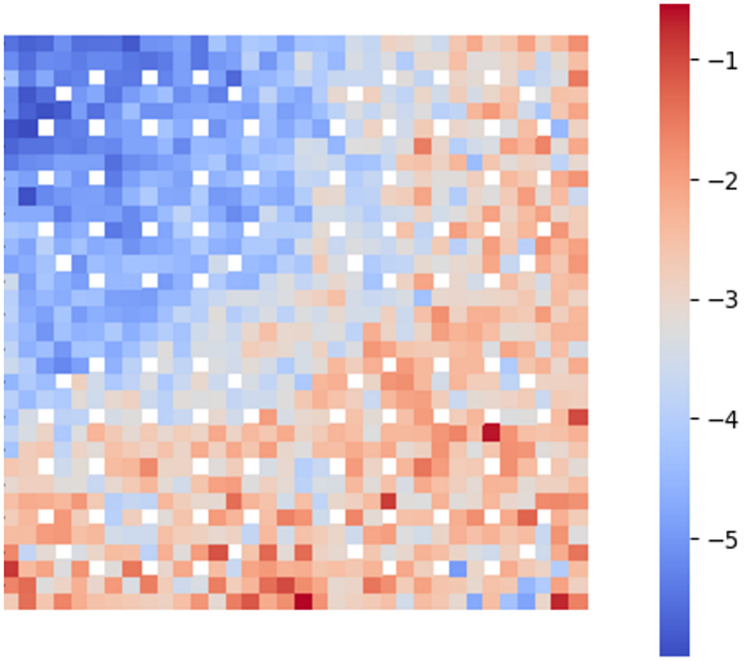

Figure 3-31: Relative difference (%) in pin powers between the omega method and adiabatic approximation at 1 second.

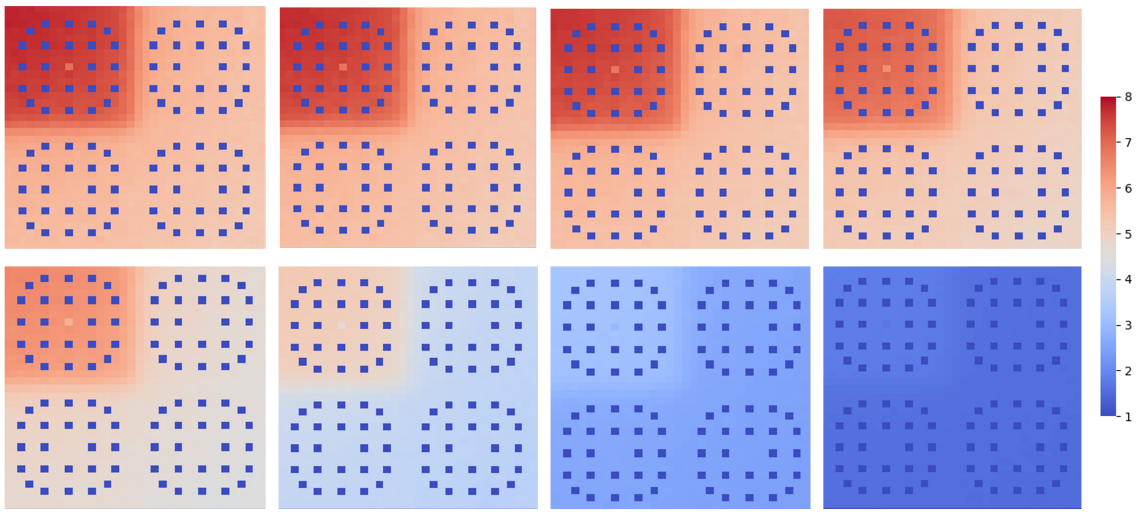

Figure 3-32: Spatial distribution of the precursor frequency terms in the C5G7 geometry at 1 second for each of the 8 delayed precursor groups (starting at group 1 in the top left-hand corner). The delayed frequency term ($\frac{\lambda}{\lambda+\omega^{D,i}}$) is dimensionless.

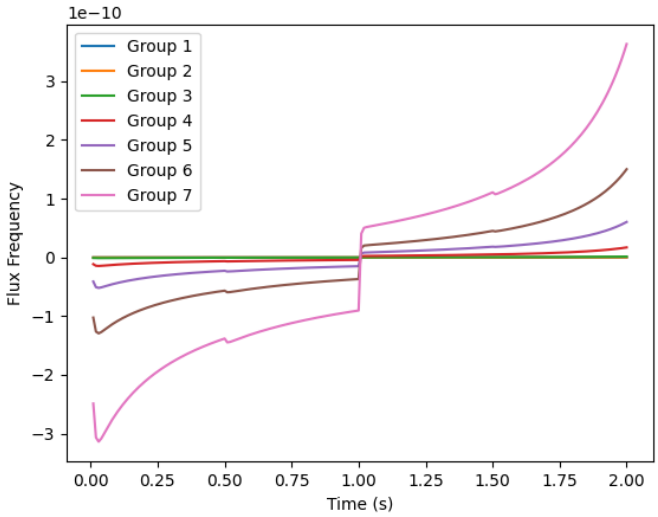

Figure 3-33: Evolution of the flux frequency terms in time ($\frac{\omega^{P,g}}{v}$). Flux frequencies are a spatially flat.

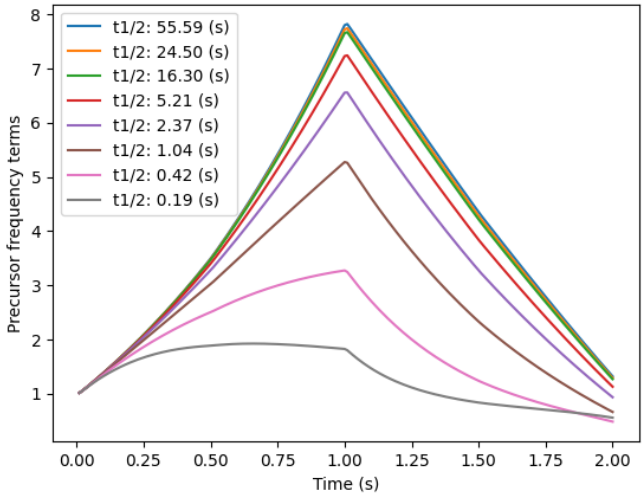

Figure 3-34: Evolution of the precursor frequency terms in the top left corner pincell in time ($\frac{\lambda}{\lambda+\omega^{D,i}}$).

Next, the behavior of the omega method in a spatially uniform problem is tested on fresh LWR fuel rod is compared to the adiabatic approximation. The geometry is 365.76 cm long without spacer grids, end plugs, or reflectors, undergoing a similar transient where the moderator density drops for 1 second then rises again. The density change is uniform across the entire fuel rod. Since the adiabatic approximation amounts to neglecting the time derivatives, it is considered less accurate than the omega method. Figure 3-35 shows the end-of-transient axial power profiles, including one for a null transient. Figure 3-36 shows the relative difference in the axial power between the two methods at the end of the transient, with discrepancies up to 2%. These results emphasize that frequencies make the biggest difference in spatial resolution, but continue to add accuracy in spatially uniform problems as well, although to a smaller extent. In this fuel rod model, the frequencies are calculated on a mesh with 30 axial cells and a single radial cell. Run parameters are 300,000 particles per batch, 1100 batches, including 100 inactive batches. Again, outer time steps are 0.5s, inner time steps are 0.01s, and outer time steps are iterated twice.

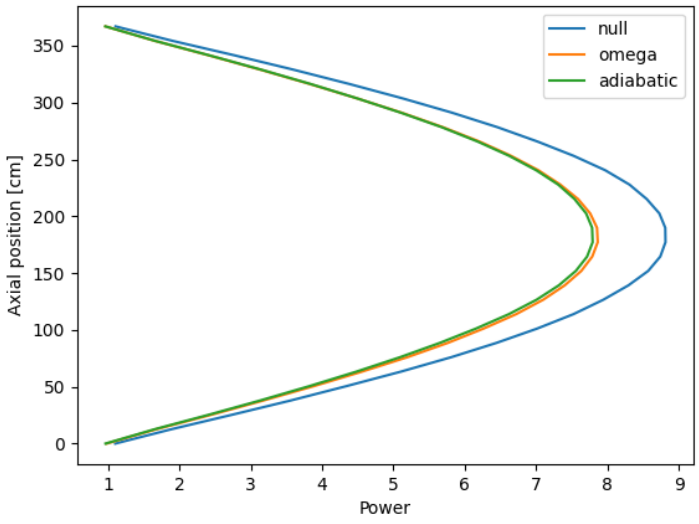

Figure 3-35: End-of-transient power for omega and adiabatic methods in a fuel rod after a 2 second moderator density transient compared to the null transient.

This method was also tested to hold steady-state over 2 seconds on a fuel rod. That is, the omega method is applied, but no transient is prescribed. The maximum

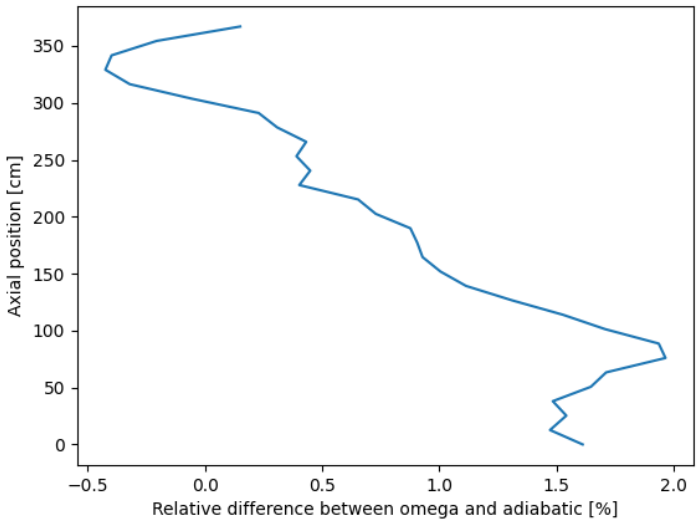

Figure 3-36: Axial power difference between the omega and adiabatic methods in a fuel rod after a 2 second transient.

variation from steady power was found to be 1% and can be further decreased with larger numbers of particles. Considering the stochastic nature of the calculation, this is considered a successful steady-state. These variations are tied directly to the neutron batch size so lower neutron counts result in larger variations in the absence of feedback effects that can contribute to stability. These results were obtained with 1 million particles per batch over 200 batches with 100 inactive batches.

3.4.2 Conclusion

The frequency transform applied to Monte Carlo as the omega method is an efficient method for simulating transients on the order of a few seconds. The method relies on the approximation of the derivative terms in the time dependent neutron transport equation by frequencies. These frequencies are calculated by time dependent Coarse Mesh Finite Difference during inner time steps and are passed as inputs to Monte Carlo during outer time steps. The results in this section reinforce the analysis in 3.3.1 to show that the omega method is more accurate than the adiabatic approximation since the time derivatives are taken into account and the spatial distribution of the frequencies provides additional information otherwise lacking in the adiabatic

method. One of the problems tested in this work showcased non-uniform changes in the geometry. By definition, the omega method allows for such shape tilts thanks to the mesh on which frequencies are calculated.

As laid out in 3.3.2 and throughout this section, the omega method emerges as the most efficient, accurate, and easily implemented method for Monte Carlo High-Order/Low-Order transients. While the time-differencing methods and alpha-eigenvalue methods analyzed in detail in Section 3.3 show promise with respect to accuracy, their implementation for Monte Carlo currently presents challenges that the omega method overcomes. In the remainder of this thesis, the omega method is used and implemented exactly as described in Section 3.4.

Chapter 4

Thermal Feedback Coupling

In the methodology laid out in Chapter 3, transients were fully prescribed, meaning that they relied on user input to change the material properties. In this chapter, a thermal feedback model has been included to make the methodology self-propagating.

4.1 Transient Thermal Feedback Model

In this coupling scheme, ramp-up Monte Carlo (as described in Chapter 2) is used in the first Monte Carlo calculation to establish the baseline temperature and density profile of the steady-state problem before the transient kicks off. To summarize the description of the steady-state thermal fluid solver, the power tally in OpenMC is used to calculate the enthalpy of each spatial cell in the geometry mesh. That enthalpy dictates how the bulk coolant is heated. Then, a heat transfer resistance model (detailed in Section 2.3.1) is used to calculate the temperature of the near-rod coolant, the cladding, the gap, and an average fuel temperature. It is known that different transients are driven by different heat transfer equations [99]. For instance, thermal capacity effects (which are included here for the fuel and coolant, but ignored in the cladding and gap) can play an important role if the characteristic time scales of the neutronics problem are short compared to the characteristic time scales of the thermal diffusion problem in the materials. However, for the purposes of showcasing the novel aspects of this work, the equations that govern steady-state heat transfer

are extended to the transient, and additional heat capacity effects are neglected. New heat transfer coefficients are calculated in each cell at each time step, but the heat transfer regime is not modified between the steady-state and the transient scenarios. These approximations in the thermal hydraulics solver are not expected to impact the numerical methods that are showcased in this chapter.

For the transient prediction of next step material properties, an explicit time stepping solver is implemented. Equations (4.1) and (4.2) are solved in tandem. The variable i is the cell of the mesh along the fuel rod, A is cell area $[m^2]$, L is cell length $[m]$, M is mass in the cell $[kg]$, c_p is specific heat $[J/K/kg]$, G is mass flux $[kg/m^2/s]$, q' is linear power $[W/m]$, h_{cool} is enthalpy in the coolant $[J/kg]$, $k_f(T_f)$ is temperature dependent fuel conductivity $[[W/m/K]]$, T_f is fuel temperature $[K]$, and T_{cool} is bulk coolant temperature $[K]$. Quantities at i are cell-averaged, while quantities at $i \pm 1/2$ refer to the cell interfaces as described in equations (2.6) (2.7), (2.8), and Figure 2-1: cell interface enthalpies are used to calculate cell interface temperatures, which are then averaged to determine properties in cell i. Thermal resistance is calculated with a gap conductance of 31 kW$/m^2$/K [38], cladding conductivity of 16.08 W/m/K [39], and temperature dependent fuel conductivity of Equation (2.11) [40]. Only convective heat transfer is used for the water at the cladding surface (no nucleate boiling is taken into account).

$$\begin{gathered} M_{cool}(T^n_{cool,i})c_p(T^n_{cool,i})\frac{T^{n+1}_{cool,i} - T^n_{cool,i}}{t^{n+1} - t^n} = \\ 4k_f(T^n_{f,i}) \cdot 2\pi L(T^n_{f,i} - T^n_{\text{fuel\_surface},i}) \\ -G^n A(h^n_{cool,i+1/2} - h^n_{cool,i-1/2}) \end{gathered} \tag{4.1}$$

$$\begin{gathered} M_f c_p(T^n_{f,i})\frac{T^{n+1}_{f,i} - T^n_{f,i}}{t^{n+1} - t^n} = \\ q'_i L - 4k_f(T^n_{f,i})2\pi L(T^n_{f,i} - T^n_{\text{fuel\_surface},i}) \end{gathered} \tag{4.2}$$

where $k_f(T^n_{f,i})$ is from Equation (2.11), and $T^n_{\text{fuel\_surface},i}$ is calculated by Equation (2.10).

For a single, 365.76 cm, light water reactor fuel rod model identical to the description in 2.3, the steady-state axial temperature distributions are shown in Figure 4-1.

The fuel is enriched to 2.4% with no burnup, and the water contains 432.473 parts per million boron. The impact of boron on water density is ignored in all the thermal hydraulic calculations in this thesis, including in the use of steam tables.

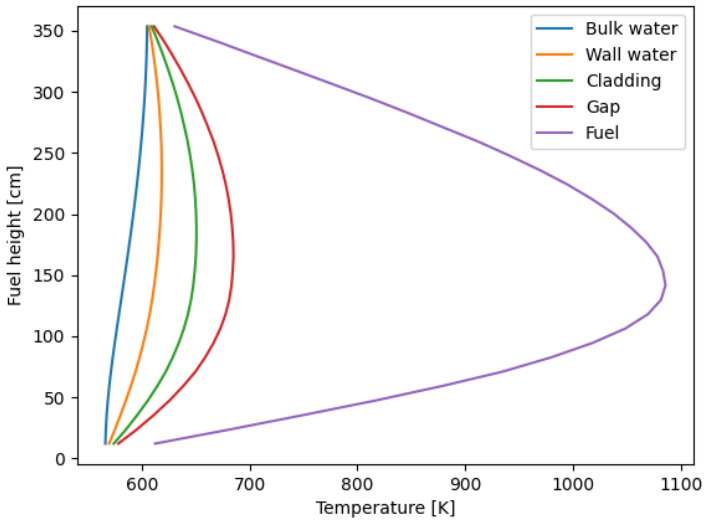

Figure 4-1: Steady-state axial temperature distributions.

4.1.1 Null transient of the thermal feedback model

As a check-point for consistency, the null transient behavior of the thermal solver alone is studied. The null transient is guaranteed to hold if the TD-TH equations above collapse exactly to the steady-state equations when no transient is simulated. The steady-state TH conditions are generated in an iterative calculation between TH and Monte Carlo neutronics, as explored in detail in Chapter 2. This steady-state calculation takes place separately before the time dependent calculation and results in input temperature distributions that are applied to the first Monte Carlo neutronics calculation in the time dependent scheme, as illustrated in Figure 4-2. The iterative nature of the SS MC/TH calculation means that the power distribution (q') is not exactly identical in two consecutive iterations of the steady-state calculation, even if convergence has been achieved. After the ensuing temperature distributions are applied to the first Monte Carlo neutronics calculation at t=0 in the time dependent

scheme, the resulting value of q' will be slightly different than the value of q' used at steady-state to generate those temperatures, within statistical fluctuations.

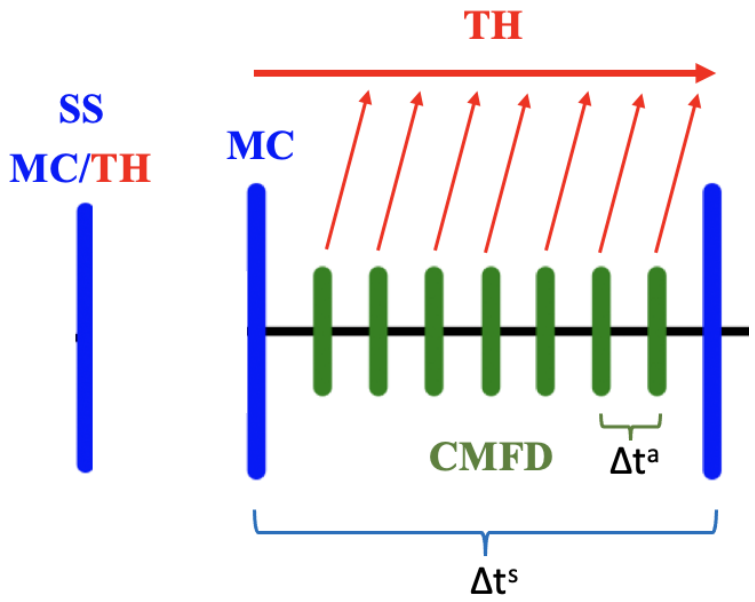

Figure 4-2: The steady-state calculation of MC/TH precedes the time dependent scheme to provide input temperatures for the geometry.

Therefore, there are two options for q' in Equation (4.2). The first is to use the power profile that generated the steady-state temperatures. This ensures that Equation (4.2) collapses exactly to the steady-state condition during the null transient and holds steady power very tightly. The second option is to use the power profile tallied during the Monte Carlo calculation at t=0, after the steady-state temperatures are imposed. If the steady-state instabilities are successfully damped, and the Monte Carlo convergence is high, this should still hold steady-state, but within a larger margin of error.

In the following figures, the initial temperatures were generated with a ramp-up Monte Carlo progression of 10, 50, 100, 200, 300, 400, 500, 600, 700 batches before the maximum temperature change across all cells was less than 10 K between iterations, as established in Chapter 2.

Figure 4-3 shows how far coolant temperatures depart from the steady-state during the TD-TH null transient using the first power option (the steady-state power

tally), and Figure 4-4 is using the second option (power tally from t=0 calculation). In both figures, a time step of 2.5 x 10^{-3} seconds is used in the calculation of equations (4.1) and (4.2), as determined in Section 4.1.2. The solver is shown to hold steady-state coolant temperatures within approximately 1.5×10^{-4} K, and fuel temperatures within 4×10^{-3} K in Figure 4-3 with the steady-state power tally. Whereas the solver holds coolant temperatures within 3×10^{-2} K and fuel temperatures within 1.5 K in Figure 4-4 with the power tally from the Monte Carlo calculation at t=0.

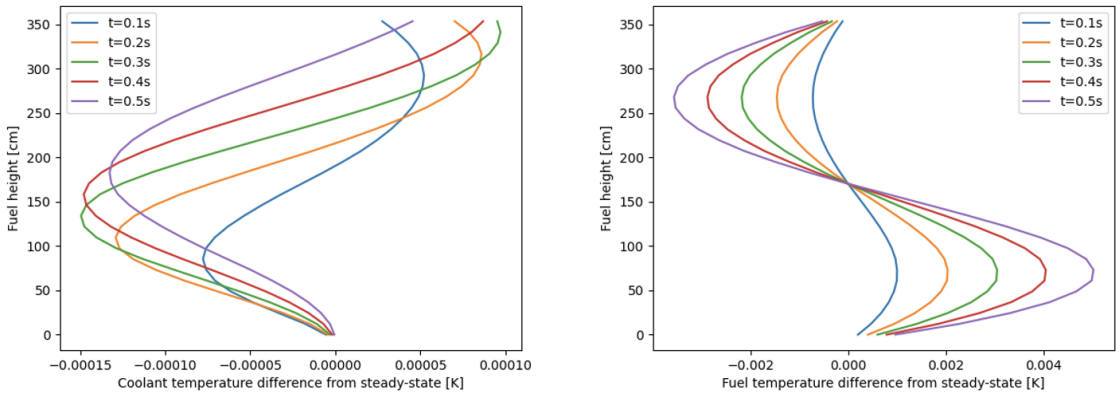

Figure 4-3: Null transient behavior of the time dependent coolant (left) and fuel (right) temperature equations, as compared to steady-state temperatures using the power profile that generated the temperature distribution.

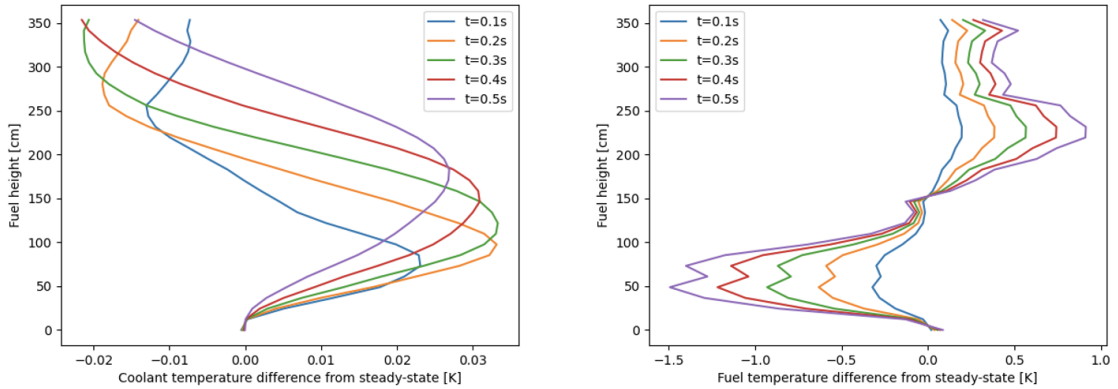

Figure 4-4: Null transient behavior of the time dependent coolant temperature equations, as compared to steady-state temperatures using the power profile tallied after the temperatures are imposed.

This shows that the fluctuations from the non-linear coupled power/temperature multiphysics iterations can be minimized by using the power profile that generated the

temperature distribution at the start of the calculation. However, it is advantageous to be able to use the solver with input temperatures without relying on the user to generate them and save the input power distribution. Therefore, option 2 will be used in the remainder of this thesis, where power is tallied during the Monte Carlo calculation at t=0.

At steady-state, the power tally is normalized to 66945.4 W total power for the fuel rod based on [44]. During the inner time steps of a transient, when the total power is changing, the power distribution is calculated by TD-CMFD as the kappa-fission tally times the flux. In order to ensure that the normalization is consistent throughout the transient, the initial power distribution is employed as a normalization factor using the following formula where i is the axial cell:

$$q_i'^{t_{n+1}} L = q_i'^{t_n} L \times \frac{\text{TD-CMFD.power}_i^{t_{n+1}}}{\text{TD-CMFD.power}_i^{t_n}} \tag{4.3}$$

with

$$q_i'^{0} L = \text{TD-CMFD.power}_i^{t_0} \times \frac{66945.4\text{Watts}}{\sum_i \text{TD-CMFD.power}_i^{t_0}} \tag{4.4}$$

4.1.2 Inlet change transient of the thermal feedback model

As the thermal solver consists of an explicit time stepping scheme, the time step size must be chosen in consideration of the Courant-Friedrichs-Levy stability criterion:

$$\frac{v\Delta t}{\Delta x} < 1 \implies \Delta t < \frac{\Delta x}{v} \tag{4.5}$$

Given a coolant mass flux of 3127 kg/m^2/s and a water density of approximately 700 kg/m^3, the coolant velocity is estimated at 4.46 m/s. With a mesh size of approximately 12 cm in the direction of flow, the estimated time step limit for stability is on the order of 3 x 10^{-2} s. A test comparing the results of a transient with time steps ranging from 2 x 10^{-2} s to 6.25 x 10^{-4} s is conducted. The transient is an instantaneous increase in inlet coolant temperature from 566.25 K to 570 K at $t = 0^+$. The coolant temperature solution of the finest time step solution in shown in Figure 4-

5. The mass flux of 3127 kg/m^2/s suggests that the inlet temperature change should take approximately 0.4 seconds to reach the center of the fuel rod, with some room for error associated with spatial discretization and the fact that the water density is not constant. Figure 4-5 shows that after 0.4 seconds, the temperature change has reached 225 cm, which is a bit higher than this estimate, but within reason. The absolute difference from that reference solution at 0.5s, using coarser time steps, is shown in Figure 4-6. Figure 4-7 shows the same data, but in absolute value on a log scale. This test estimates that the thermal solver converges at a nearly linear rate, as shown in Figure 4-8. A time step size of 5 x 10^{-2} s (not shown here) was found to be too large and cause instability.

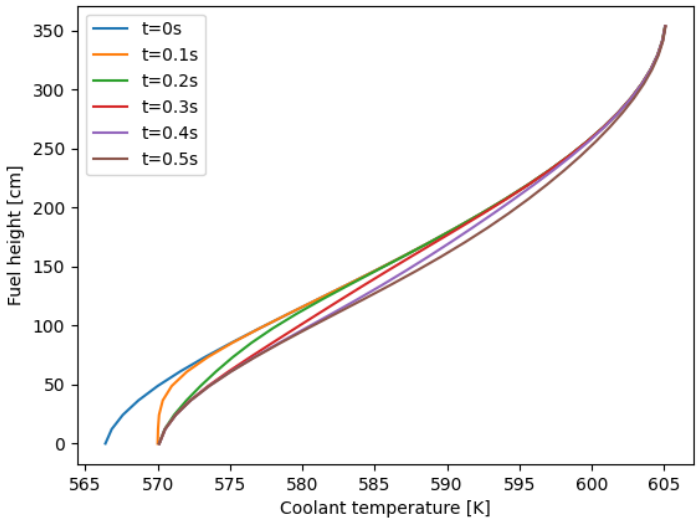

Figure 4-5: Coolant temperature solution of the inlet temperature rise with dt=6.25x10^{-4} s.

While Figures 4-5 through 4-8 demonstrate the accuracy of the time dependent thermal hydraulics solver for coolant temperature, the associated fuel temperature change is too small to evaluate accuracy. An additional example is needed to demonstrate the fuel temperature response to transients. To simulate fuel cooling, the power term q' in Equation (4.2) is set to 0. Fuel temperature behavior over 1.0 second is shown in Figure 4-9.

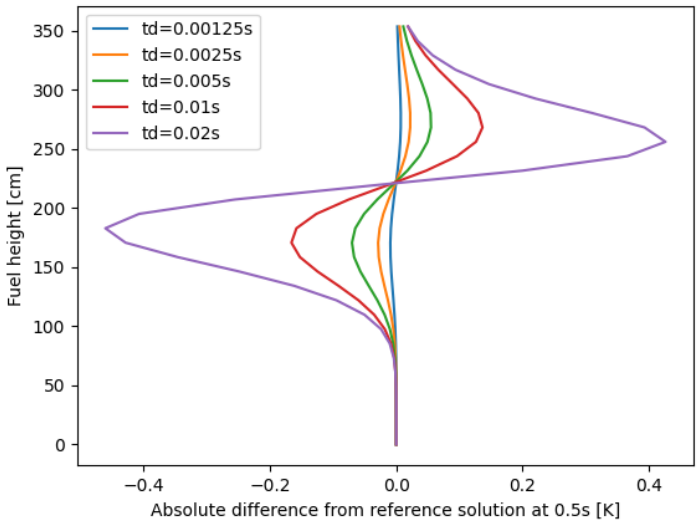

Figure 4-6: Absolute difference in axial temperature from the dt=6.25x10^{-4} s solution at 0.5s.

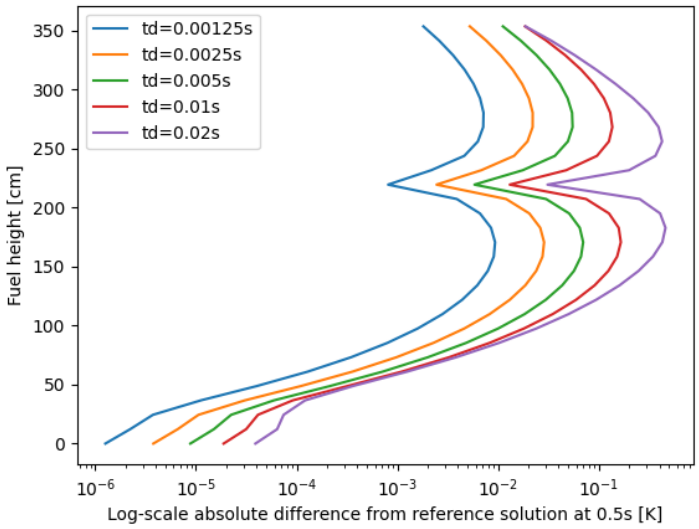

Figure 4-7: Log scale difference in axial temperature from the dt=6.25x10^{-4} s solution at 0.5s (using absolute values).

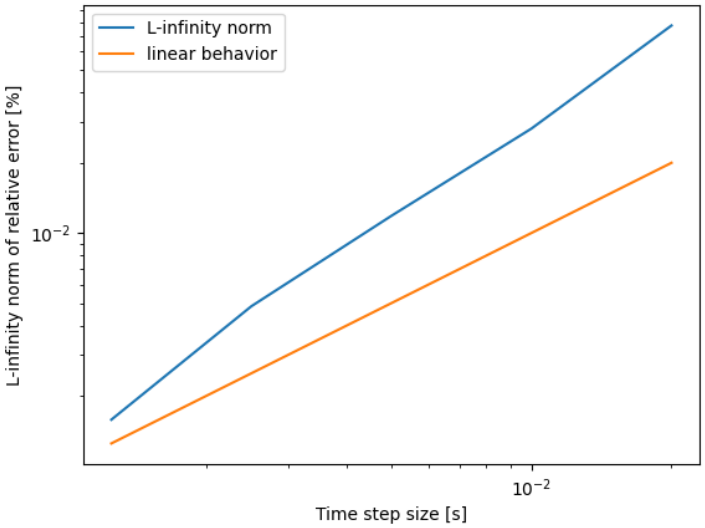

Figure 4-8: L-infinity norm of difference from the dt=6.25x10^{-4} s solution.

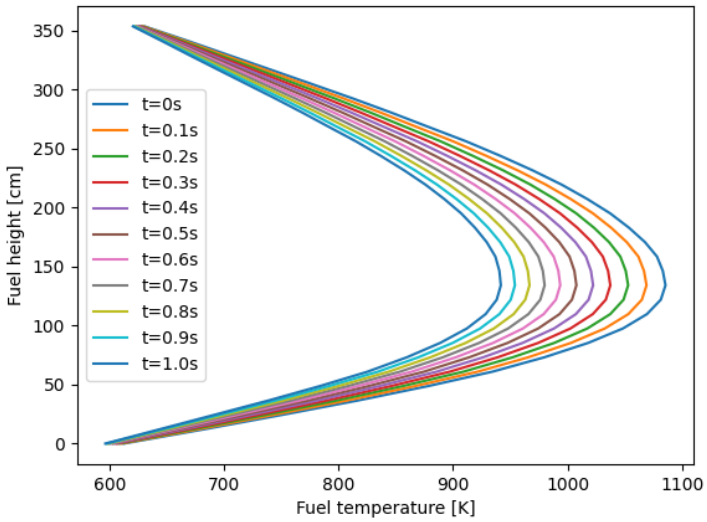

Figure 4-9: Fuel temperature solution of the zeroed power transient for 1 second with dt=6.25x10^{-4}s.

4.1.3 TD- TH/CMFD coupling

With the addition of the methods described above, the inner time steps, noted t^a (for amplitude), represent a TD- TH/CMFD calculation, whereby the time dependent diffusion (TD-CMFD) communicates at every step with the transient thermal hydraulics (TD-TH) to incorporate an updated value of power into Equation (4.2), as illustrated in Figure 4-10. This is a staggered communication, in that the power provided by TD-CMFD at time step t^{a-1} is used to calculate the temperatures in the TD-TH solver at t^a. The outer time steps are noted t^s (for shape).

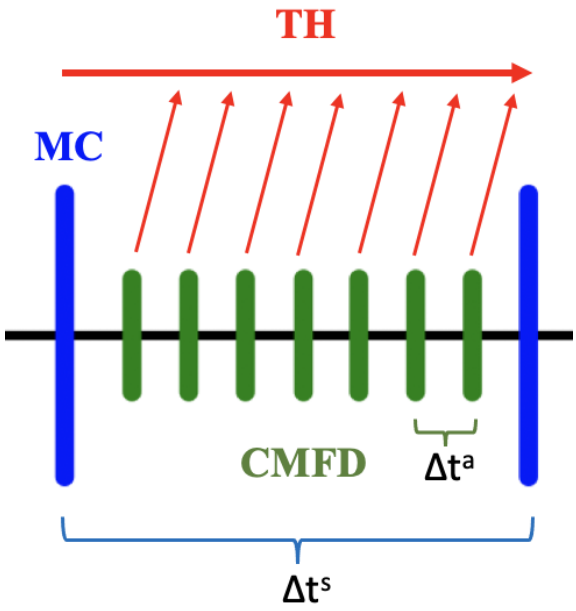

Figure 4-10: Time stepping illustration of the transient procedure with TD-TH/CMFD.

Then, the transient scheme from Chapter 3 can be implemented for self-propagating problems in the steps of Figure 4-11, which is similar to the flowchart found in [76], but modified to reflect the TD-TH solver addition. In the TD- TH/CMFD loop, the first decision branch focuses on the solution of the forward outer Monte Carlo solve at t^s. If this is the preliminary iteration over the inner time steps, that forward outer solution is not known, and therefore the TD- TH/CMFD must keep constant MGXS, coupling coefficients, and shape while calculating the temperature changes in the geometry. This implies that TD-CMFD will remain constant, and therefore the

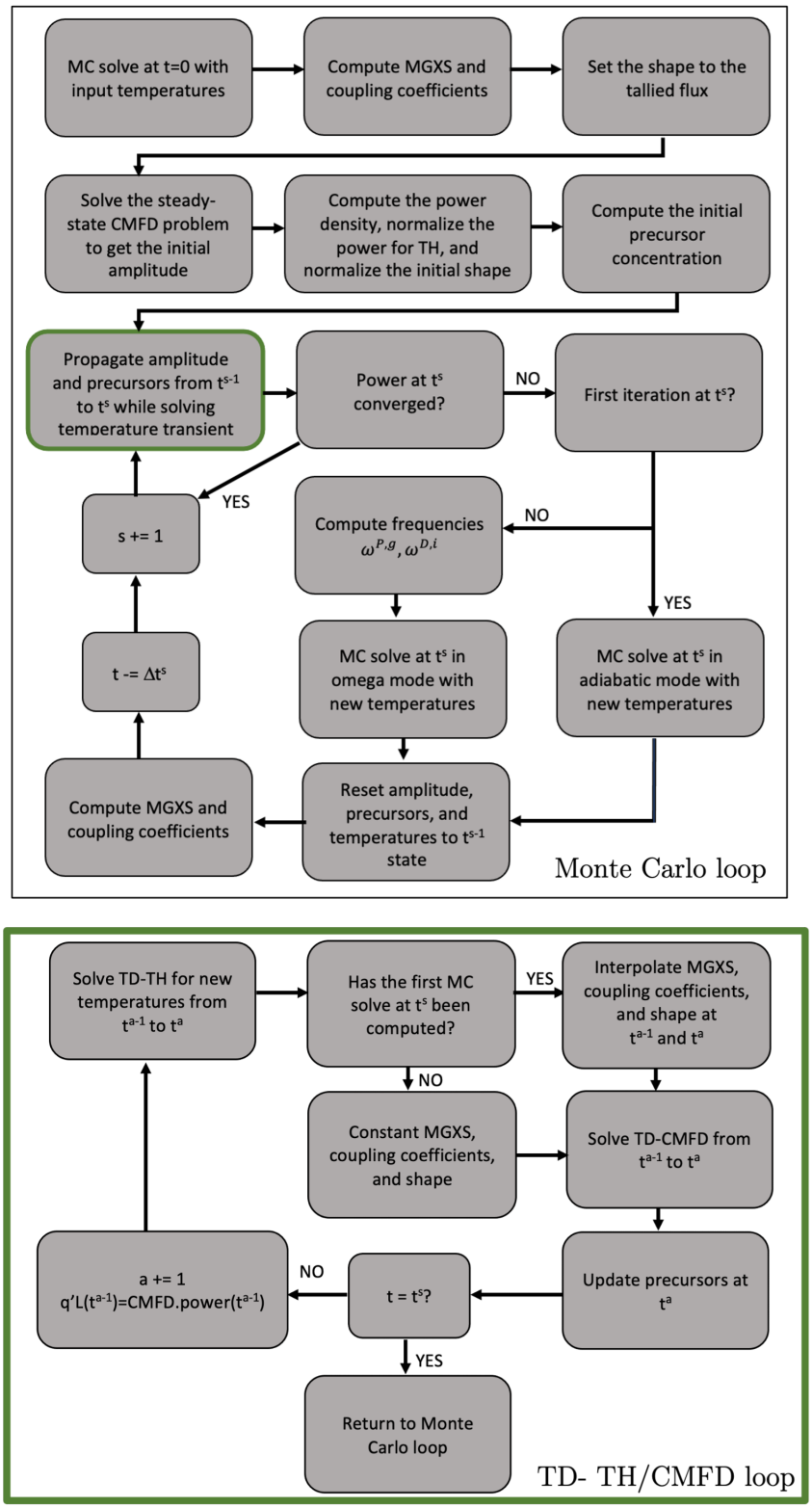

Figure 4-11: Flowchart of solution procedure.

power term in the TD-TH fuel equation will also remain constant. This iteration is referred to in this thesis as the "zeroth iteration" of the inner time steps as it allows a preliminary prediction of the temperatures at t^s, but an accurate power calculation cannot take place. After this zeroth iteration, the forward outer Monte Carlo can be calculated with new temperatures as indicated in the flowchart.

After the Monte Carlo calculation at t^s has taken place, the interpolation of MGXS, coupling coefficients, and shape in the TD- TH/CMFD loop can be implemented in various ways. In Chapter 3, a linear-in-time weighted interpolation of cross sections is used, according to the following equations, where Σ_a is the inner step cross section, w is the weight of the previous Monte Carlo cross section at t_s, $(1-w)$ is the weight of the forward Monte Carlo cross section at t_{s+1}, and i is the axial cell:

$$\Sigma_{a,i} = \Sigma_{s,i} * (1 - w_i) + \Sigma_{s+1,i} * w_i \tag{4.6}$$

$$w = 1 - \frac{t_{s+1} - t_a}{t_{s+1} - t_s} \tag{4.7}$$

This expression for w is not cell-dependent, as time advances at the same rate in every cell. This approximation appears sufficient for 2-D radial problems, such as in Figure 3-29. However, in the axial transient of Figure 4-5, the cross sections at the base of the fuel rod are changing before the cross sections at the top of the fuel rod. Therefore, an interpolation based on the time dependent evolution of the material properties becomes necessary to capture the different rates at which cross sections are changing. Since this particular transient is primarily driven by the density of the water, each cell's water density change can be evaluated against the total water density change expected for that cell over the time step. This uses the dominant material change to gauge the time scale of the cross section change, but non-dominant effects (such a fuel temperature change) still influence the magnitude of those cross section changes. However, this logic is not full-proof: for instance, water density changes in a lower cell can affect a higher cell due to related changes in the overall flux shape, resulting in some cross sections changing out of step with the spatially dominant change. The choice of which metric or material property to use to track spatial

progress of the transient in such a scheme depends on the nature of the transient. Equation (4.7) now becomes spatially dependent, with w as a vector of cells i.

$$w_i = 1 - \frac{\rho_{s+1,i} - \rho_{a,i}}{\rho_{s+1,i} - \rho_{s,i}} \tag{4.8}$$

While this captures the axial non-symmetry of the first second of the transient, there are cross section changes that continue after the water density has stabilized. For instance, when the water density is changing at the top of the fuel rod, the bottom of the rod is experiencing cross section changes that result from the new temperature distribution. Once the inlet temperature change has reached the top of the fuel, the transient continues with a more axially uniform power change. To account for both regimes where water density can be used as a spatial gauge for cross section changes, and when it cannot, a combined approach to cross section interpolation may be needed: if water density changes are small (less than 0.5% change over the entire outer time step), linear-in-time cross section interpolation is used. If the water density changes are larger, they are used to weight the cross section interpolation. This combination allows a smoother cross section solution, as will be seen in Chapter 5 examples and analysis. Ideally, a more general way of tracking the spatial progress of the transient could be developed to remove the necessity of understanding the driving transient effects before the calculation and to separate those effects (such as water density and fuel temperature).

4.2 Tally Derivatives

The transient procedure presented above requires running a previous and a forward outer Monte Carlo calculation in order for TD-CMFD to estimate the cross sections and the shape on the inner time steps. Only after this can the frequencies be accurately calculated and incorporated into an omega method run of the forward Monte Carlo. In total, this scheme requires a minimum of two Monte Carlo runs for every outer time step: one in adiabatic mode before the frequencies are calculated, and one with the omega method which uses frequencies.

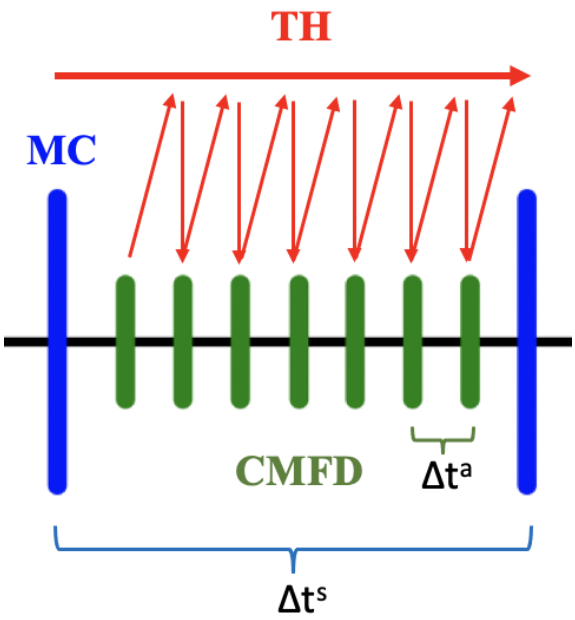

Figure 4-12: Time stepping illustration of the TD- TH/CMFD procedure with the tally derivatives.

This scheme can become more efficient if TD-CMFD does not rely on knowing the forward outer time step to estimate the cross sections. In fact, with the addition of the thermal solver, cross section changes can be estimated thanks to differential tallies (or tally derivatives). With cross section estimates at every inner time step, Figure 4-10 becomes Figure 4-12 where the changes from TD-TH can impact the cross sections in TD-CMFD. As a consequence, the frequencies can be calculated for the omega method the first time the forward Monte Carlo run is performed, potentially reducing the number of required Monte Carlo calculations.

4.2.1 Tally derivative example (single group)

Tally derivatives are a first-order perturbation technique that estimates how a Monte Carlo tally would change based on a perturbation in the material properties. A full derivation of tally derivatives and details of their implementation in OpenMC can be found in Reference [21].

To illustrate the basic function of a tally derivative, an infinite cylinder of fuel (2.4% enrichment with a helium gap and zirconium cladding surrounded by non-

borated water and reflective boundary conditions) is run in several states:

1. At a temperature of 800 K with a water density of 0.70 g/cc
2. At a temperature of 900 K with a water density of 0.70 g/cc
3. At a temperature of 800 K with a water density of 0.75 g/cc

The temperature of the water is held constant at 500 K, cladding at 600 K, and gap at 700 K. Given the reflective nature of this problem and the normalization of Monte Carlo tallies to the absorption rate, the k-inf eigenvalue of this problem is directly proportional to the nu-fission tally. Therefore, the tally derivative of the nu-fission tally can be used to estimate k-inf in these three states. Table 4.1 reports the tallied k-inf values and tally derivatives for each state with respect to fuel temperature (T) and water density (ρ). Tables 4.2, 4.3, and 4.4 show the absolute difference between states and the absolute error with which k-inf is predicted. These results show that k-inf changes of up to 1000 pcm can be accurately predicted within 71 pcm absolute error using tally derivatives.

	k-inf	$\frac{d}{dT}$	$\frac{d}{d\rho}$
State 1	1.27920	-3.29E-5	0.20
State 2	1.27647	-3.09E-5	0.20
State 3	1.28921	-3.19E-5	0.186

Table 4.1: Tally derivative values for each state.

	Difference (Δk) from state 1	Prediction error (absolute)
State 1	0 pcm	N/A
State 2	-273 pcm	-56 pcm
State 3	1001 pcm	-1 pcm

Table 4.2: Predicting k-eff with tally derivatives from state 1

	Difference (Δk) from state 2	Prediction error (absolute)
State 1	273 pcm	36 pcm
State 2	0 pcm	N/A
State 3	1274 pcm	35 pcm

Table 4.3: Predicting k-eff with tally derivatives from state 2

	Difference (Δk) from state 3	Prediction error (absolute)
State 1	-1001 pcm	71 pcm
State 2	-1274 pcm	25 pcm
State 3	0 pcm	N/A

Table 4.4: Predicting k-eff with tally derivatives from state 3

4.2.2 Cross section derivative example (2-group)

A tally is a reaction rate, or the product of a cross section with the flux. Therefore, the tally derivative estimates the change in that reaction rate. To extract the cross section derivative, the chain rule must be applied:

$$\frac{d\Sigma\Psi}{d\theta} = \Sigma\frac{d\Psi}{d\theta} + \Psi\frac{d\Sigma}{d\theta} \tag{4.9}$$

$$\implies \frac{d\Sigma}{d\theta} = \Big(\frac{d\Sigma\psi}{d\theta} - \Sigma\frac{d\psi}{d\theta}\Big)\frac{1}{\psi} \tag{4.10}$$

where Σ is the multigroup cross section, ψ is the tallied flux, and θ is the material property being perturbed. With the tallied values for cross section, flux, tally derivative, and flux derivative, a cross section derivative can then be determined using Equation (4.10) if the change in material property is known. In the following example, 2-group k-inf values can be predicted using cross section derivatives for a water density change. The geometry is identical to the infinite cylinder of fuel described

above. Although no boron is present in the water used here, the same example was also executed with boron, with similar results. The equation used for k-inf is:

$$\text{k-inf} = \frac{\nu\Sigma_{f_1} + \Sigma_{s_{1\to2}}\nu\Sigma_{f_2}/(\Sigma_{t_2} - \Sigma_{s_{2\to2}})}{\Sigma_{t_1} - \Sigma_{s_{1\to1}} - \Sigma_{s_{1\to2}}\Sigma_{s_{2\to1}}/(\Sigma_{t_2} - \Sigma_{s_{2\to2}})} \tag{4.11}$$

Tally derivatives are not available for all tallies. In particular, "differential material reaction rates of perturbed materials cannot be computed with outgoing energy or outgoing angle dependence" as explained in [100]. Therefore, group-to-group scattering tally derivatives are not available. The nu-scatter matrix prediction must rely on the tally derivative of the scatter rate tally alone. There are several options to predict the proportion of up-scatter and down-scatter in the perturbed state. Firstly, each component of the scatter matrix could be predicted to evolve according to the scatter cross section derivative. However, this does not conserve the total scatter cross section prediction. Secondly, the total scatter cross section can be predicted, and then the proportion of up- and down- scattering can be assumed to remain constant. However, this is not an accurate assumption, as the proportion changes with water density. The third option which is used in this work, prioritizes the correct prediction of down-scattering because it is the most significant effect on the spectrum. The total scattering cross section in each energy group is predicted using a cross section derivative. The down-scatter term is approximated to scale with the change in density, as:

$$\Sigma_{s_{1\to2},\text{predicted}} = \Sigma_{s_{1\to2},\text{old density}} \times \frac{\text{new density}}{\text{old density}} \tag{4.12}$$

Then, the predicted fast-to-fast group scattering can be back-computed as:

$$\Sigma_{s_{1\to1},\text{predicted}} = \Sigma_{s_1,\text{predicted}} - \Sigma_{s_{1\to2},\text{predicted}} \tag{4.13}$$

In the thermal group, the up-scattering is approximated as remaining unchanged because it is the smallest effect, so its change carries relatively low impact. The thermal-to-thermal group scattering can be similarly back-computed from the total predicted thermal group scattering.

An example of a water density change of 0.05 g/cc incurs a reactivity change of approximately 700 pcm. k_{inf} was predicted to within 160 pcm. The contributions of each cross section prediction to the overall error are shown in Tables 4.5 and 4.6, where the down-scatter prediction is shown to carry most of the error.

	Tallied at 0.70 g/cc and relative error	Predicted at 0.70 g/cc from 0.75 g/cc	Contribution to k-inf error [pcm]
nu-fission (1)	0.00583 +/- 6pcm	0.00584	15
nu-fission (2)	0.11675 +/- 10pcm	0.11680	37
absorption* (1)	0.00898	0.00898	-22
absorption* (2)	0.07199	0.07201	-32
nu-scatter (1→2)	0.01715 +/- 8pcm	0.01731	158
nu-scatter (2→1)	0.00108 +/- 98pcm	0.00107	2
k-inf (calculated)	1.28464	1.28667	157

Table 4.5: Predicted cross sections in infinite cylinder for a water density of 0.70g/cc based on cross section derivatives tallied and calculated at 0.75g/cc.
*absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$

	Tallied at 0.75 g/cc and relative error	Predicted at 0.75 g/cc from 0.70 g/cc	Contribution to k-inf error [pcm]
nu-fission (1)	0.00588 +/- 5pcm	0.00588	-1
nu-fission (2)	0.11767 +/- 10pcm	0.11772	35
absorption* (1)	0.00907	0.00907	-7
absorption* (2)	0.07293	1.07296	-30
nu-scatter (1→2)	0.01855 +/- 8pcm	0.01838	-152
nu-scatter (2→1)	0.00107 +/- 95pcm	0.00108	-2
k-inf (calculated)	1.29365	1.29161	-157

Table 4.6: Predicted cross sections in infinite cylinder for a water density of 0.75g/cc based on cross section derivatives tallied and calculated at 0.70g/cc.
*absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$

A similar test was conducted for temperature derivatives with a 100 K change between two states. Since changes in the scattering matrix are dominated by change in water density and not fuel temperature, the scattering matrix is approximated as being unchanged between the original state and the perturbed state. The reactivity change is 270 pcm, and k_{inf} is predicted within 115 pcm. The contributions of each cross section prediction to the overall error are shown in Tables 4.7 and 4.8, where

the thermal absorption and thermal fission prediction are now shown to carry most of the error.

	Tallied at 800 K and relative error	Predicted at 800 K from 900 K	Contribution to k-inf error [pcm]
nu-fission (1)	0.00583 +/- 6pcm	0.00584	2
nu-fission (2)	0.11675 +/- 10pcm	0.11658	-116
absorption* (1)	0.00898	0.00899	-60
absorption* (2)	0.07199	0.07190	98
nu-scatter (1→2)	0.01715 +/- 8pcm	0.01712	-29
nu-scatter (2→1)	0.00108 +/- 98pcm	0.00112	-10
k-inf (calculated)	1.28464	1.28316	-115

Table 4.7: Predicted cross sections in infinite cylinder for a fuel temperature of 800 K based on cross section derivatives tallied and calculated at 900 K.
*absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$

	Tallied at 900 K and relative error	Predicted at 900 K from 800 K	Contribution to k-inf error [pcm]
nu-fission (1)	0.00583 +/- 6pcm	0.00583	-2
nu-fission (2)	0.11658 +/- 10pcm	0.11675	116
absorption* (1)	0.00903	0.00902	49
absorption* (2)	0.07189	1.07197	-93
nu-scatter (1→2)	0.01712 +/- 8pcm	0.01715	30
nu-scatter (2→1)	0.00112 +/- 90pcm	0.00108	10
k-inf (calculated)	1.28120	1.28261	110

Table 4.8: Predicted cross sections in infinite cylinder for a fuel temperature of 900 K based on cross section derivatives tallied and calculated at 800 K.
*absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$

The full analysis of the root cause of error in thermal group tally derivatives is out of the scope of this thesis. In fact, temperature derivatives are most important in the prediction of Doppler feedback for which the fast energy absorption cross section is the main agent of change. This is because cross section resonances, which are very narrow, widen at higher temperatures. This resonance broadening in uranium results in an increase in absorption in the resonance region and a subsequent decrease in thermal fission, thus provoking a negative feedback effect. A good prediction in $\Sigma_{a,1}$ is expected to produce a good prediction of the Doppler effect in a transient. Furthermore, the error in the thermal absorption cross section and the fission cross section

are nearly equal and opposite, as fission is an absorption reaction that contributes oppositely to the balance equation. Therefore, their errors compete and become small when taken together.

As a summary of the impact of tally derivatives on the accurate prediction of reactivity changes in a transient scenario, Table 4.9 shows how each cross section error impacts the prediction of reactivity in each case studied in this section. Each entry is the cross section's contribution to k-inf error in each case, from Table 4.5 to 4.8, divided by the reactivity change of that case. For a density perturbation (the first and second columns of Table 4.9), the total error in reactivity change prediction is 22%, concentrated almost entirely in the down-scattering term. For a temperature perturbation (the third and fourth columns), the error in reactivity change prediction is about 40%, with thermal fission and thermal absorption errors canceling each other out, leaving the bulk of the error carried by fast absorption. In both cases, the error is concentrated in the cross section that is most impactful for that perturbation. While far from a high-fidelity method, the magnitudes of these errors suggest that cross section derivatives are a good tool for first-order prediction of transient changes given no other information than the initial cross sections and the change in material properties.

	0.7 g/cc from 0.75 g/cc	0.75 g/cc from 0.7 g/cc	800 K from 900 K	900 K from 800 K
nu-fission (1)	2%	-0%	-1%	1%
nu-fission (2)	5%	5%	43%	-43%
absorption* (1)	-3%	-1%	22%	-18%
absorption* (2)	-5%	-4%	-37%	35%
nu-scatter (1→2)	23%	-22%	11%	-11%
nu-scatter (2→1)	0%	-0%	4%	-4%

Table 4.9: Impact of cross section prediction error on total reactivity change prediction.
*absorption: defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$

4.2.3 Derivatives for TD-CMFD

Leveraging cross section derivatives, it is possible to create a more efficient solution to the transient scheme as shown in Figure 4-13. This new scheme does not fundamentally vary from Figure 4-11, but the first TD- TH/CMFD solution includes tally derivatives. In this way, the "zeroth iteration" of TD-CMFD has information about how the cross sections are changing before the forward outer Monte Carlo calculation has been run. This implies that a first-order predication of power over the inner time steps can take place, and that the frequencies from TD-CMFD are available when the forward outer step is first calculated.

In this implementation, fission (applied to prompt-nu-fission, delayed-nu-fission, and kappa-fission) and total tallies are modified with fuel temperature and water density tally derivatives, whereas scattering is modified only with water density tally derivatives.

The fuel temperature tally derivative results in a cell-by-cell derivative of the cross section derivative. This is in contrast to the water density tally derivative which results in an array of each cell's effect on all other cells as explained in [21]. Each cell has a derivative that affects itself most strongly but also surrounding cells with an opposite and diminishing effect. Figure 4-14 shows the structure of a fuel temperature tally derivative which is in contrast to Figure 4-15 showing how a single cell's water density derivative has neighboring effects (only a few cells shown for readability). Figures 4-16 and 4-17 show the complete water density derivative structure.

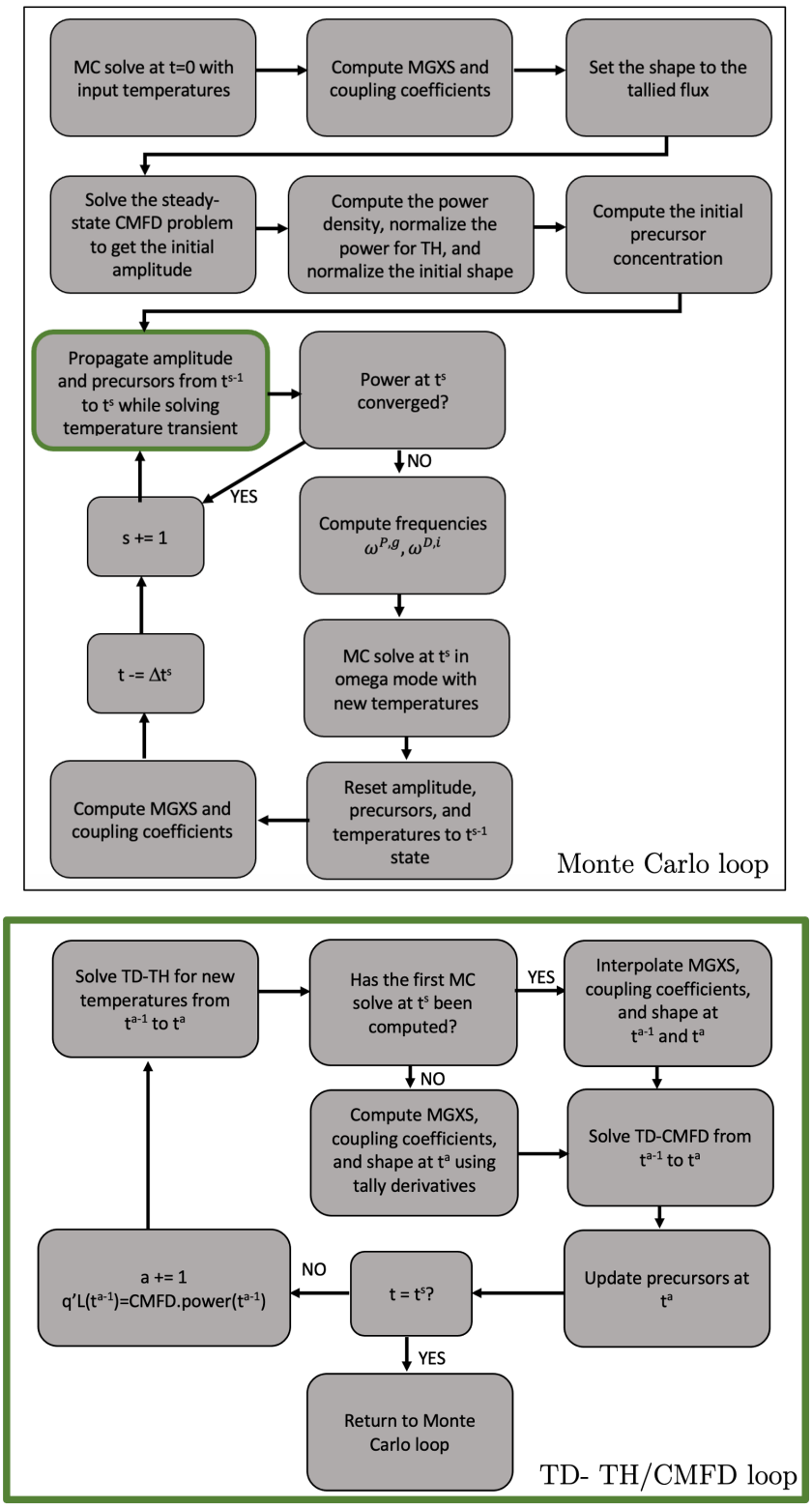

Figure 4-13: Flowchart of modified solution procedure with tally derivatives.

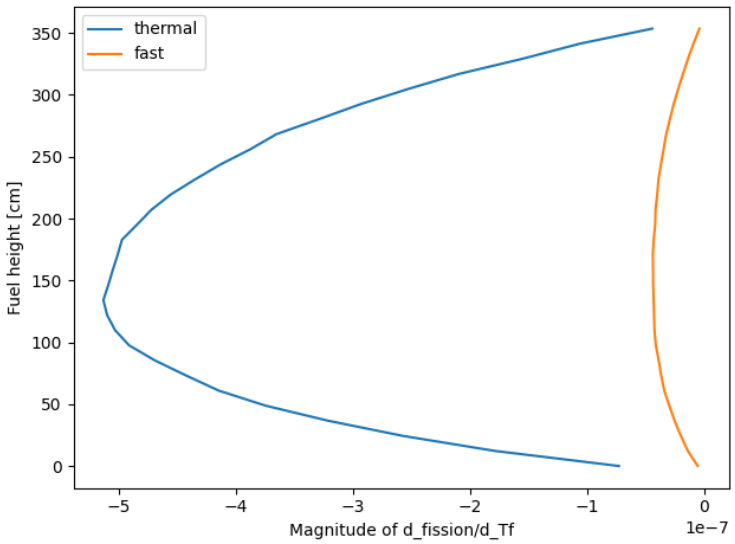

Figure 4-14: A representative example of fuel temperature derivatives for a fission rate tally in a 1-D fuel rod geometry. The x-axis units are reactions per source particle per degree Kelvin.

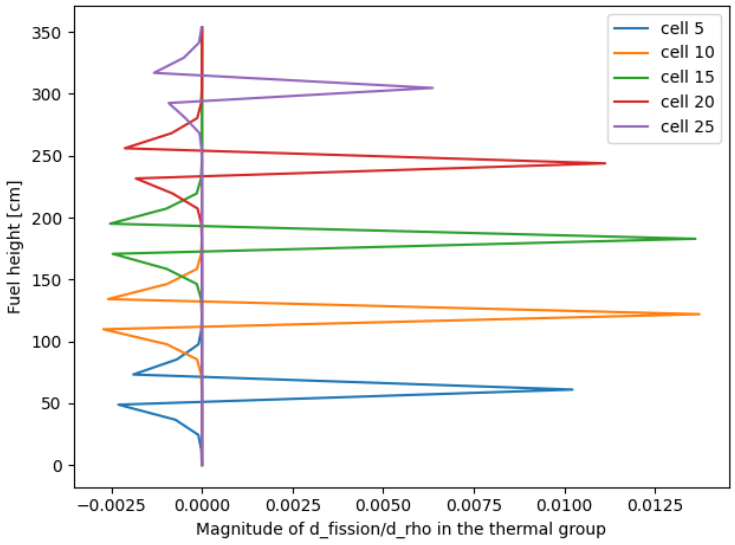

Figure 4-15: A representative example of water density derivatives for a fission rate tally for select cells in a 1-D fuel rod geometry. The x-axis units are reactions per source particle per g/cc.

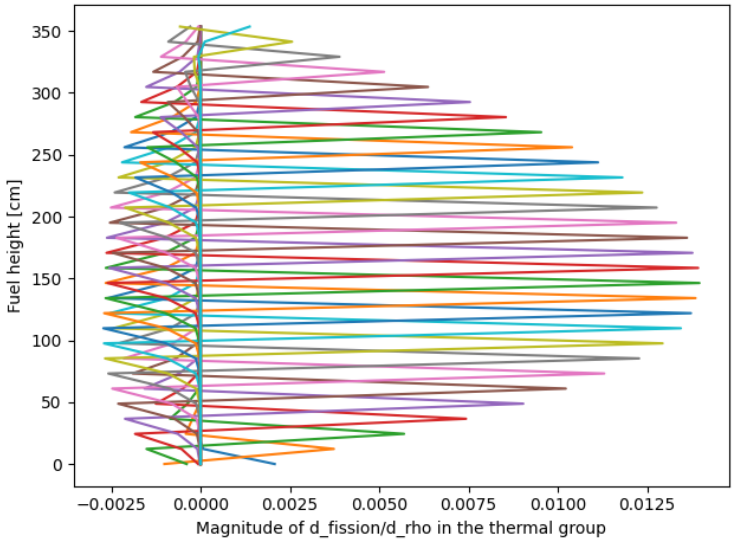

Figure 4-16: A representative example of water density derivatives for the thermal group for a fission rate tally for all cells in a 1-D fuel rod geometry. The x-axis units are reactions per source particle per g/cc.

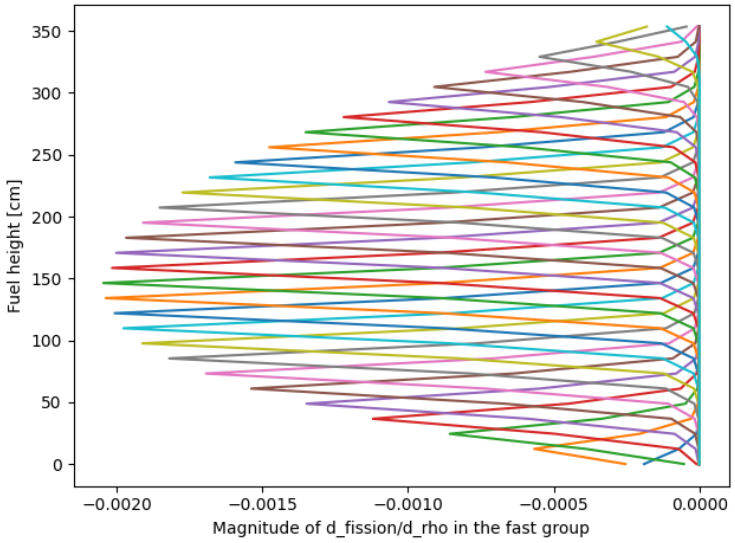

Figure 4-17: A representative example of water density derivatives for the fast group for a fission rate tally for all cells in a 1-D fuel rod geometry. The x-axis units are reactions per source particle per g/cc.

A rigorous implementation should use a matrix multiplication to account for each

cell's density change on all other cells, as illustrated below for a 3 cell example.

$$\begin{bmatrix} \frac{\partial P_1}{\partial \rho_1} & \frac{\partial P_1}{\partial \rho_2} & \frac{\partial P_1}{\partial \rho_3} \\ \frac{\partial P_2}{\partial \rho_1} & \frac{\partial P_2}{\partial \rho_2} & \frac{\partial P_2}{\partial \rho_3} \\ \frac{\partial P_3}{\partial \rho_1} & \frac{\partial P_3}{\partial \rho_2} & \frac{\partial P_3}{\partial \rho_3} \end{bmatrix} \cdot \begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \Delta\rho_3 \end{bmatrix} = \begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \Delta P_3 \end{bmatrix} \tag{4.14}$$

As a simplification, this work takes advantage of the fact that each cell has the largest effect on itself ($\frac{\partial P_i}{\partial \rho_i}$) and diminishing and opposite effects on surrounding cells the further away they are ($\frac{\partial P_i}{\partial \rho_j}$). Therefore, when the water density tally derivative rows are summed together, the density change in the corresponding cell is approximated to dominate the effect, adjusted for the small effect of other cells:

$$\frac{\partial P_1}{\partial \rho *} = \frac{\partial P_1}{\partial \rho_1} + \frac{\partial P_1}{\partial \rho_2} + \frac{\partial P_1}{\partial \rho_3} \tag{4.15}$$

$$\implies \frac{\partial P_1}{\partial \rho *} \times \Delta\rho_1 \approx \Delta P_1 \tag{4.16}$$

The density derivative is tallied the same way but processed with this approximation to yield a derivative that more closely resembles the fuel derivative format as shown in Figure 4-18.

In addition to cross sections, flux is modified by tally derivatives and then used in the estimate of those cross sections. Diffusion coefficients are also estimated by tally derivatives, as follows:

$$\frac{d(1/D)}{d\theta} = 3\frac{d\Sigma_t}{d\theta} \tag{4.17}$$

$$\implies D = 1/\Big(\frac{1}{D} + \frac{d(1/D)}{dT_f} \times \Delta T_f + \frac{d(1/D)}{d\rho_w} \times \Delta\rho_w\Big) \tag{4.18}$$

During the inner time steps, the flux and diffusion coefficients are modified by the tally derivatives and the coupling terms $\tilde{D}$ and $\hat{D}$ of Equations (2.2) and (2.4) are recalculated. However, no tally derivative for the currents has been implemented, which

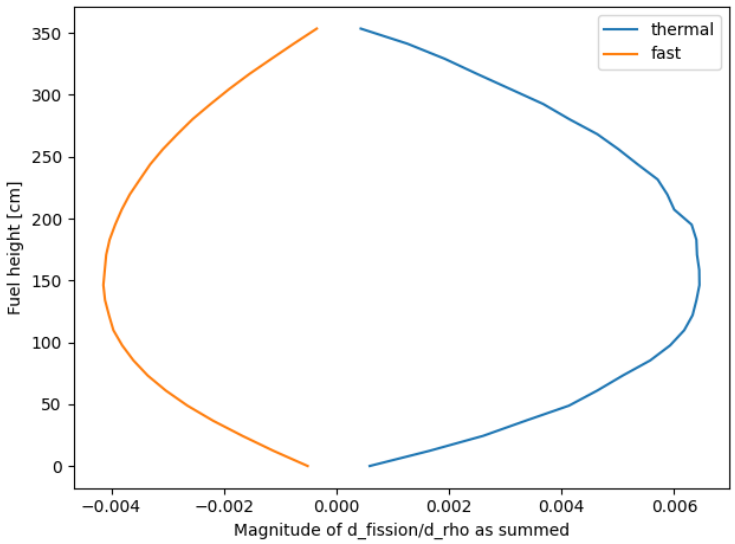

Figure 4-18: A representative example of water density derivatives for a fission rate tally in a 1-D fuel rod geometry as summed by approximation in Equation (4.15). The x-axis units are reactions per source particle per g/cc.

introduces an inconsistency in the method. The χ^P and χ^D values are not modified by tally derivatives as they are not expected to change in a 2-group problem. The decay constant and the inverse velocity are also untouched.

To test the use of tally derivatives in transient scenarios, prescribed changes in fuel temperature and water density are imposed on a single-cell, fully reflective fuel rod geometry, but cut short to be only 10 cm in length to speed up calculations. To impose the prescribed change, the transient thermal solver was bypassed in favor of an enforced change of temperature or density:

1. A linear increase in fuel temperature of 20 K over the course of 0.5 seconds with no other changes in material properties.
2. A linear decrease in water density of 0.007 g/cm^3 over 0.5 seconds with no other changes in material properties.

Case 1 was found to have a reactivity feedback coefficient of -1.75 pcm/K, and case 2 had a reactivity feedback coefficient of 2.5 pcm/(kg/m^3). While not exact matches to the true expected feedback coefficients, these results are in keeping with the errors identified in the cross section analysis of Section 4.2.2.

Chapter 5

Transient Implementation

5.1 Fuel Rod Geometry

With the methods presented in Chapter 4, a fuel rod geometry serves as the initial testbed for convergence criteria and the impact of different run parameters. This geometry features a 365.76 cm rod of fresh UO2 fuel at 2.4% enrichment, with a radius of 0.39218 cm, surrounded by a helium gap of 0.00787cm, and zirconium cladding of 0.05715 cm. This rod is surrounded by borated water with 432.473 ppm boron in a pitch of 1.25984 cm. Axially, there are no spacer grids, and the rod is divided into 30 cells of equal length. Each cell contains the same fuel, gap, and zirconium materials but each material in each cell has different temperatures. Water is a separate material in every cell so that density, in addition to temperature, can be modified independently in each cell. The top and bottom of the fuel rod are immediately truncated by vacuum boundary conditions (there are no end plugs or water reflectors). Radially, the outer pitch of the geometry has reflective boundary conditions.

Temperature treatment in OpenMC uses windowed-multipole data [28] between 290 and 2500 K in the resonance region, and interpolation is used where windowed-multipole data is not available (that is, linear-linear stochastic interpolation between temperatures at which nuclear data are present [22]). The tally mesh is identical to the cell mesh (30 axial cells). All tallies are in 2 energy groups, except for the transport cross section which is tallied in the CASMO 40 energy group structure

using the flux-limited approximation [101]. Once diffusion coefficients are calculated as $\frac{1}{3\Sigma_{tr}}$, the diffusion coefficients are condensed to 2 groups as recommended in [102]. Additionally, 6 delayed neutron precursor groups are used, as done in [76], from the ENDF/B-VII.1 library.

The thermal fluids parameters are detailed in sections 2.3.1 and 4.1. The mass flux is 3127 kg/m^2/s, the pressure is 15.51 MPa, and the water inlet temperature is 566.25 K as specified in [44].

5.1.1 Null transient considerations

In order to successfully hold a null transient using this transient scheme, each numerical component needs to be converged. This includes:

1. The thermal fluids solver must be sufficiently converged to avoid substantial temperature changes between outer time steps.
2. The Monte Carlo particle convergence must be sufficiently high to avoid flux shape fluctuations between outer time steps.
3. The tally derivatives must be tallied to tight enough statistics for accuracy.

Success in holding the null transient is not entirely possible because of the stochastic nature of the underlying Monte Carlo solver that provides tallies, cross sections, and temperature distributions. The fluctuations in achieving a null transient will provide an error estimate associated with the stochastic nature of the problem, within practical particle count limits. These fluctuations are quantified by the change in normalized power density as well as the change in axial power distribution at every outer time step from the initial power distribution that generated the starting temperature distribution.

The temporal convergence of the thermal fluids solver was studied in Section 4.1. In accordance with those results, inner time steps of 0.0025 seconds are used for thermal fluids and TD-CMFD. The spatial convergence of the thermal fluids solver was determined in [31, 21].

To test the Monte Carlo convergence threshold needed for the transient scheme, a single outer step of 0.5 seconds is run in a null transient scenario (with Monte Carlo shape calculations at 0 and 0.5 seconds) with increasing numbers of simulated particles per batch: 50,000, 100,000, 200,000, 400,000, and 800,000 particles per batch (each with an unchanging number of 1100 batches which includes 100 inactive batches). Every case starts with the same temperature profile and converges the steady-state Monte Carlo solution with the specified number of particles per batch. This is done because the stochastic convergence is assessed based on how different the previous outer and forward outer Monte Carlo calculations are from each other. If using the same initial solution in all cases, the variability between the two Monte Carlo calculations cannot be fully known, as one of the stochastic components would be fixed. These tests use an imposed number of iterations on the outer time steps instead of a convergence criteria: each outer time step is repeated five times. This gives five samples of the stochastic fluctuations from the true null transient result, giving a better picture of the error associated with simulated particle numbers. In the next section, convergence criteria will be used to determine how many iterations are needed in different run conditions.

Each of the five iterations consists of a previous outer Monte Carlo run at t=0s, a TD- TH/CMFD calculation to estimate temperatures at t=0.5s, and a forward outer Monte Carlo run at t=0.5s. The lines in Figure 5-1 show the CMFD power solution for the inner time step. The "zeroth iteration" represents the first TD- TH/CMFD calculation, before the first forward outer Monte Carlo calculation. Without information about the forward outer time step to use in TD-CMFD cross section interpolation, and with the null transient conditions in the TD-TH, the solution of the zeroth iteration is exactly flat. All the following iterations use the forward outer Monte Carlo calculation for cross section interpolation, thereby fluctuating in accordance with the statistical differences between the previous and forward outer Monte Carlo calculations. Figure 5-2 shows the range of fluctuations in power at 0.5 seconds for each particle count for the null transient. The same figure is presented with a prescribed null transient in Figure 5-3, where the temperatures are forced to be constant. This

removes the error associated with the TD-TH solver and purely identifies the convergence of the neutronics.

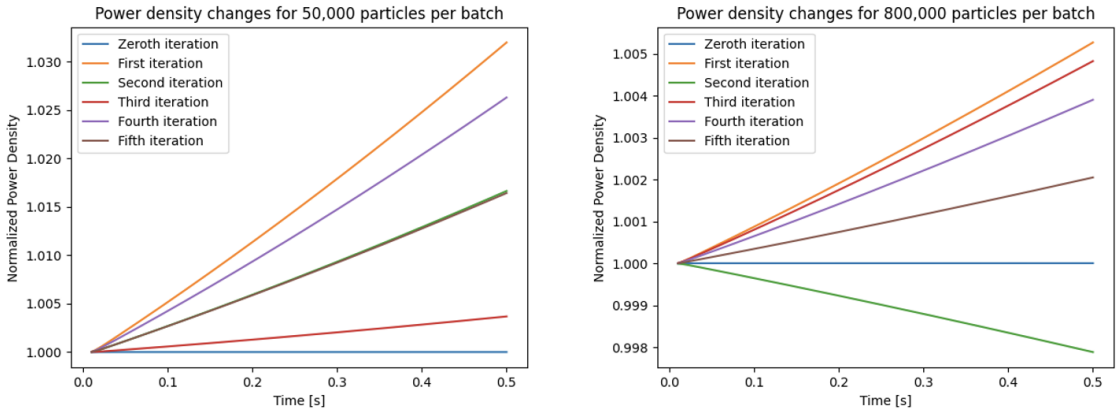

Figure 5-1: Fluctuations in normalized power density [W/cc] during a null transient. The left-hand figure uses 50,000 particles per batch; the right-hand figure uses 800,000 particles per batch.

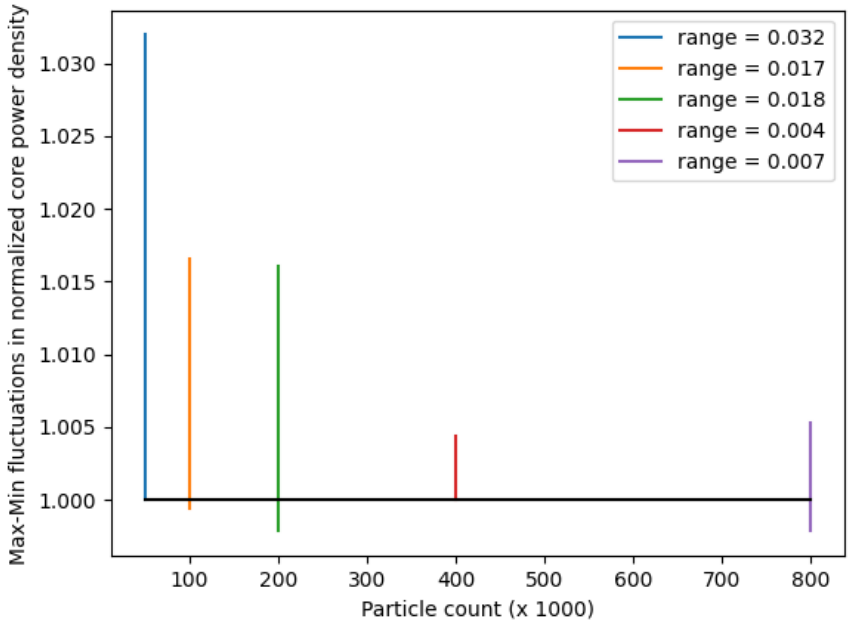

Figure 5-2: Fluctuations in normalized power density [W/cc] during a null transient. The particle count represents the number of particles per batch.

These results highlight the negligible error associated with the TD-TH solver, as the prescribed transient fluctuations are similar to the non-prescribed transient.

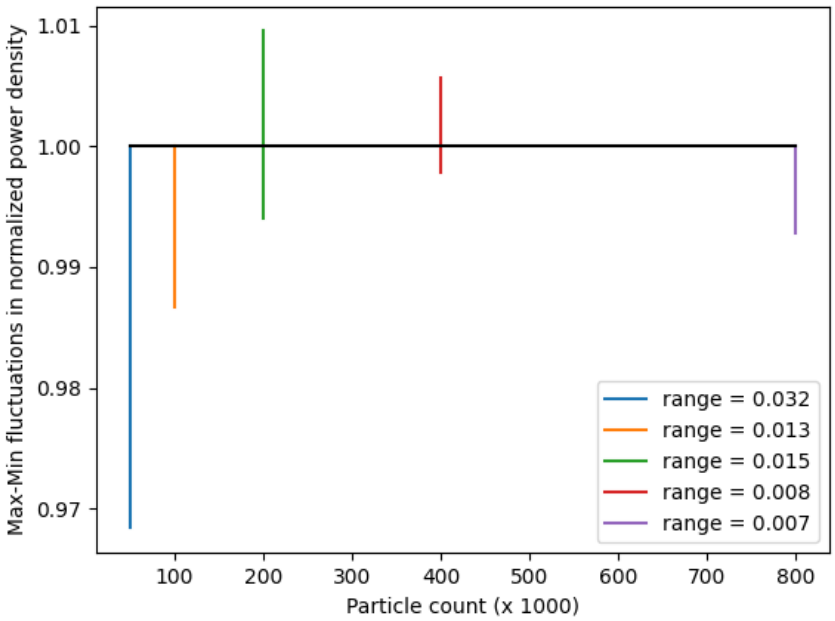

Figure 5-3: Fluctuations in normalized power density [W/cc] during a prescribed null transient. The particle count represents the number of particles per batch.

Furthermore, as the particle count increases, the fluctuations of the null transient decrease, but 800,000 particles per batch does not appear to improve upon 400,000 particles despite twice the cost.

This study concludes that 400,000 particles is sufficient to statistically converge the null transient. An error of up to 0.5% in normalized power level is attributed to the particle count based on Figure 5-2. With these parameters, the kappa-fission tally carries an average relative standard deviation of 6.2×10^{-4}. The fuel temperature tally derivative of the fission tally carries an average relative standard deviation of 9.9×10^{-3}, and the water density tally derivative of the fission tally carries an average relative standard deviation of 3.1×10^{-1}. However, many of the values of the water density tally derivatives are very near zero. Taking into account only the values greater than 10^{-6}, the average relative standard deviation is 8.8×10^{-3}.

5.1.2 Inlet temperature transient

Using 400,000 particles for the fuel rod geometry, a transient is now conducted to determine temporal convergence. The transient is an instantaneous increase in inlet coolant temperature from 566.25 K to 570 K at $t = 0^+$. The coolant temperature solution for the first 0.5 seconds is shown in Figure 4-5, where the advection wave is seen traveling up the rod in time. Inner time steps of 0.0025 seconds are used. As shown, the change does not reach the top of the fuel rod right away. This transient introduces an axial change in the temperature distribution over time which changes the neutron distribution and the overall power. Included here are plots of the time dependent coolant and fuel temperatures to give an idea of the transient progression. These use 0.5 second outer time steps, 3 imposed iterations, a linear-in-time cross section interpolation, and no tally derivatives (these parameters will each be tested in the following subsections). As seen in Figures 5-4 and 5-5, the coolant temperature undergoes a shape shift that stabilizes after 1 second while the fuel temperature steadily decreases with more spatial uniformity.

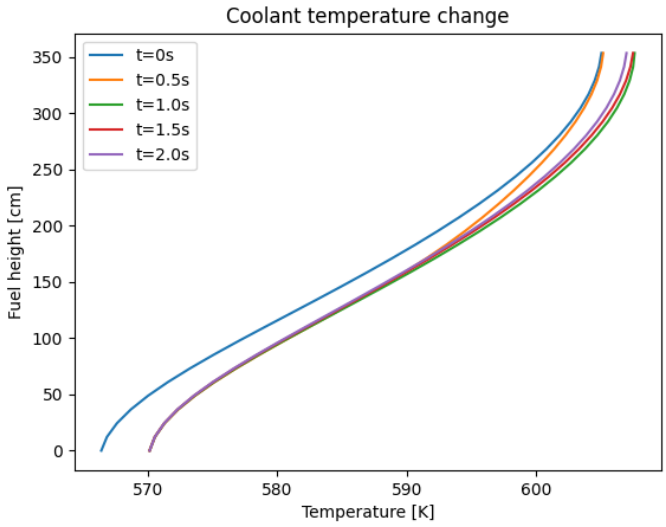

Figure 5-4: Time dependent progression of the coolant temperature during the inlet temperature transient.

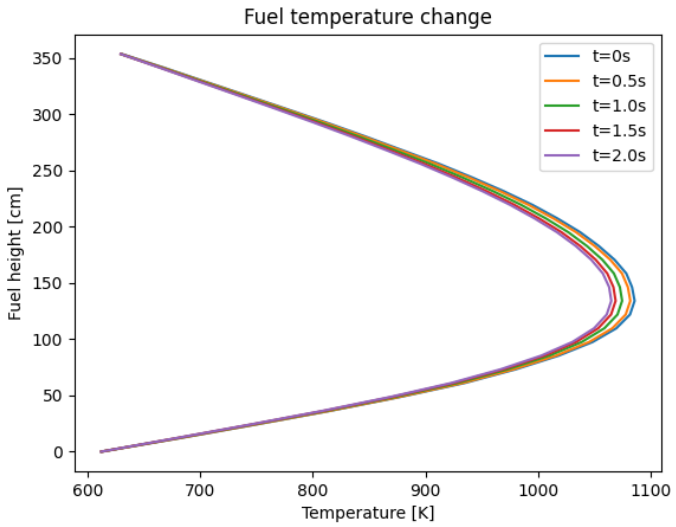

Figure 5-5: Time dependent progression of the fuel temperature during the inlet temperature transient.

Temporal convergence and interpolation schemes

Firstly, the temporal convergence is evaluated. The goal is to determine the limit for the outer time step length for this particular transient given the approximations of these methods, using a fixed number of 2 iterations per time step. Using a linear-in-time weighted interpolation of cross sections, according to Equation (4.7), results using 0.5s, 0.25s, and 0.125s outer time step sizes are run where some thermal feedback appears to come into play in the second half of the transient, but the solution is very poorly resolved, even with the smallest outer time step.

From Figure 5-6, it is seen that the finer time step solutions offer more resolution of the transient, but the coarser time step solutions agree at several of the shared outer time steps where synchronization with MC is performed. This suggests that the CMFD solver cannot capture the inner shape of the rapidly changing flux, especially in the first second of the transient. This is most likely caused by a poor interpolation assumption of the CMFD parameters. This is also a possible explanation for the choppiness of the solution between outer time steps. The TD-CMFD solver relies on cross sections at the previous and forward Monte Carlo time step to estimate

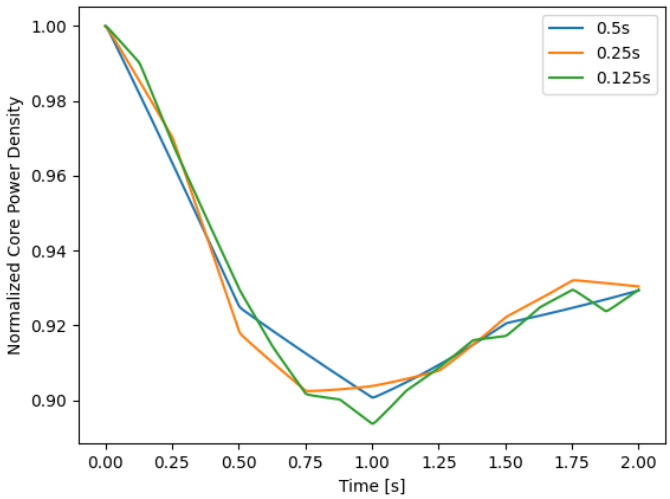

Figure 5-6: Time dependent normalized power density [W/cc] during an inlet temperature transient with varying outer time step sizes.

the cross sections during the inner time steps. As explained in 4.1.3, the linear-in-time interpolation causes all cross sections in the geometry to advance at the same rate in every cell without accounting for the axially different rates of the transient progression. A comparison of the cross section evolution in this transient during a 0.5 second outer time step using time weighting and density weighting (as described in 4.1.3) compared to a reference solution shows the significant improvement in cross section interpolation. In Figures 5-7, 5-8, and 5-9, the reference consists of time weighting interpolation on fine outer time steps of 0.1 seconds, such that the evolution of cross sections during the inner time steps can be observed. The cross sections shown are the total cross section, the thermal absorption cross section (defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$; fast absorption is a small effect and results contain statistical noise), and the downscattering cross section. Each is shown for a cell 1/3 from the bottom of the rod and 2/3 from the bottom of the rod.

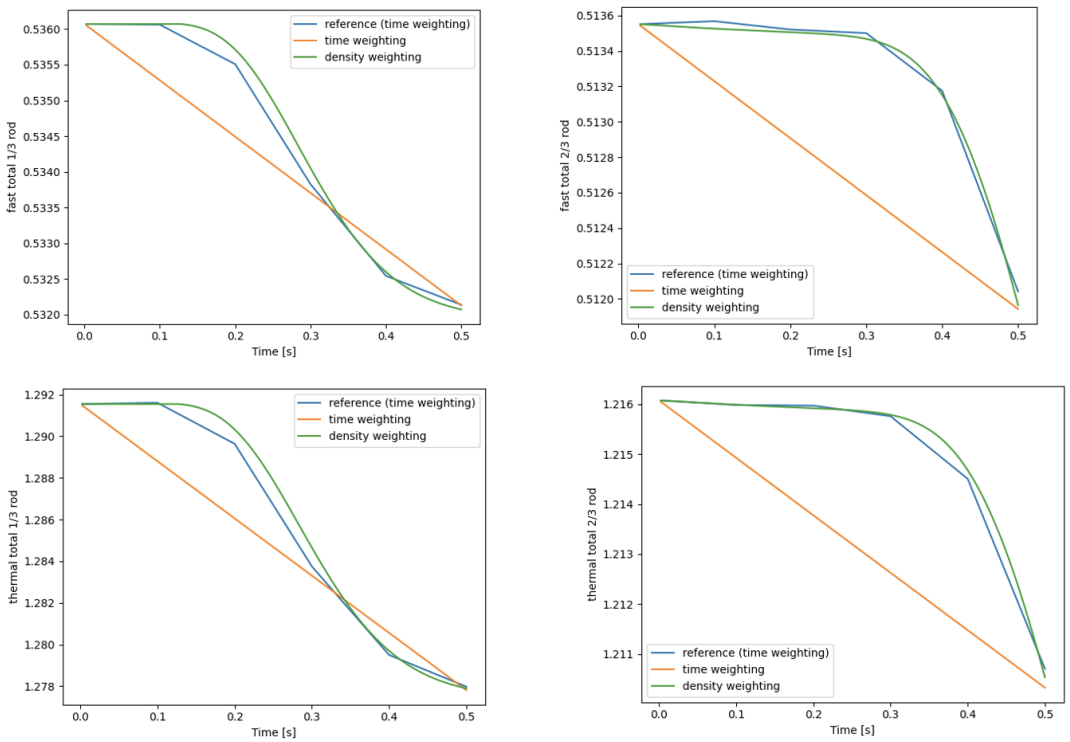

Figure 5-7: Evolution of the total cross section during the first outer time step under two types of weighting compared to a reference solution. Fast cross section 1/3 from the bottom of the rod (top left), fast cross section 2/3 from the bottom of the rod (top right), thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right).

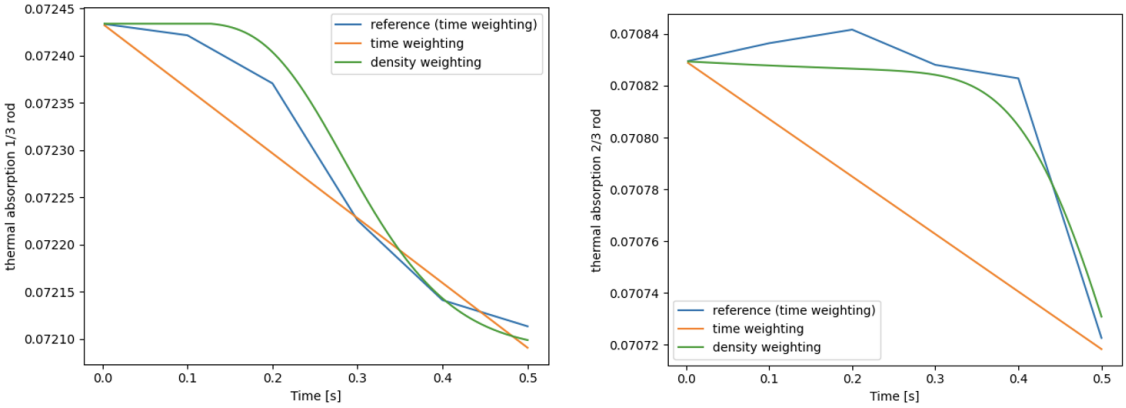

Figure 5-8: Evolution of the thermal absorption cross section (defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$) during the first outer time step under two types of weighting compared to a reference solution. Thermal cross section 1/3 from the bottom of the rod (left), and thermal cross section 2/3 from the bottom of the rod (right).

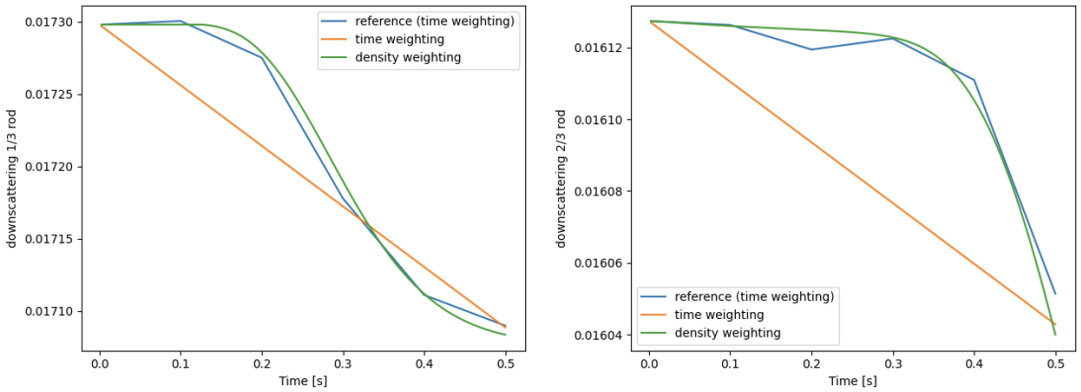

Figure 5-9: Evolution of the downscattering cross section during the first outer time step under two types of weighting compared to a reference solution. Location 1/3 from the bottom of the rod (left), and location 2/3 from the bottom of the rod (right).

Based on the results in these figures, density weighting the cross section interpolation is a significantly better approximation than time weighting in this example. However, looking at the evolution of the cross sections over all time steps reveals that there are some problematic behaviors that emerge between time steps, seen most sharply at 0.5 seconds in the cross sections 1/3 from the bottom of the rod in Figures 5-10, 5-11, and 5-12. An intermediate approach, noted "mixed weighting" (as described in 4.1.3), shows similar improvement in cross section interpolation while mitigating the discontinuities.

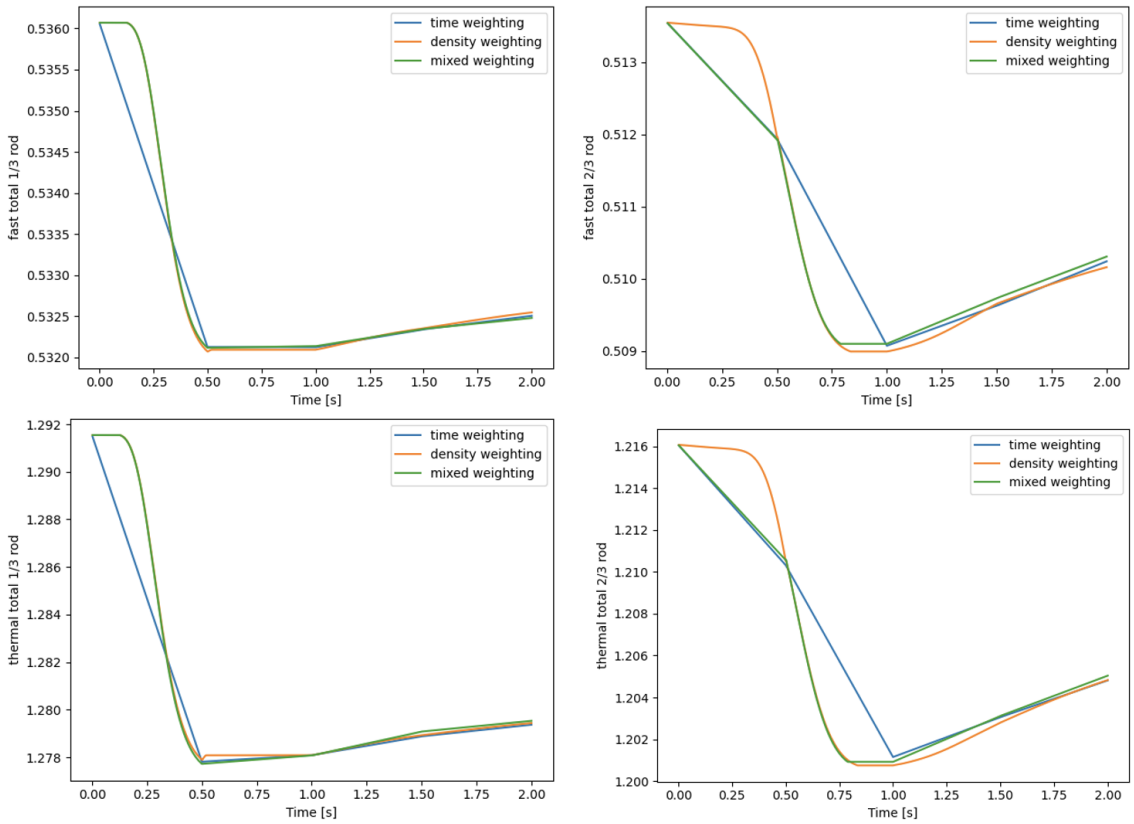

Figure 5-10: Evolution of the total cross section during the inner time steps under three types of weighting. Fast cross section 1/3 from the bottom of the rod (top left), fast cross section 2/3 from the bottom of the rod (top right), thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right).

Figures 5-10, 5-11, and 5-12 illustrate the differences between time weighting, density weighting, and mixed weighting. With purely time weighting, the cross section is assumed to change linearly in time between its previous outer time step value and

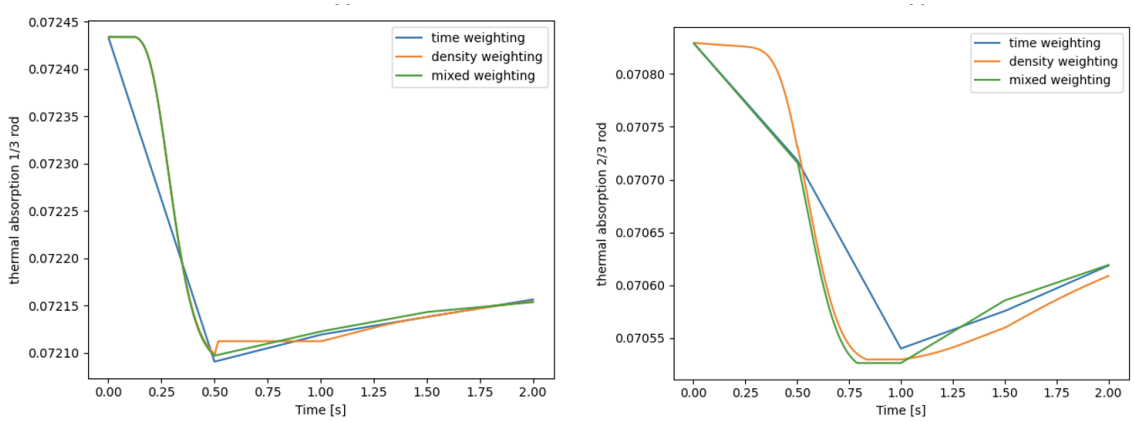

Figure 5-11: Evolution of the thermal absorption cross section (defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$) during the inner time steps under three types of weighting. Thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right).

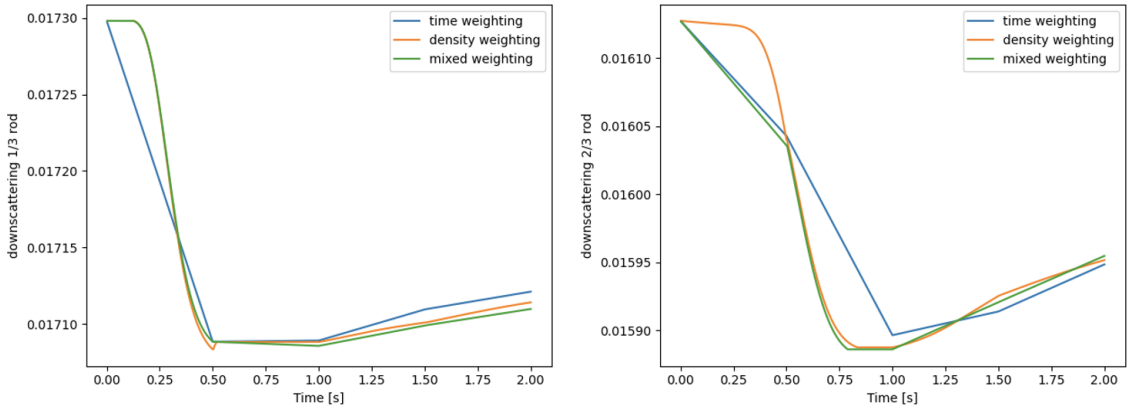

Figure 5-12: Evolution of the downscattering cross section during the inner time steps under three types of weighting. Location 1/3 from the bottom of the rod (left), and location 2/3 from the bottom of the rod (right).

its forward outer time step value. However, because the transient progresses up the rod in time, this is not an accurate representation of the cross section evolution. The density weighting allows the cross sections of a particular cell not to change until the transient reaches that cell. However, after the density in the cell stops changing, there are continued cross section changes that occur that the density weighting scheme cannot capture. The mixed weighting allows this step to use time weighting instead, resulting in a smoother solution. With this improved mixed weighting, Figure 5-6 can be updated as Figure 5-13.

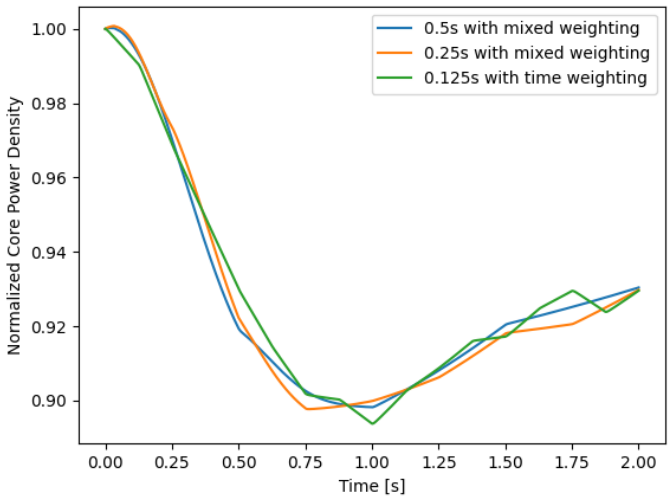

Figure 5-13: Time dependent normalized power density [W/cc] during an inlet temperature transient with mixed time and density weighting.

Figure 5-13 shows that more resolution can be obtained with the spatially-aware mixed time and density weighting scheme. This greatly improves accuracy because the underlying cross sections are being more accurately interpolated. The differences between the 0.5s solution and the 0.25s solution reach a maximum of 0.52% at 0.75s.

Iterative convergence

Iterative convergence is tested to determine an appropriate error threshold for outer time step repetitions using a root mean square error, defined as follows, where P is the Monte Carlo axial power array, and n is the number of fissile nodes:

$$RE = \sqrt{\frac{1}{n}\sum\left(\frac{P_{\text{old}} - P_{\text{new}}}{P_{\text{new}}}\right)^2} \tag{5.1}$$

This root mean square error compares axial power arrays at the end of a time step between iterations of the same simulation. Therefore, separate simulations may still converge to different results.

Runs were launched with up to 4 imposed iterations, each identical to each other but varying the number of outer time step iterations. Tables 5.1, 5.2, and 5.3 show

the residual error of the power after every iteration during the 2 second transient. Based on these tables, error thresholds of 0.01 and 0.005 were compared to imposed iterations, shown in Figure 5-14. At 1.0s, the run with 4 imposed iterations differs by 0.45% from the results obtained with other iterative strategies, and at 2.0s, the differences between results is up to 0.19%. Both are in keeping with the 0.5% fluctuations determined from the null transient. Furthermore, the run with 4 imposed iterations does not obtain a final residual error that is lower than that obtained by the run with 2 imposed iterations. This information can be used to determine the threshold of the convergence criteria.

The iterative convergence criteria of 0.01 achieves similar results to the criteria of 0.005 with significantly fewer iterations as detailed in Tables 5.4 and 5.5, suggesting that a 0.01 criteria can provide a reasonable solution to this problem while being more efficient. A 0.005 criteria nears the statistical limits of this simulation which could hinder convergence unnecessarily. The fact that the run with 0.005 convergence criteria takes 3 iterations to converge the first time step to the 0.01 convergence criteria (and 4 to converge to the 0.005 criteria), whereas the run with 0.01 convergence criteria only take 2 iterations to do so further underlines the statistical nature of the iterative convergence of this problem. Seeding the random number generator is common to avoid such fluctuations. This is done in these calculations, so the results are expected to be identical not only at the first iteration as observed in the tables, but also at subsequent iterations. In practice, the first residual error that does appear to match in all cases (0.08715) is exact to 10^{-10} precision. Then, the following Monte Carlo calculations are all the same up until the 12th, 13th, or 14th batch at which point they diverge from each other. It is possible that a machine precision difference in the TD-TH/CMFD affects the temperatures loaded into Monte Carlo and eventually cause them to differ from each other.

Regardless of the convergence criteria, a minimum number of 2 iterations is needed to allow for the first forward outer Monte Carlo calculation to use the adiabatic approximation and the second one to use the omega method. (This is not necessary when using the tally derivatives, as the first forward outer Monte Carlo calculation

uses the omega method, see Section 4.2.3).

	t=0 to t=0.5s	t=0.5 to t=1.0s	t=1.0 to t=1.5s	t=1.5s to t=2.0s
first iteration	0.08715	0.03050	0.03376	0.02073
second iteration	0.00367	0.00559	0.00255	0.00437

Table 5.1: Residual error of the calculated power during the transient after every iteration with 2 imposed iterations.

	t=0 to t=0.5s	t=0.5 to t=1.0s	t=1.0 to t=1.5s	t=1.5s to t=2.0s
first iteration	0.08715	0.03703	0.03275	0.02029
second iteration	0.00872	0.00308	0.00192	0.00204
third iteration	0.00355	0.00424	0.00190	0.00638

Table 5.2: Residual error of the calculated power during the transient after every iteration with 3 imposed iterations.

	t=0 to t=0.5s	t=0.5 to t=1.0s	t=1.0 to t=1.5s	t=1.5s to t=2.0s
first iteration	0.08715	0.03627	0.03492	0.01518
second iteration	0.00373	0.00330	0.01509	0.00181
third iteration	0.00342	0.00219	0.00419	0.00880
fourth iteration	0.00420	0.00454	0.00771	0.00803

Table 5.3: Residual error of the calculated power during the transient after every iteration with 4 imposed iterations.

	t=0 to t=0.5s	t=0.5 to t=1.0s	t=1.0 to t=1.5s	t=1.5s to t=2.0s
first iteration	0.08715	0.03012	0.03595	0.02118
second iteration	0.00366	0.00318	0.00442	0.00280

Table 5.4: Residual error of the calculated power during the transient after every iteration with an iterative convergence criteria of 0.01.

	t=0 to t=0.5s	t=0.5 to t=1.0s	t=1.0 to t=1.5s	t=1.5s to t=2.0s
first iteration	0.08715	0.03861	0.03921	0.03154
second iteration	0.01176	0.00566	0.01115	0.01625
third iteration	0.01081	0.00224	0.00396	0.00609
fourth iteration	0.00772	-	-	0.00395
fifth iteration	0.00383	-	-	-

Table 5.5: Residual error of the calculated power during the transient after every iteration with an iterative convergence criteria of 0.005.

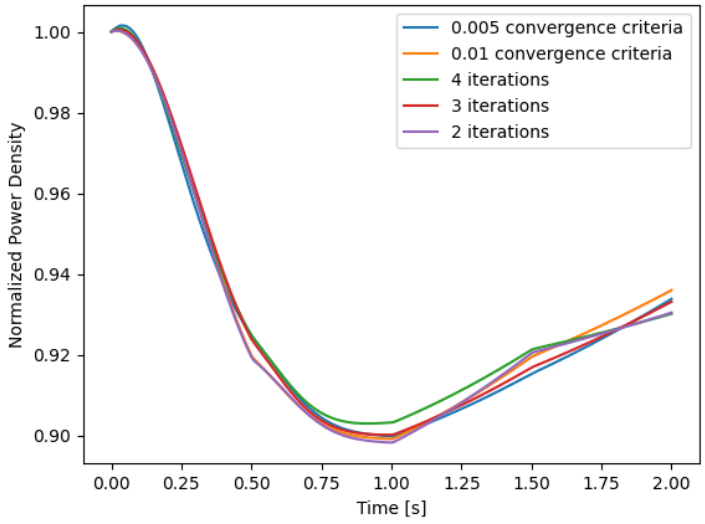

Figure 5-14: Time dependent normalized power density [W/cc] during an inlet temperature transient with varying iterative strategies.

Impact of tally derivatives on iterative convergence

Figure 5-15 shows the change in the solution during iterations over each time step in the case run with 0.5 second outer time steps and 0.01 convergence criteria (inner time steps are still 0.0025 seconds, and the Monte Carlo particle settings remain unchanged). This is without tally derivatives: the zeroth iteration of the first time step, the top left of Figure 5-15 (before the forward outer Monte Carlo step is com-

puted), holds steady-state because the solver assumes that the forward outer step cross sections are identical to the previous outer step. Then, once the forward outer step is computed, the first and second iterations improve upon the inner time step calculations with mixed time- and density- weighted cross sections. Similarly, the zeroth iteration of the second time step, the top right of Figure 5-15 assumes that the forward outer time step cross sections are identical to the previous outer time step, but it does propagate delayed neutron precursors, therefore allowing for their decay to be calculated into the transient. Then, once the forward outer step is computed, the following iterations use mixed time- and density- weighted cross sections. The same is shown for the third and fourth time steps. Notice that in these last two steps, the zeroth iteration predicts the wrong trend of the transient because it takes into account changes in delayed neutron precursors, but is unable to account for thermal feedback.

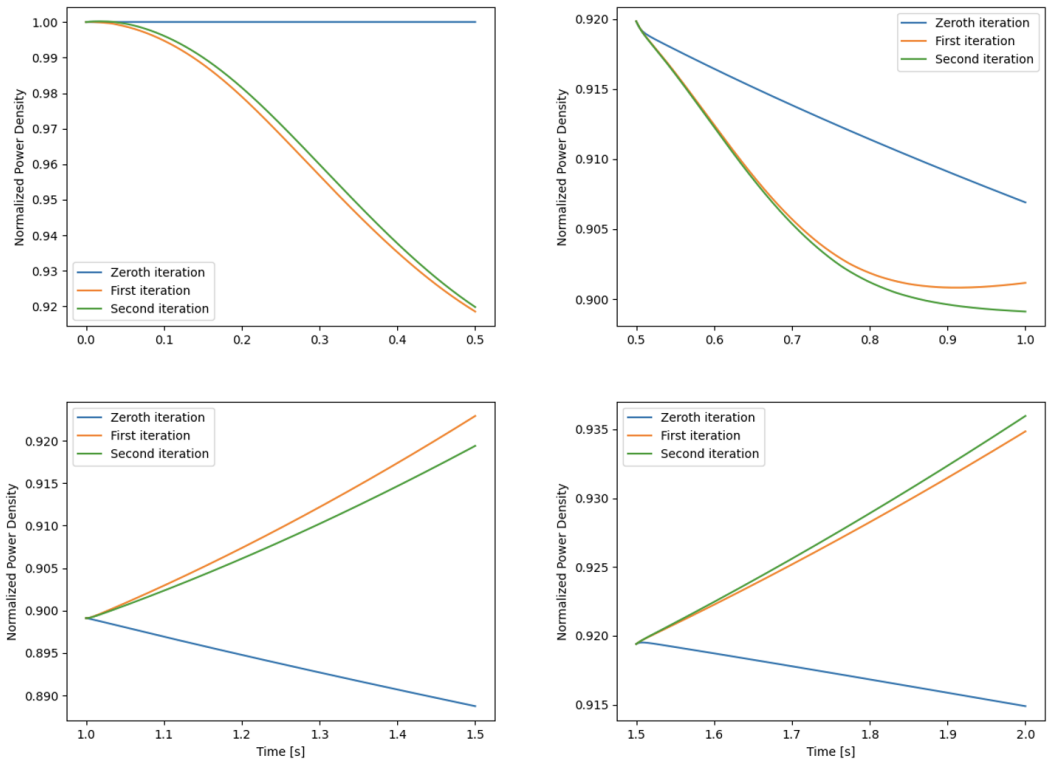

Figure 5-15: Iterative convergence during the four time steps without tally derivatives in the zeroth iteration.

Zeroth iteration tally derivative extrapolation The use of tally derivatives allows a more sophisticated first order prediction of cross sections during the zeroth iteration. Tally derivatives were demonstrated in Section 4.2 on a simple problem to predict the changes between two steady-state conditions which also allowed a quantification of the errors they carry. The use of tally derivatives between transient states is a potential improvement to the mixed density weighting interpolation as it can leverage both coolant density and fuel temperature changes without needing to know the extent of the time step's material changes a priori. Figure 5-16 is identical to 5-15, with the same convergence criteria, but the tally derivatives are used during the zeroth iteration. It is immediately clear that the zeroth iteration is greatly improved in every time step. This also allows the use of the omega method in the first iteration. In the third and fourth time steps, the zeroth iteration without tally derivatives wrongly predicts the direction of the transient, but with tally derivatives, that is corrected. In the final step, this results in a reduction in the number of required iterations, as the convergence criteria is met after just the first iteration. However, since the rate of iterative convergence is stochastic, as shown in the previous section, 5.1.2, this isn't always the case. Regardless, with the clear improvement of the zeroth iteration, and the ability to use the omega method in the first iteration, it is reasonable to conclude that the tally derivatives lend an advantage to the iterative convergence of this problem.

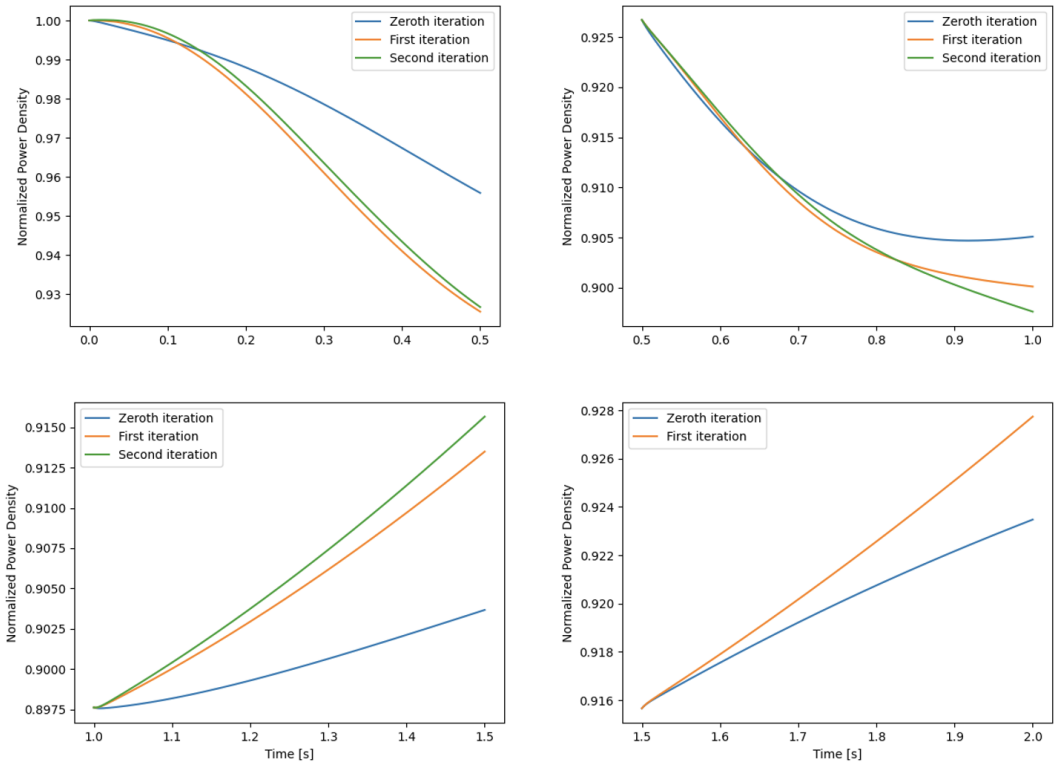

Figure 5-16: Iterative convergence during all four time steps with tally derivatives in the zeroth iteration.

The tally derivatives appear to predict the general trend of the transient without having any information about the cross sections at the forward outer step. To verify that the behavior of the tally derivatives is physical, the evolution of several cross sections are shown, comparing their behavior during different iterations.

The zeroth iteration uses information taken at the previous outer time step and extrapolates based on tally derivatives and the predicted material changes. Tally derivatives have shortcomings that are discussed in Section 4.2, and the longer the time step, the less extrapolation can be trusted. The subsequent iterations use information at the previous and forward outer time steps to perform an interpolation of cross sections that is weighted in relation to the predicted material changes, but do not leverage tally derivatives. In this case, the anchoring at the outer time steps is considered accurate, but the interpolation schemes used are approximations. As such, the tally derivative extrapolation and the cross section interpolation can be compared

against each other to understand the flaws of each approach. Similarities between the two may indicate that the solution is good, but that cannot be guaranteed because both approaches are approximations. The following plots show the total cross section, the thermal absorption cross section (defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$; fast absorption is a small effect and results contain statistical noise due to how it's defined as the difference of two other cross sections), and the downscattering cross section at 1/3 and 2/3 from the bottom of the fuel rod at every time step. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation. They are expected to yield the same answers as they iteratively converge to a 0.01 residual error threshold.

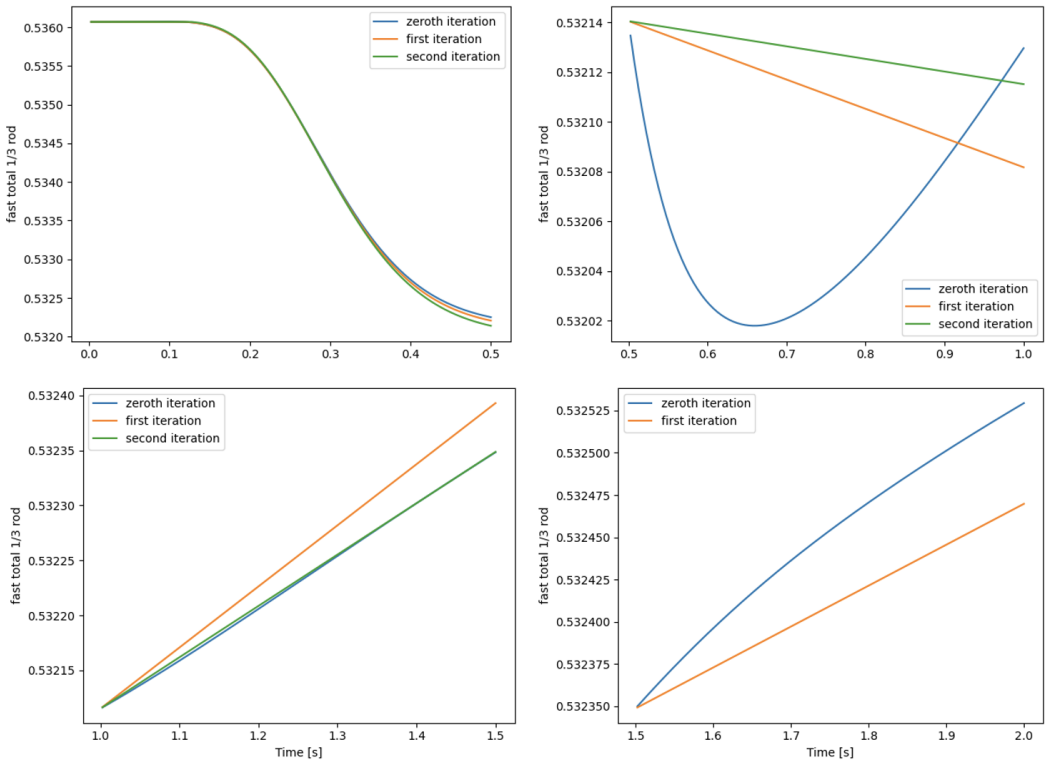

Figure 5-17: Evolution of the fast total cross section 1/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation.

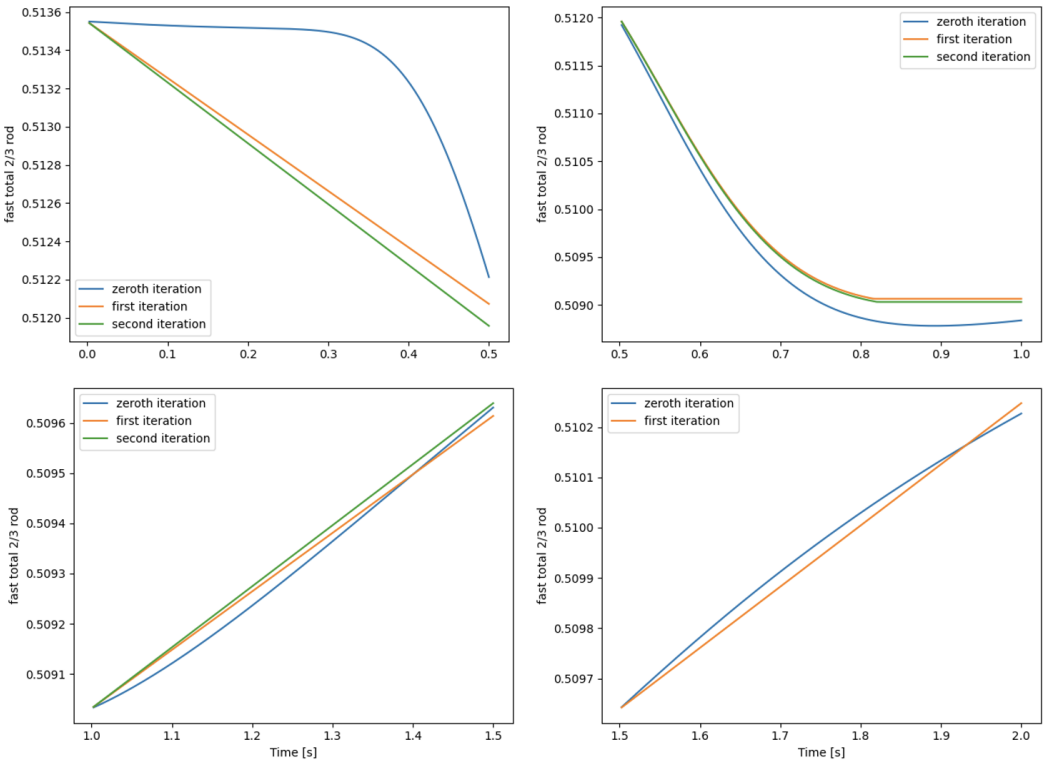

Figure 5-18: Evolution of the fast total cross section 2/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation.

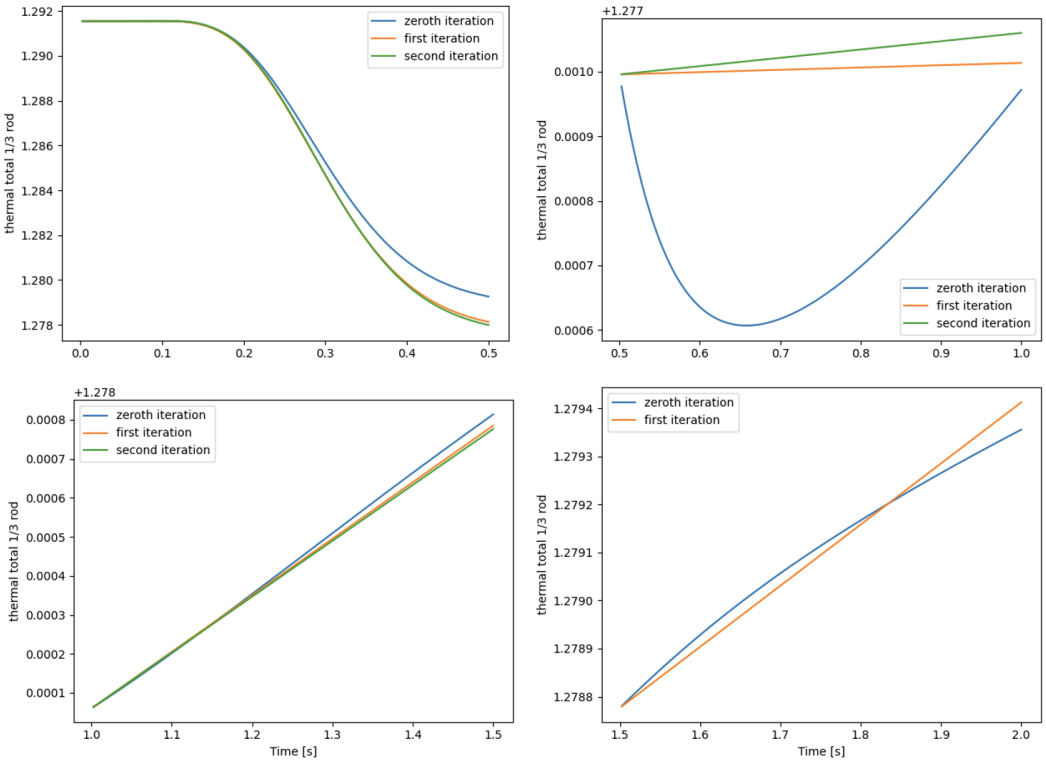

Figure 5-19: Evolution of the thermal total cross section 1/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation.

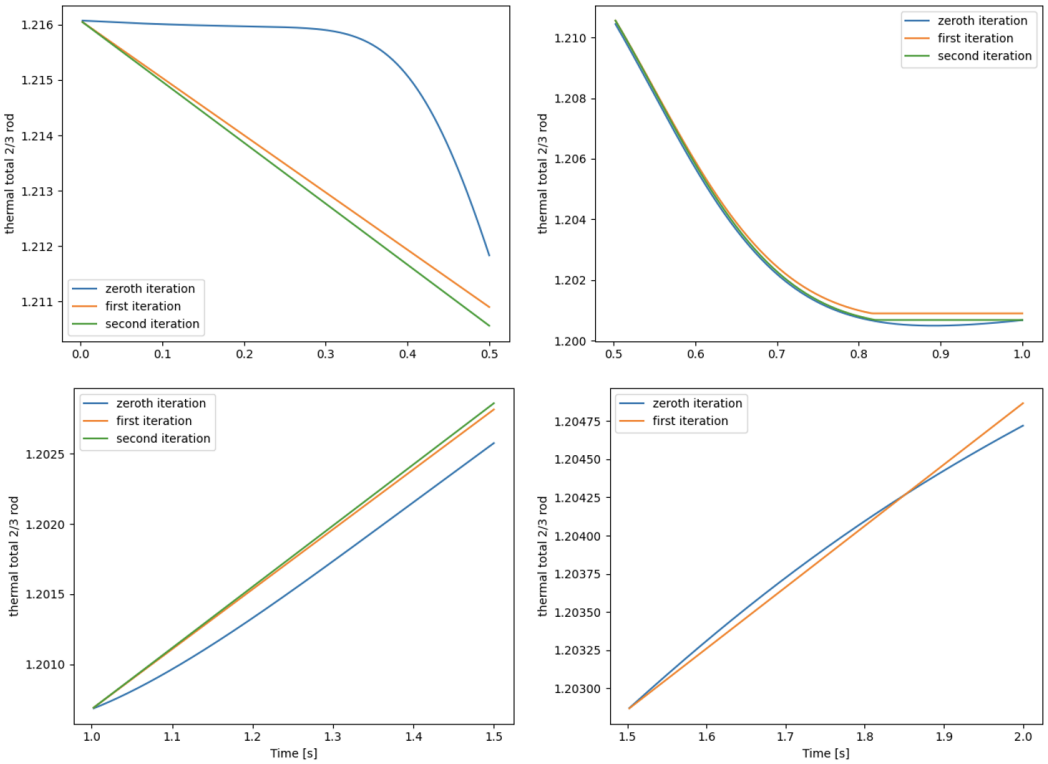

Figure 5-20: Evolution of the thermal total cross section 2/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation.

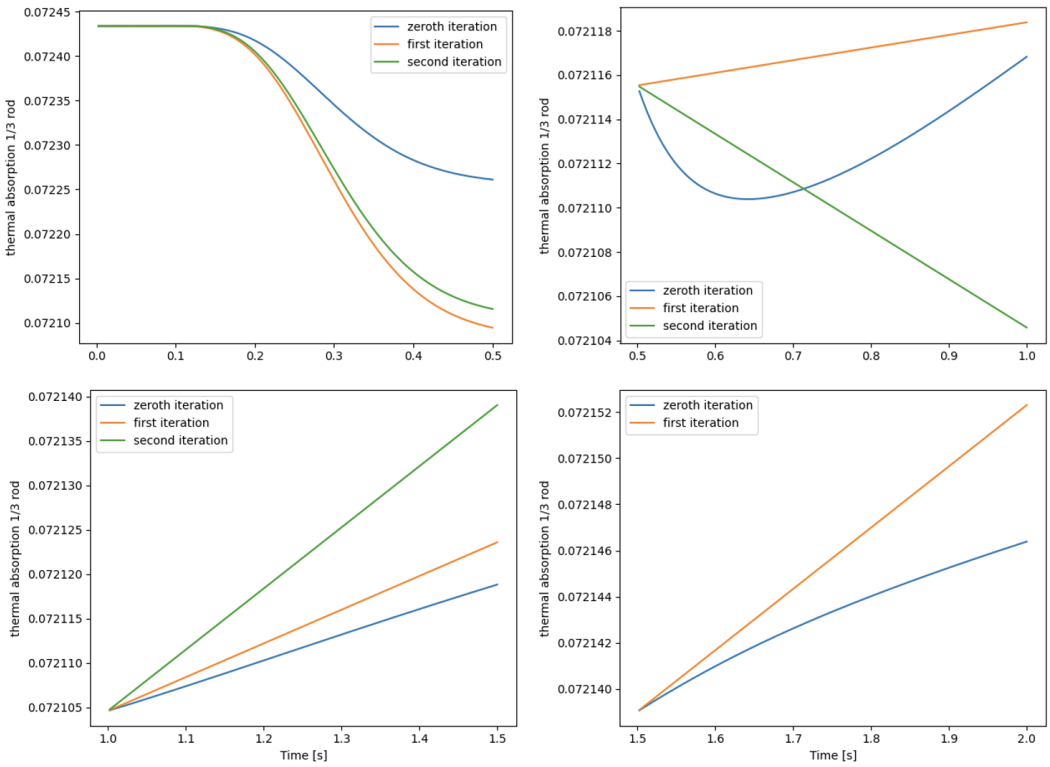

Figure 5-21: Evolution of the thermal absorption cross section 1/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation.

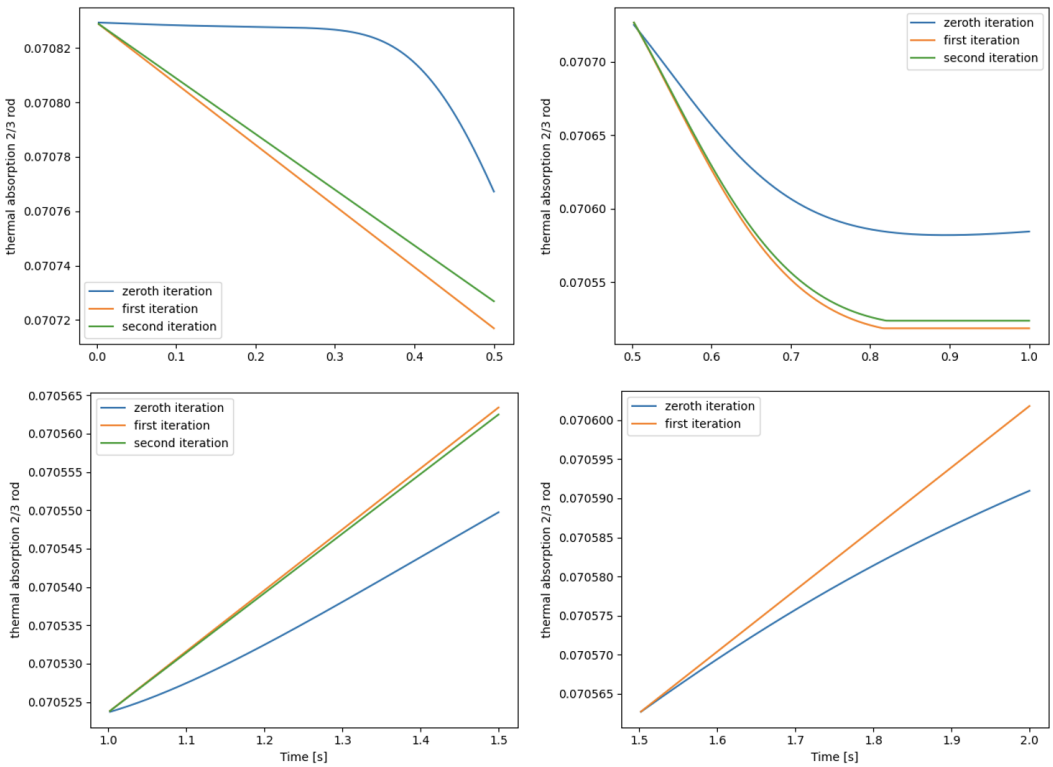

Figure 5-22: Evolution of the thermal absorption cross section 2/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation.

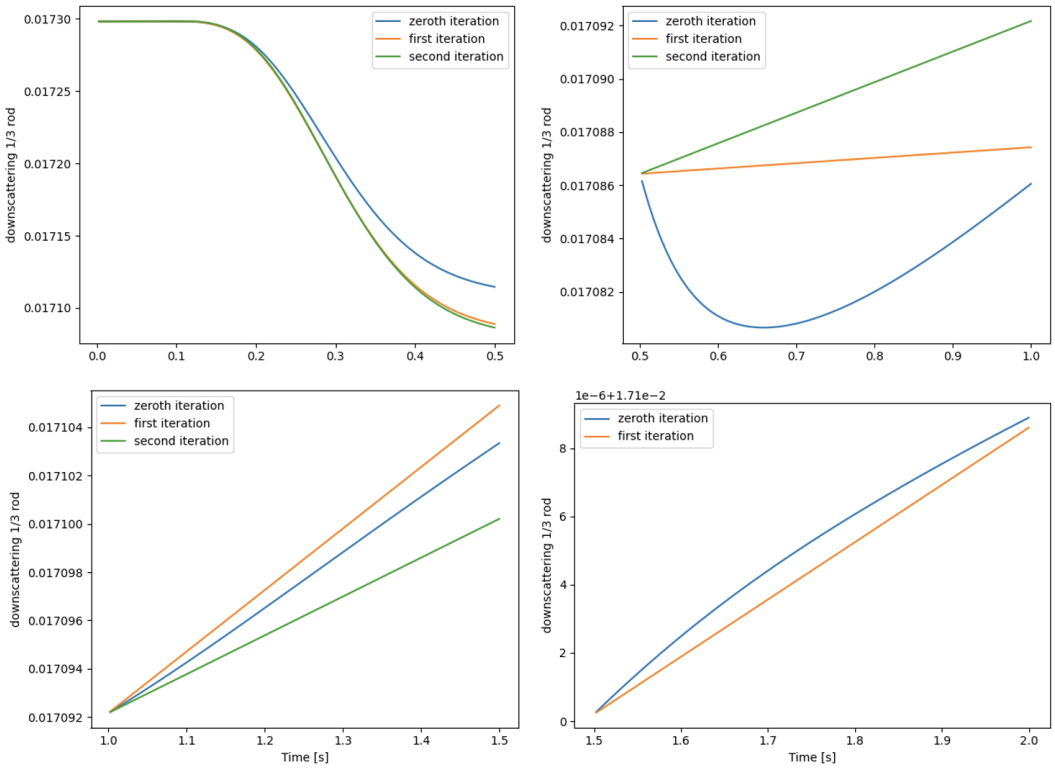

Figure 5-23: Evolution of the downscattering cross section 1/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation.

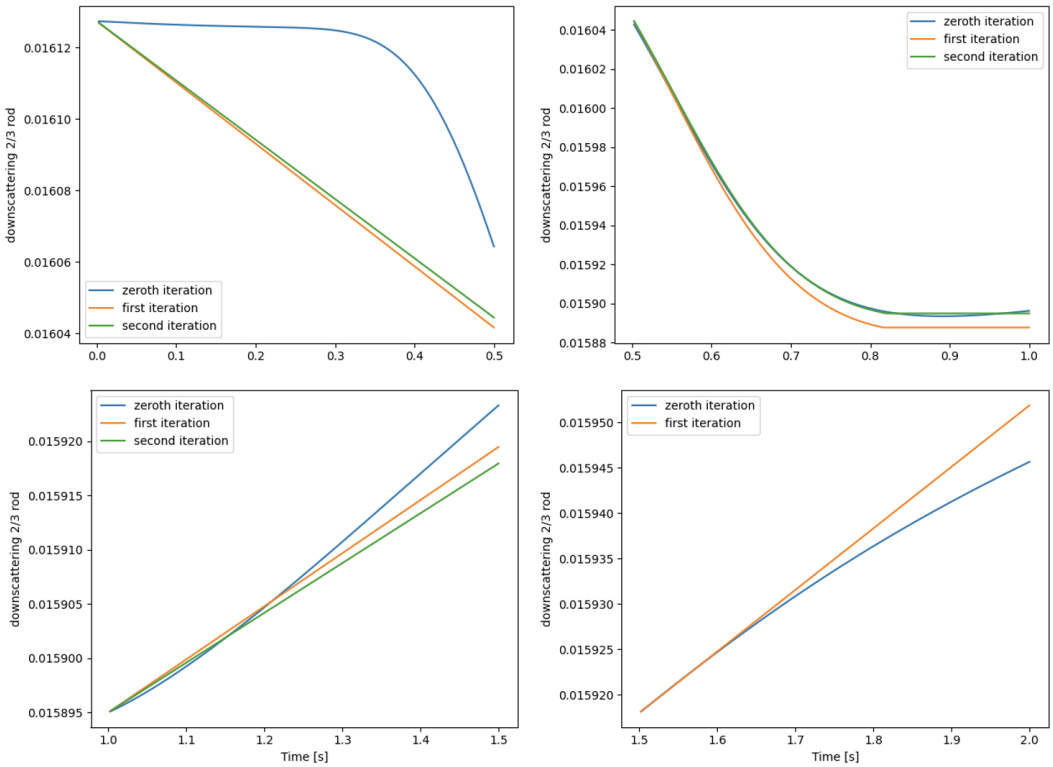

Figure 5-24: Evolution of the downscattering cross section 2/3 from the bottom of the rod during the inner time steps at every iteration over the course of the 2 second transient. The zeroth iteration uses tally derivative extrapolation, while the first and second iterations use cross section interpolation.

In these plots, the tally derivative extrapolation at the end of the time step is near or between the subsequent iterations in all but a few of the cases shown. This suggests that the time step size and the material changes are small enough for the tally derivative extrapolation to be reliable in this problem. Conversely, the interpolation follows the same cross section progression as the tally derivatives in the majority of cases shown, lending credibility to its implementation in this problem.

It would be advantageous to use tally derivatives in all iterations as they are a very flexible and physics-based cross section modeling tool. Several interpolation methods are implemented that use modifications of the tally derivative extrapolation to incorporate information about the forward outer Monte Carlo calculation.

All iteration tally derivative extrapolation One possible scheme favors tally derivatives at the start of the time step, and the Monte Carlo forward outer cross section at the end, such that the weighting scheme of Section 4.1.3 becomes:

$$\Sigma_a = \Sigma_{a,\text{td}} * (1 - w) + \Sigma_{s+1} * w \tag{5.2}$$

$$w = 1 - \frac{t_{s+1} - t_a}{t_{s+1} - t_s} \tag{5.3}$$

where

$$\Sigma_{a,\text{td}} = \Sigma_s + \left(\frac{\partial \Sigma}{\partial \rho}\right)_s \Delta\rho + \left(\frac{\partial \Sigma}{\partial T}\right)_s \Delta T \tag{5.4}$$

The cross sections being interpolated are (1) the previous outer Monte Carlo cross section extrapolated with tally derivatives and (2) the Monte Carlo forward outer cross section. This weighting scheme does, however, mitigate the effect of tally derivatives on the cross section prediction, as shown in Figure 5-25. The total cross section at 1/3 height of the fuel rod is not expected to change at all until the density and temperature changes reach that point in the geometry. But the weighting scheme muddles the physics of the tally derivatives to force agreement with the forward outer cross section. The overall behavior of the transient is shown in Figure 5-26 which does not result in the expected power shape. This calculation used a convergence criteria of 0.01, consistent with Section 5.1.2, which required 3 iterations on the first

two time steps, and 2 iterations on the final two time steps.

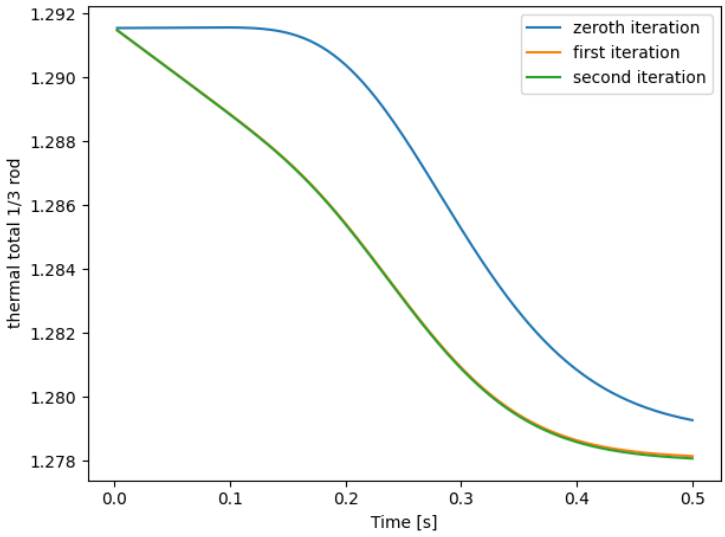

Figure 5-25: The thermal total cross section 1/3 from the bottom of the rod during the inner time steps in an interpolation scheme that leverages tally derivative extrapolation during all iterations.

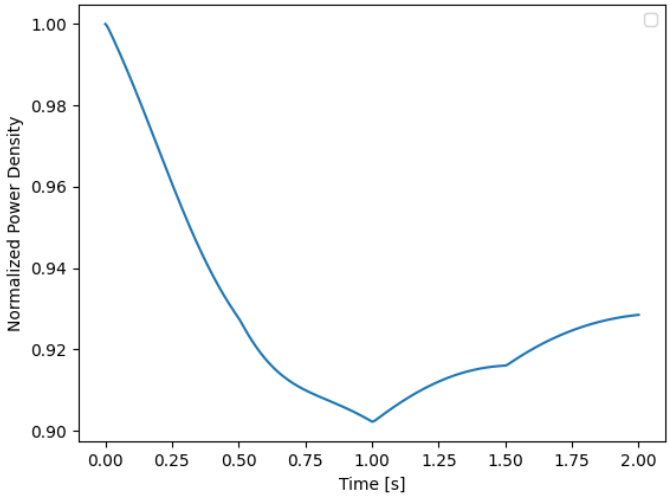

Figure 5-26: Low accuracy power in an interpolation scheme that leverages tally derivative extrapolation during all iterations.

All iteration tally derivative two-way extrapolation An improvement to its accuracy is possible if tally derivative extrapolation occurs from both the previous

outer Monte Carlo cross section and the forward outer Monte Carlo cross section to model inner time step cross sections as described below.

Extrapolation from the previous outer Monte Carlo cross section:

$$\Sigma_{a1} = \Sigma_s + \left(\frac{\partial \Sigma}{\partial T}\right)_s (T_a - T_s) + \left(\frac{\partial \Sigma}{\partial \rho}\right)_s (\rho_a - \rho_s) \tag{5.5}$$

Extrapolation from the forward outer Monte Carlo cross section:

$$\Sigma_{a2} = \Sigma_{s+1} + \left(\frac{\partial \Sigma}{\partial T}\right)_{s+1} (T_a - T_{s+1}) + \left(\frac{\partial \Sigma}{\partial \rho}\right)_{s+1} (\rho_a - \rho_{s+1}) \tag{5.6}$$

Using both to inform the cross section modeling results in:

$$\Sigma_a = \Sigma_{a1} * (1 - w) + \Sigma_{a2} * w \tag{5.7}$$

with

$$w = 1 - \frac{t_{s+1} - t_a}{t_{s+1} - t_s} \tag{5.8}$$

The resulting power behaves similarly to the mixed-density weighting scheme, seen in Figure 5-27, making this a very attractive approach to tally derivative use in transients. The two-way extrapolation is more flexible and more robust against axial variations in cross section changes than the mixed-density weighting scheme which relies on a specifically chosen marker (density) to determine the rate of cross section change. Figure 5-27 does however lend credibility to its use in this thesis as its results closely resemble those of the two-way extrapolation scheme, with the exception of the very beginning of the power profile. Figures 5-28 through 5-30 show the cross section modeling during the first time step compared to a reference solution. This calculation used a convergence criteria of 0.01, consistent with Section 5.1.2, which required 3 iterations on the first and third time steps, and 2 iterations on the second and fourth time steps.

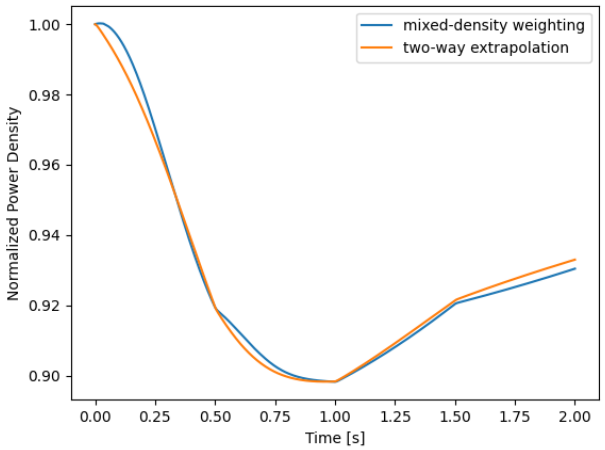

Figure 5-27: Power in a two-way tally derivative extrapolation scheme.

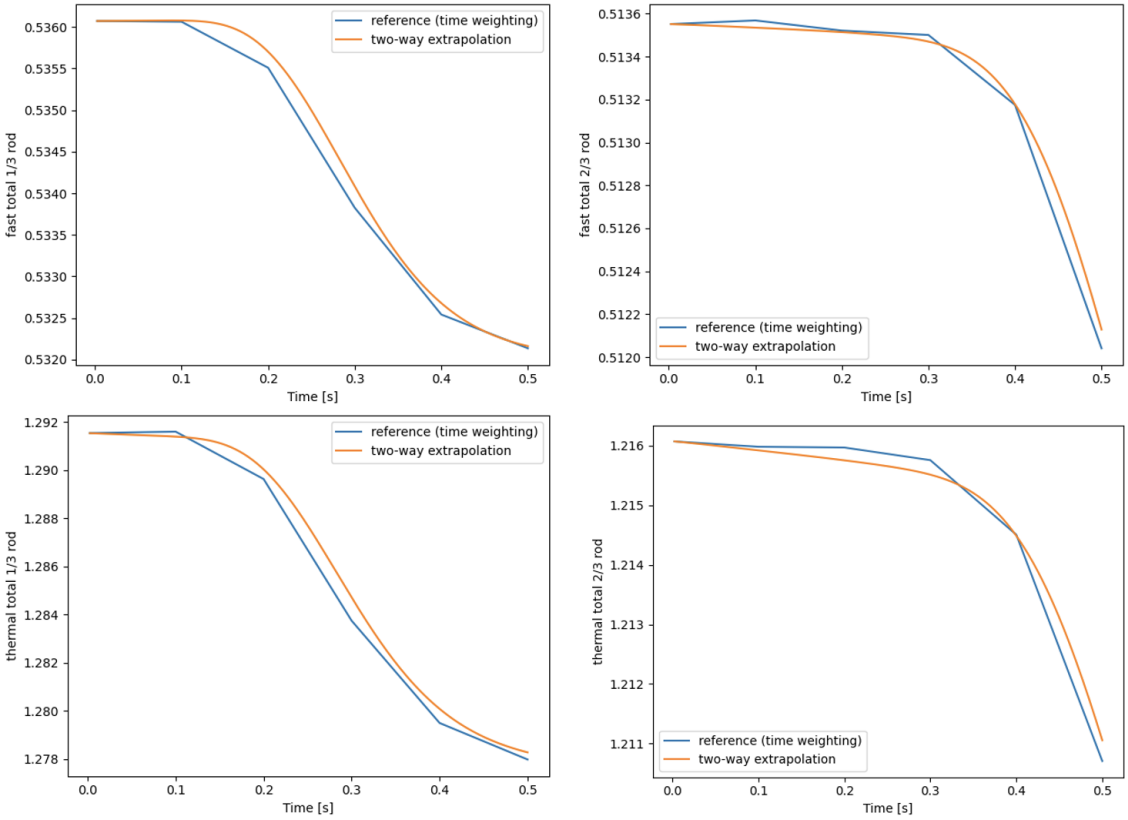

Figure 5-28: Evolution of the total cross section during the first outer time step compared to a reference solution. Fast cross section 1/3 from the bottom of the rod (top left), fast cross section 2/3 from the bottom of the rod (top right), thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right).

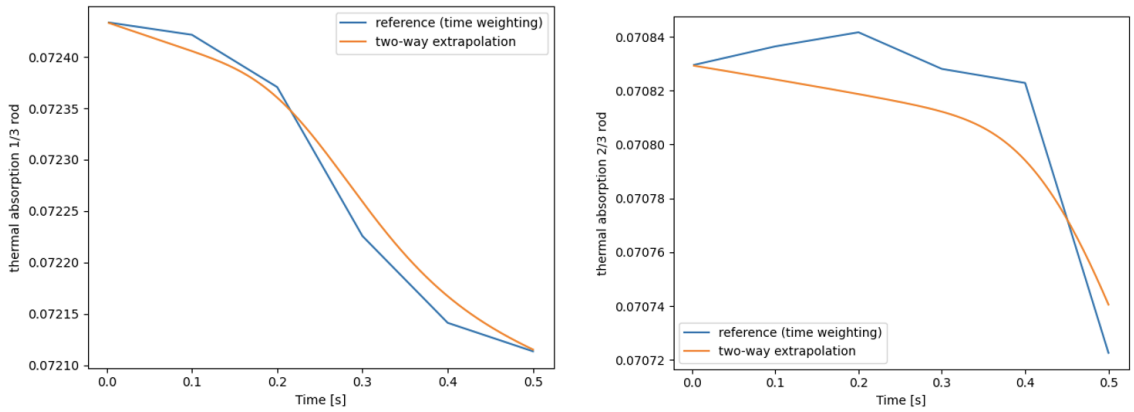

Figure 5-29: Evolution of the thermal absorption cross section (defined as $\Sigma_{a,g} = \Sigma_{t,g} - \Sigma_{s,g}$) during the first outer time step compared to a reference solution. Thermal cross section 1/3 from the bottom of the rod (bottom left), and thermal cross section 2/3 from the bottom of the rod (bottom right).

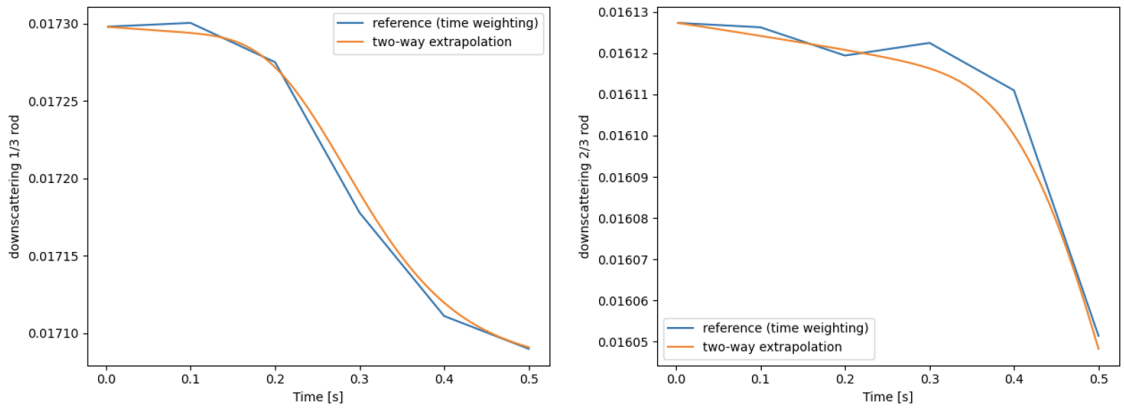

Figure 5-30: Evolution of the downscattering cross section during the first outer time step compared to a reference solution. Location 1/3 from the bottom of the rod (left), and location 2/3 from the bottom of the rod (right).

The mixed time- and density- weighting scheme developed in Section 4.1.3 and analyzed above appears to perform very well compared to alternatives. This is due in part to the fact that it is tailored toward this transient. Knowing the expected behavior of the transient allows water density to be the driving component of the axially-varying cross section behavior while fuel temperature is known to behave more uniformly in this case. The use of tally derivatives in all iterations of this scheme is shown to be successful and worth further exploration as it provides a more generic approach to transients of unknown behavior. In this two-way extrapolation scheme, the tally derivatives are not expected to change very much between outer time steps during this transient because of how mild the material changes are. This is not necessarily true of all transients, but may afford the opportunity to save computational time by computing tally derivatives only at the first Monte Carlo calculation for select problems. Figures 5-31 and 5-32 show the evolution of the tally derivative for fission during this transient for fuel temperature and water density, respectively.

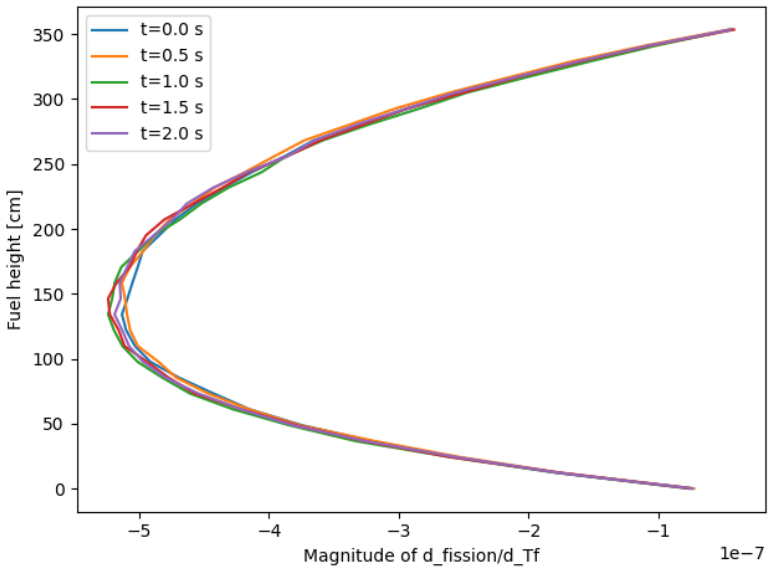

Figure 5-31: Evolution of the fission tally derivative with respect to fuel temperature at every outer time step. The x-axis units are reactions per source particle per degree Kelvin.

Tally derivatives come at a computational cost that depends on their implementation details and the geometry of the problem. In a simulation without tally deriva-

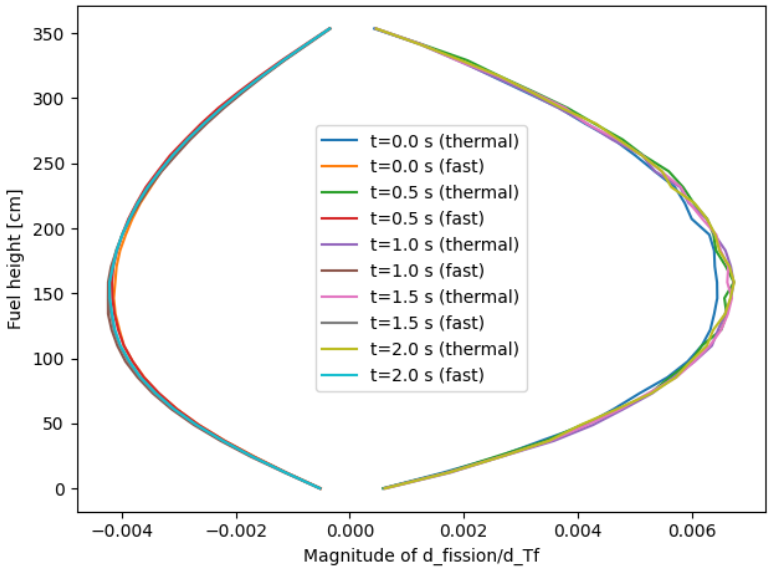

Figure 5-32: Evolution of the fission tally derivative with respect to water density at every outer time step. The x-axis units are reactions per source particle per g/cc.

tives, the HOLO scheme described in this work is already very costly. In Table 5.6, the cost of inactive Monte Carlo batches is significantly lower than the active batches, where tallies are calculated. Without tally derivatives, 13 tallies are generated by the OpenMC multi-group cross section library. These tallies are already merged with each other where possible to reduce the total number of tallies and avoid duplicates. These tallies ensure that all the cross sections, currents, and diffusion coefficients needed by TD-CMFD are available on a mesh scale compatible with the TD-TH, resulting in a 6.5 times slow down of the calculation with respect to the inactive (where no tallies are present). The tallies therefore constitute a major bottleneck of the Monte Carlo computational speed. With the addition of tally derivatives, the simulation of active batches runs approximately 20% slower than without tally derivatives. These tally derivatives are responsible for 31 new tallies in OpenMC: one for the fuel temperature derivative, and one for each of the 30 water materials' density derivatives. In the following section, a necessary approximation is made when larger geometries are simulated to be able to use tally derivatives without drastically increasing the number of tallies.

	inactives	actives without tally derivatives	actives with tally derivatives
Runtime ratio to baseline (inactives)	1.0	6.5	7.2

Table 5.6: Runtime quantification of active cycles with and without tally derivatives based on data saved by OpenMC statepoint files after running on approximately 200 CPUs.

In summary, several approximations for inner time step cross sections have been presented. The first is tally derivative extrapolation which uses a costly stochastic derivative for Monte Carlo tallies to predict how cross sections change under material perturbations. The scope of material changes under which a tally derivative can be used is unclear, but the results shown above provide evidence that this problem's parameters are within that scope. The implementation of tally derivatives in a transient scenario is a novel contribution to the scientific literature that warrants further research. The second approximation for inner time step cross sections is a relatively inexpensive interpolation between outer time steps. Previous work on spatially uniform, prescribed transients [76, 77] exclusively used time-weighted interpolation assuming a constant rate of change across all cross sections. However, that was deemed insufficient in a spatially non-uniform transient where material changes incurred different rates of cross section progression throughout the geometry. To amend this issue, a mixture of time and material property based interpolation weighting was implemented that showed promise for this problem but might not work across other transients that are fuel temperature dominated. The comparison between tally derivative extrapolation and cross section interpolation gives the opportunity to evaluate their merits. Further tests leveraging tally derivatives in all iterations were found to be successful and worth further exploration.

5.2 Uniform Assembly Geometry

In order to assess the behavior of these methods in a larger configuration, an assembly geometry has been selected for analysis. This geometry consists of 289 rods in a 17 by 17 configuration as shown in Figure 5-33. Each rod is identical to the rod described in Section 5.1. This implies that the assembly results should match the results in Section 5.1, allowing the assembly calculation to act as verification for scaling up from a single fuel rod to an assembly. Furthermore, this thesis relies on a simplified TD-TH solver that accounts only for the properties of UO2 fuel, zirconium cladding, and other material-specific equations. This solver has been tailored toward the properties of this fuel rod, so using different rod types (MOX fuel or guide tubes) in addition would require different heat transfer considerations for each rod. Therefore, the assembly is not an industry realistic geometry, but is representative of the computational cost of an assembly calculation. The geometry is surrounded by reflective boundary conditions on all sides, and vacuum boundary conditions on the top and bottom. Each rod is surrounded by its own coolant in a "rod channel" with no cross flow between these. Thus, each rod has its own thermal solution, uncorrelated to other rods, except by the power shape determined by Monte Carlo. Every cell in every rod is defined separately so that its temperature properties can be updated independently. This is also true for all 30 axial cells of each material in the y-z direction. This amounts to 8,670 individual cells for each material in the geometry (fuel, cladding, gap, and water). Furthermore, every water cell is filled with a separate water material so that its density can be updated independently. Side view images of the material and cell configurations are shown in Figures 5-34 and 5-35.

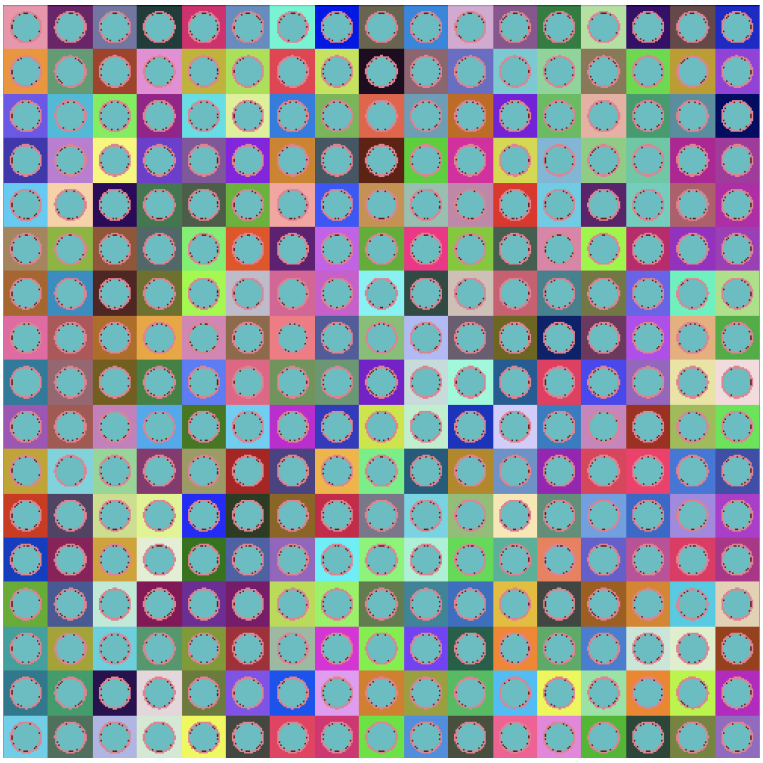

Figure 5-33: 17x17 assembly geometry from the x-y view, coloring by material. Since temperatures can be assigned to cells (as opposed to materials), all fuel, cladding, and gap materials are the same material definition across the geometry. However, densities must be assigned to materials, so water materials are different in every cell.

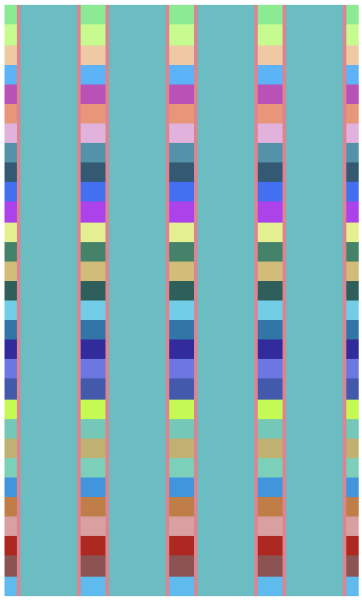

Figure 5-34: Four rods of the geometry from the y-z view, coloring by material, showcasing the individual water materials at every axial cell.

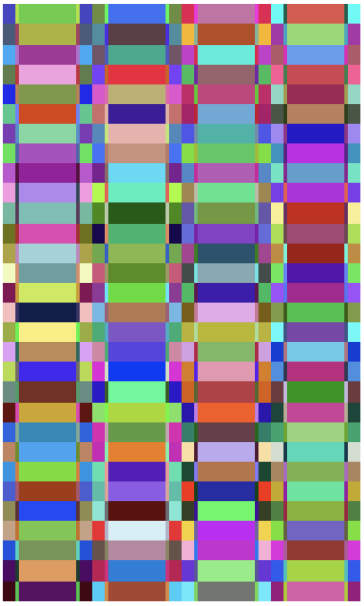

Figure 5-35: Four rods of the geometry from the y-z view, coloring by cell, showcasing the individual cells of every material at every axial cell.

The initial conditions of this geometry are determined through ramp-up Monte Carlo (described in Chapter 2, using a batch progression of 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, and 500 batches): at each cycle of the ramp-up process, the entire geometry undergoes a prescribed number of Monte Carlo batches to determine the power distribution with 5 million particles per batch. The progression of batch numbers in every cycle is designed to converge any possible numerical oscillations of the coupled power/temperature solution. Figure 5-36 shows how the relative standard deviation of tallies and the eigenvalue evolve during ramp-up Monte Carlo, as well as the root mean square change in power across all cells, as calculated below where P_j is the power vector at cycle j, i is each cell of the geometry, and n is the total number of cells:

$$\sqrt{\frac{1}{n}\sum_i \left(\frac{P_j^i - P_{j-1}^i}{P_j^i}\right)^2} \tag{5.9}$$

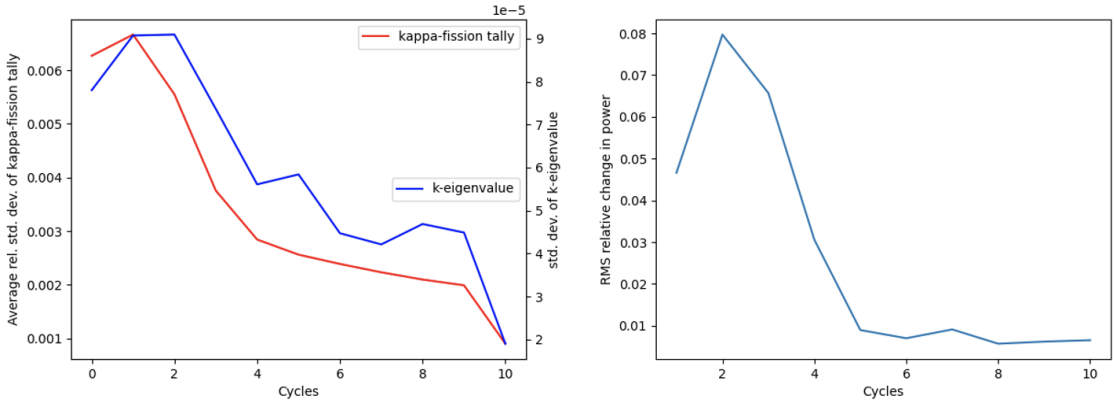

Figure 5-36: Evolution of key convergence figures during the ramp-up Monte Carlo procedure. On the left: tally convergence and k-eigenvalue convergence increase with each cycle. On the right: the root mean square change in power between consecutive cycles, with the final change being of 0.5%.

This steady-state calculation resulted in a convergence of the k-eigenvalue to 1.9 pcm and the maximum relative standard deviation of the kappa-fission tally of 0.46%. The spatial variations in power are shown in Figure 5-37 where power difference from the assembly-average is shown for a radial cut in the geometry half way up the assembly. The differences smooth out and reduce in magnitude as convergence is reached.

Between each cycle, the temperature distribution of each rod is determined individually. Figures 5-38 and 5-39 show the final water and fuel temperature distribution resulting from this ramp-up Monte Carlo method.

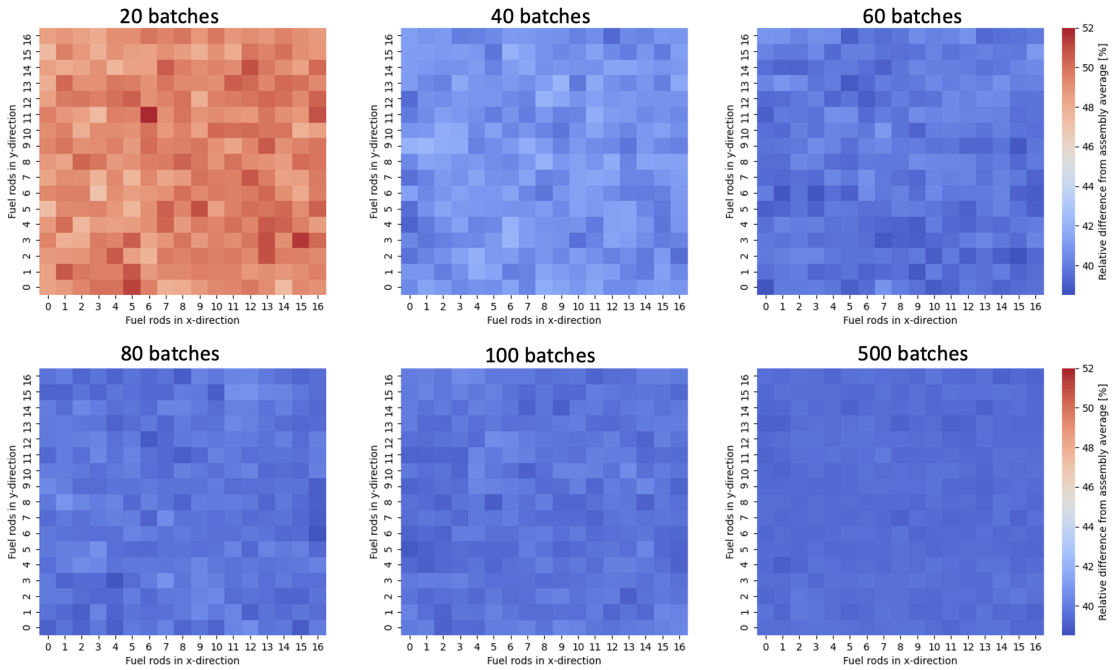

Figure 5-37: Evolution of the power difference from the assembly-average for select cycles: 20 batches, 40 batches, 60 batches, 80 batches, 100 batches, and 500 batches (starting at the top left).

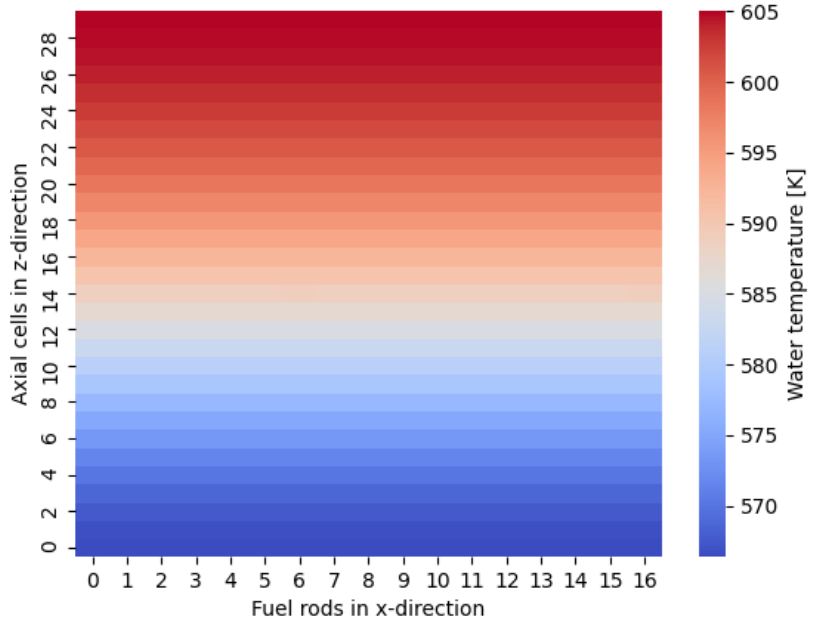

Figure 5-38: Water temperature distribution in axial view of the assembly geometry.

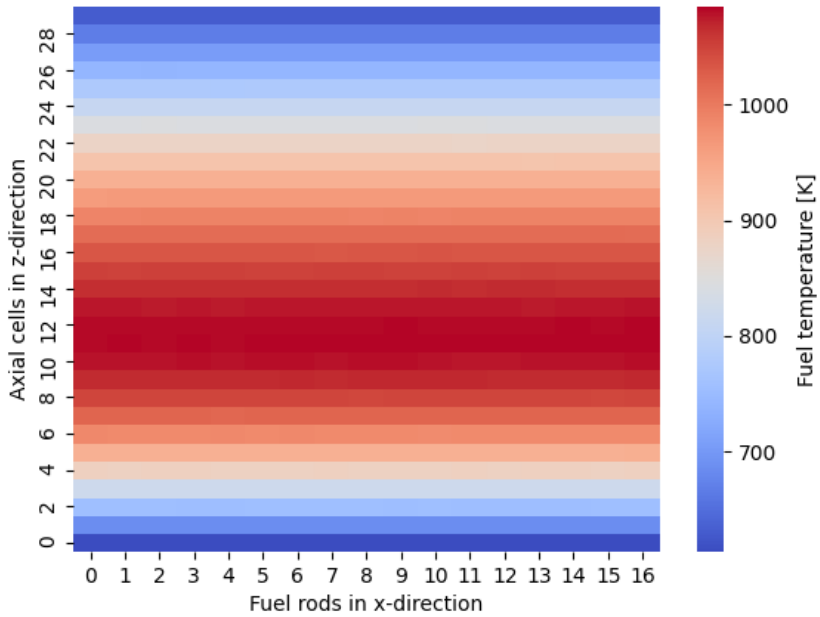

Figure 5-39: Fuel temperature distribution in axial view of the assembly geometry.

5.2.1 Null transient considerations

For all tests on this geometry, the Monte Carlo parameters are 1100 batches, including 100 inactive batches, with 5 million particles per batch. The null transient is run to determine the error associated with stochastic fluctuations. The study does not need to be as comprehensive as in 5.1.1 because the solvers involved with the assembly are a parallel implementation of the same rod-wise calculation (in the case of the TD-TH) and maintain a rod-level mesh for calculation (in the case of TD-CMFD) and tallies (for Monte Carlo). Therefore, the knowledge gained in Section 5.1 can be used to select the best cases for the assembly.

Figure 5-40 shows that the normalized power density diverges from 1.0 W/cc by 0.1% after two time steps which is within the fluctuations identified for a single fuel rod for one time step in 5.1.1. Additionally, Figure 5-41 shows that the maximum temperature change over the course of the 1.0 second null transient is 0.01 Kelvin.

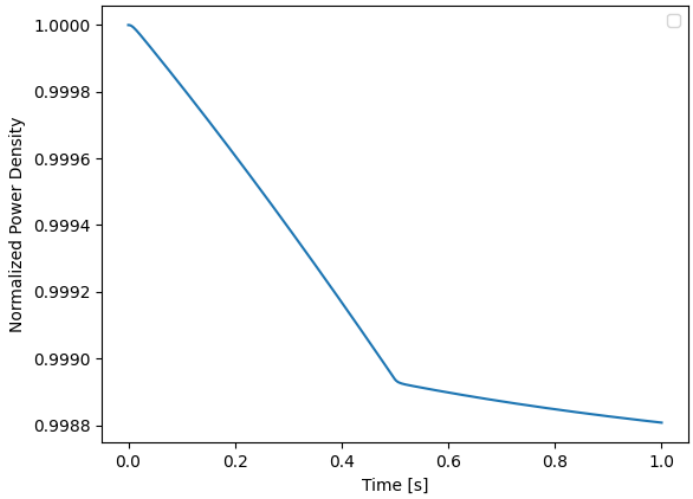

Figure 5-40: Change in normalized power density [W/cc] during a null transient on the assembly geometry.

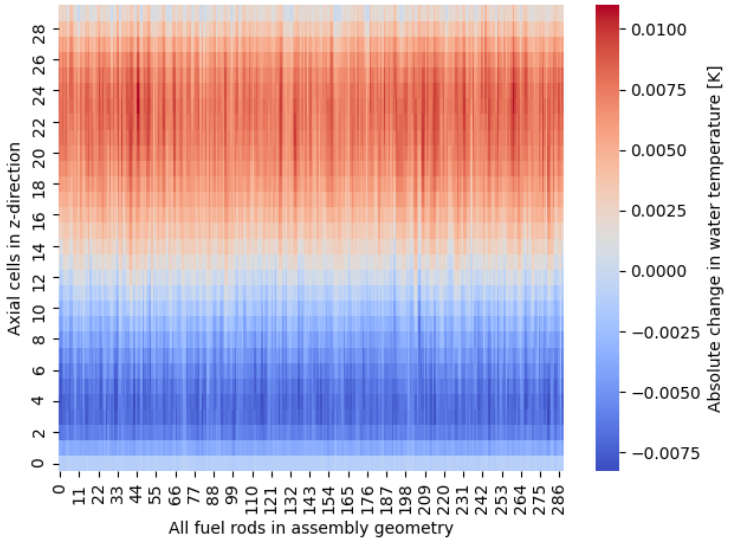

Figure 5-41: Absolute change in water temperature in Kelvin between t=1s and t=0s among all 289 fuel rods during the null transient.

5.2.2 Uniform inlet temperature transient

The first transient studied on this geometry is identical to the inlet temperature change described in 5.1.2, applied to all rods in the assembly. This is a verification step to ensure that the parallel fuel rod implementation is correct, as the power transient should be identical to the single rod case. All 289 water channels will have their inlet temperature rise from 566.25 K to 570 K instantaneously at t=0. One significant implementation change compared to the single fuel rod is the application of the water density tally derivative. In the assembly geometry described herein which consists of 289 rods, each with 30 cells (8,670 cells total), the tally derivative calculation of each cell's impact on all other cells results in 8,670 tallies, each with 8,670 data points, for a total of over 75 million data points. This is infeasible to run even in a steady-state calculation. The application of this tally was already simplified as explained in 4.2.3, where the overall effect on a cell was calculated by a sum rather than a matrix multiplication of all the data points. Now, a further simplification is used: tally derivatives are only calculated on a single fuel rod of the geometry (only 30 cells for a total of 900 data points). The effects on each of those cells are summed as described above,

and then applied to all fuel rods. This is expected to work well for uniform problems where all rods undergo the same transient. However, for asymmetrical problems, this will be a source of error. Conversely, because the tally derivatives are only used as a first order prediction tool in this numerical scheme, only the accuracy of the zeroth iteration is expected to be impacted in this case. A similar approximation, called the Uniform Derivative Approximation, was needed in the demonstration of tally derivatives for steady-state calculations when applied to large geometries in Reference [21]: "In order to compute a differential tally with respect to the temperature in region i, the MC solver must track each particle that enters region i and update that particle's weight derivative each time it moves in region i or undergoes a reaction in region i. Note that this is true even if the tally itself is limited to a different region [...]. To compute each element in the full Jacobian, the MC solver must keep track of i perturbed weights for each particle—one weight per Jacobian column. Instead, this work circumvents the issue by treating all of the fuel as a single perturbed region for the purposes of the fuel temperature derivatives. Instead of i weight derivatives, the solver just tracks one weight derivative for temperature. This one weight derivative is updated whenever a particle interacts with a fuel region, regardless of which fuel region it is. Similarly, all of the coolant is treated as one perturbed region in order to compute the coolant density derivatives." This approximation allowed the computational expense to be low enough for a quarter core calculation for a steady-state multiphysics calculation in [21]. In this work, the 2 second transient scheme requires approximately 9 steady-state calculations, further constraining the size of problem that can be run within the computational limits of the resources available.

The change in normalized power density of this uniform transient in the assembly geometry is expected to be the same as in the fuel rod transient within statistical fluctuations. This is shown in Figure 5-42, where the differences between the fuel rod and the assembly are similar in magnitude to the differences between two fuel rod calculations shown in Figure 5-13. In fact, the statistics may be even less reliable in this case because with 5 million particles per batch for the assembly, that is approximately 17,300 particles per fuel rod, whereas the single fuel rod could be run with

400,000 particles per batch.

The temperature change during the 2-second transient is subtle in Figure 5-43, but can be observed to increase across the entire geometry. For clarity, the temperatures of a central rod from Figure 5-43 is plotted at t=0s and t=2s in Figure 5-44.

The precursor frequency terms ($\frac{\lambda}{\lambda+\omega^{D,b}}$), which are spatially distributed, are shown in Figure 5-45. They give a clear picture of how the transient gradually progresses up the geometry. The magnitude of the frequencies is not large because the transient is not very drastic. Compared to Figure 3-34, in which the transient is a 20% drop in moderator density, this transient only incurs a 1% change in water density. Furthermore, the geometry is smaller which causes the drop in power to be drastically lower in this case. While Figure 3-29 shows a power drop of 80%, Figure 5-55 shows only an 8% drop in power.

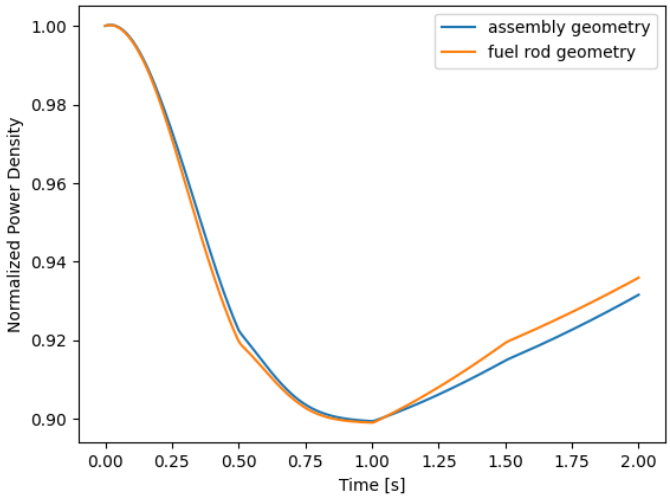

Figure 5-42: Comparison of normalized power density change between identical transients applied to the fuel rod geometry and the assembly geometry.

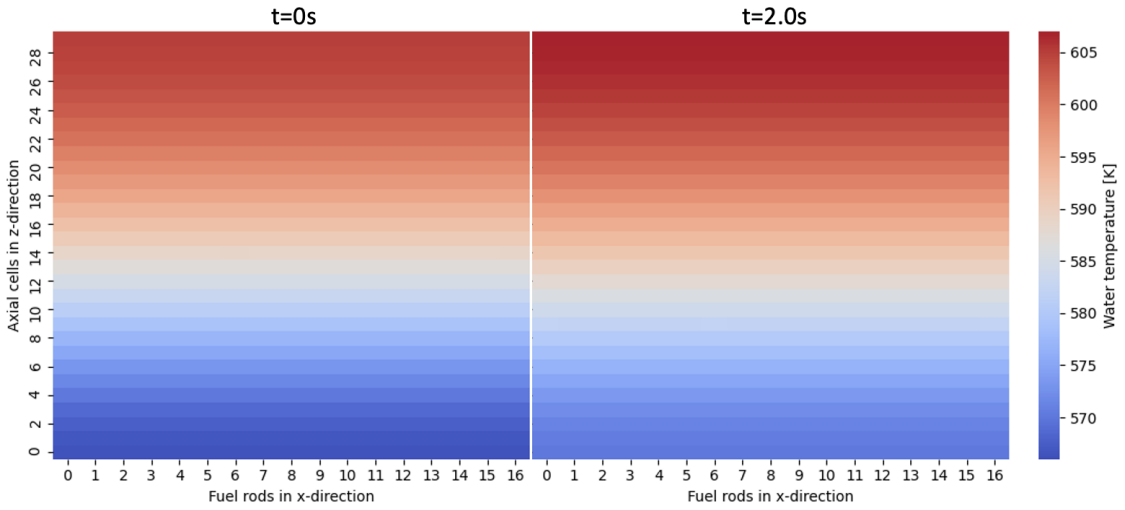

Figure 5-43: Change in water temperature profile in the assembly geometry between t=0s (left) and t=2s (right).

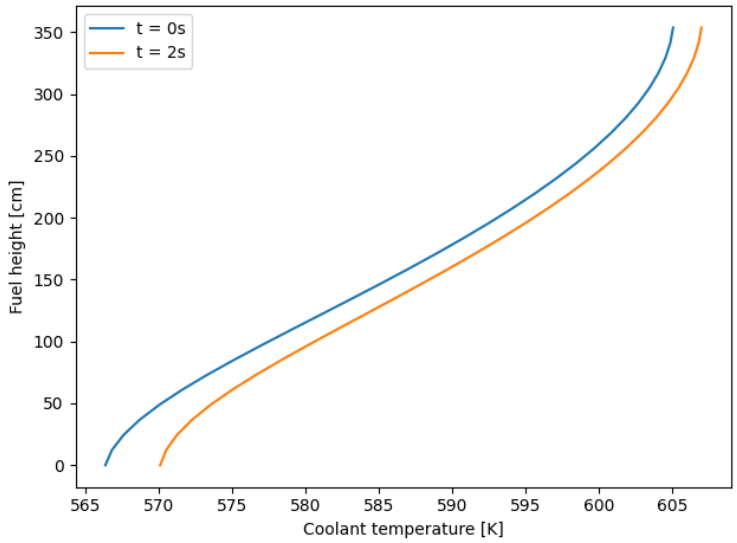

Figure 5-44: Change in water temperature profile in one rod of the assembly geometry between t=0s (left) and t=2s (right).

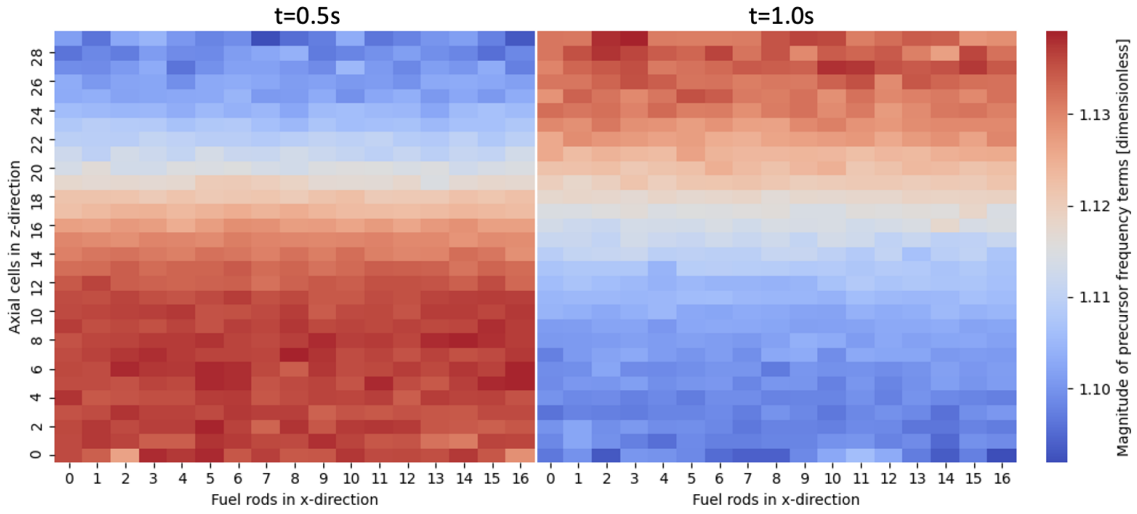

Figure 5-45: Change in precursor frequency terms ($\frac{\lambda}{\lambda+\omega^{D,b}}$) for group 1 delayed neutron precursors in the assembly geometry between t=0.5s (left) and t=1s (right).

5.2.3 Radial effects with inlet temperature transient

Due to limitations in the TD-TH solver, a larger transient simulation is only feasible in this solver up to the saturation point of the water under 15.51 MPa, that is 617 K. Considering an inlet temperature of 566.25, the inlet temperature change could be increased from 570 K to 580 K while maintaining this limit at the outlet condition, as seen in Figure 5-46. This larger inlet change is implemented in the next transient, which also adds a radial component by applying the transient only to the top left-hand corner of the assembly as illustrated in Figure 5-47. Those 64 rods will have their inlet temperature rise from 566.25 K to 580 K instantaneously at t=0. All other rods will be subject to the ensuing power changes, but no cross-flow is modeled between water channels, isolating the direct consequence of the water temperature change to the top left-hand corner. While this is not realistic, it reproduces a similar spatial consequence from a moderator density change in the corner assembly of the C5G7 geometry as studied in Chapter 3. Compared to the C5G7 example, this smaller geometry results in a smaller radial power shift, and the smaller moderator density change will result in a smaller change in frequencies. The calculation is carried out with the same specifications as above, with 0.5s outer time steps, 0.0025 inner time steps, 1100 batches per Monte Carlo run, with 5 million particles per batch.

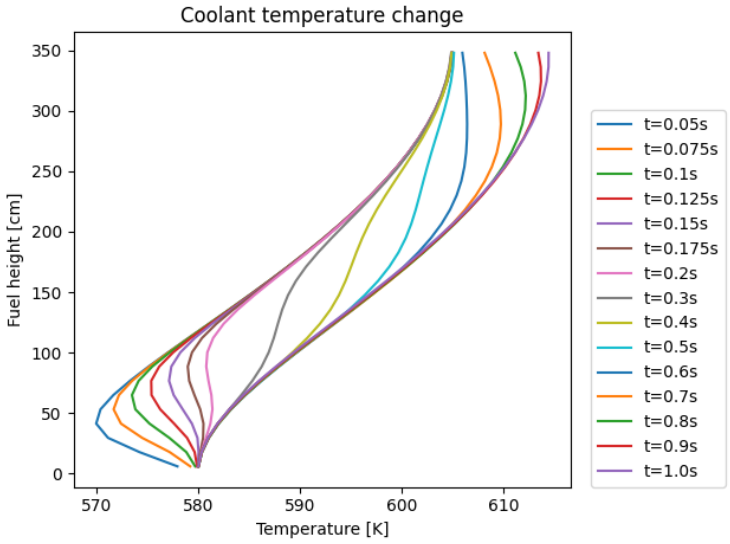

Figure 5-46: Time dependent progression of the coolant temperature during the inlet temperature transient.

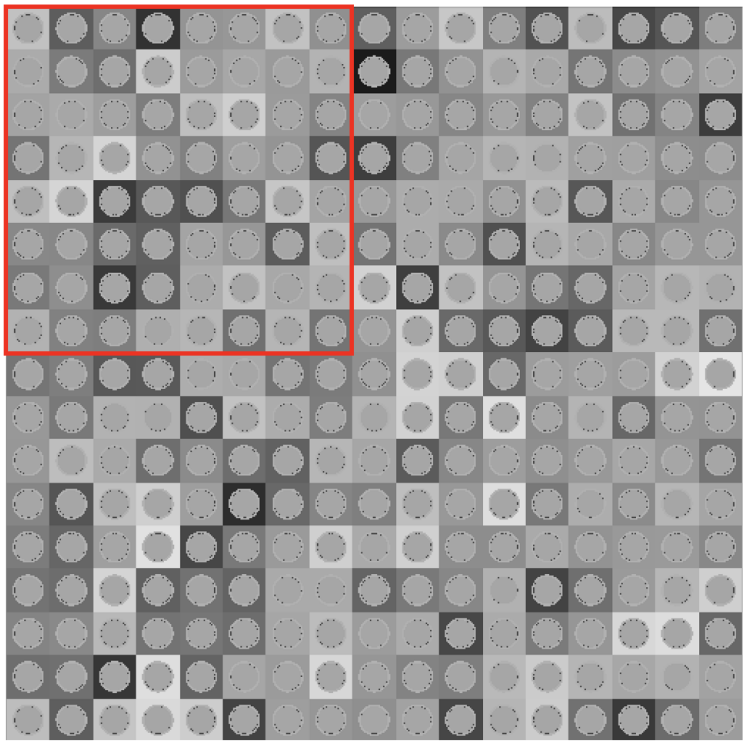

Figure 5-47: Assembly area affected by the inlet temperature change transient: 64 rods in the top left hand corner of the assembly.

The temperature change is shown in Figure 5-48 which highlights the larger temperature shift as compared to Figure 5-43 and the lack of cross flow which allows adjacent rods to have very different temperature profiles.

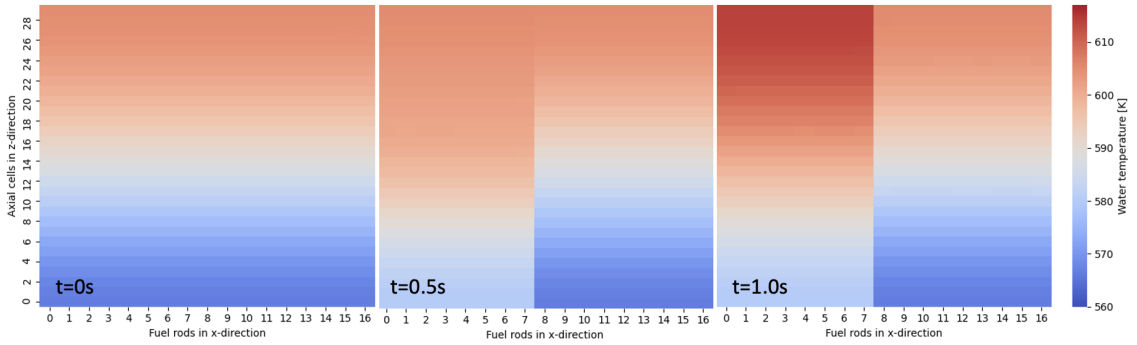

Figure 5-48: Change in water temperature profile in the assembly geometry between t=0s (left), t=0.5s (middle), and t=1.0s (right) in axial view during a quarter-assembly inlet temperature change.

Additionally, the radial power shift from this transient can be observed at several different axial cuts, as shown in Figures 5-49, 5-50, and 5-51. At the lowest axial cut, the power shift can be seen after the first time step, but higher in the assembly, it takes more time for the radial shift to occur because of the time it takes the transient to progress axially up the water channels. Note that the initial power in these plots is not exactly 1.0 at t=0 at these particular axial cuts due to the power profile of the assembly which is shown in Figure 5-52 to have a bottom-bellied axial shape.

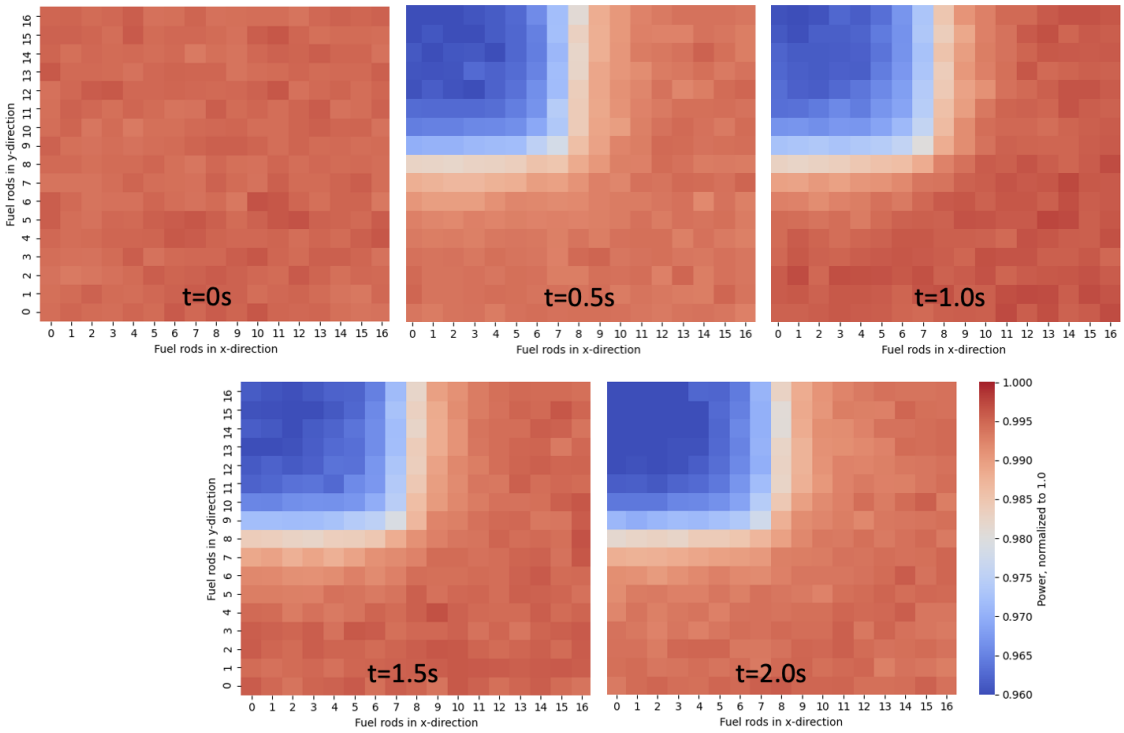

Figure 5-49: Radial shift in power, normalized to unity, at an axial cut at 1/3 the height from the bottom of the assembly for several time points of the transient.

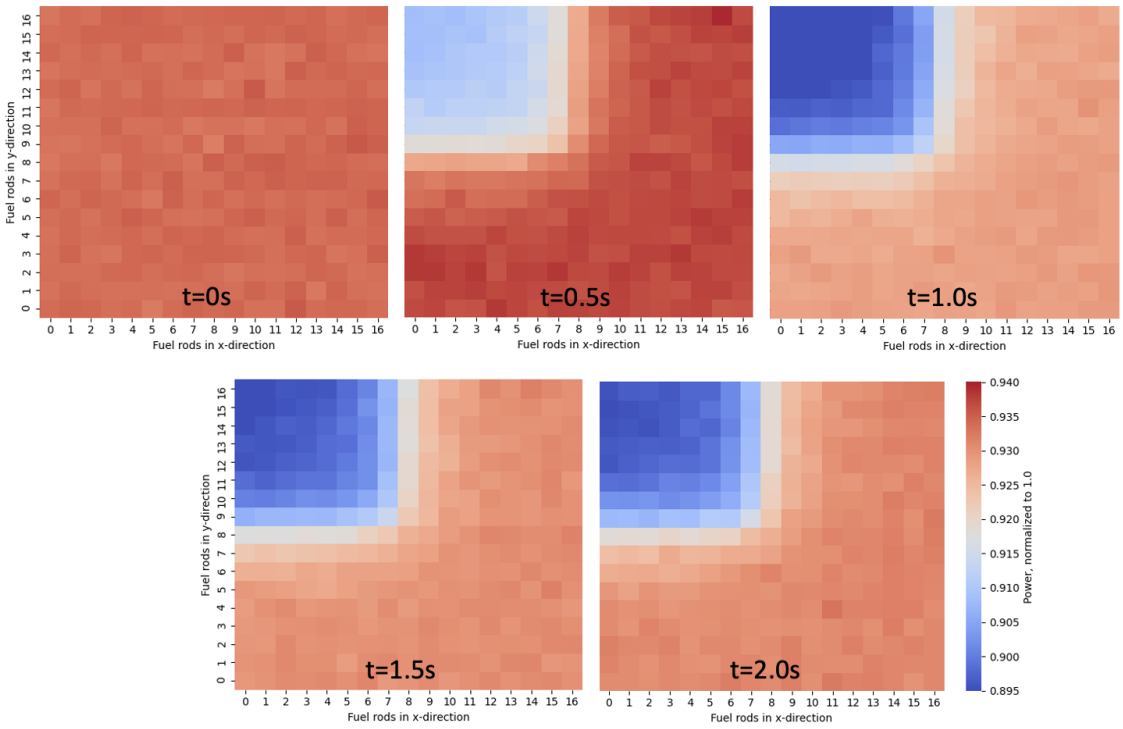

Figure 5-50: Radial shift in power, normalized to unity, at an axial cut at 1/2 the height of the assembly for several time points of the transient.

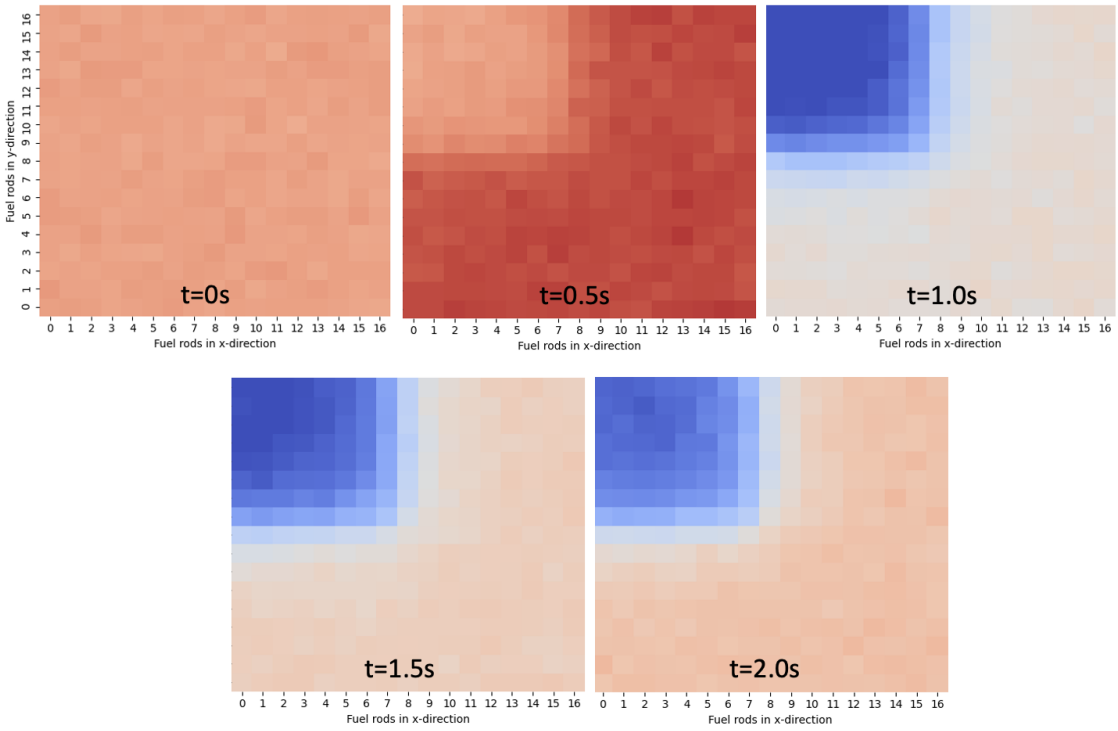

Figure 5-51: Radial shift in power, normalized to unity, at an axial cut at 2/3 the height from the bottom of the assembly for several time points of the transient.

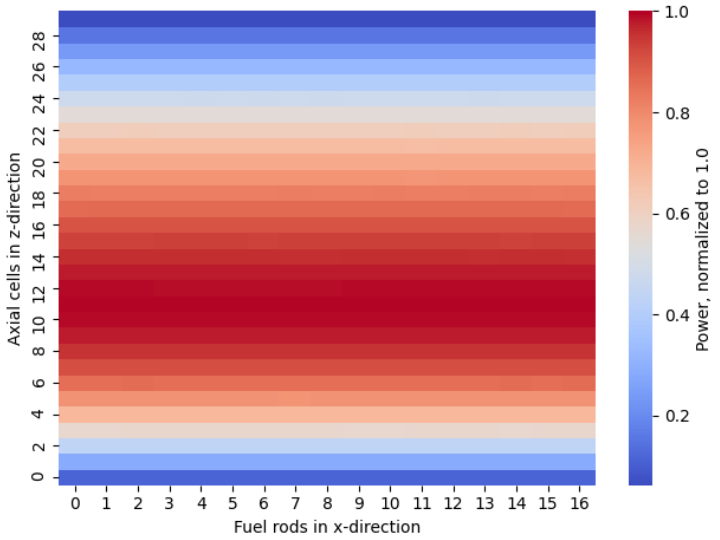

Figure 5-52: Axial power distribution, normalized to unity at t=0.

The precursor frequency terms ($\frac{\lambda}{\lambda+\omega^{D,b}}$) can be seen from the axial direction (same as Figure 5-48) or the radial direction. The axial direction is shown in Figure 5-53 at t=0.5s and t=1s as the transient is progressing up the water channels. The radial direction is shown in Figure 5-54 for each delayed neutron precursor group.

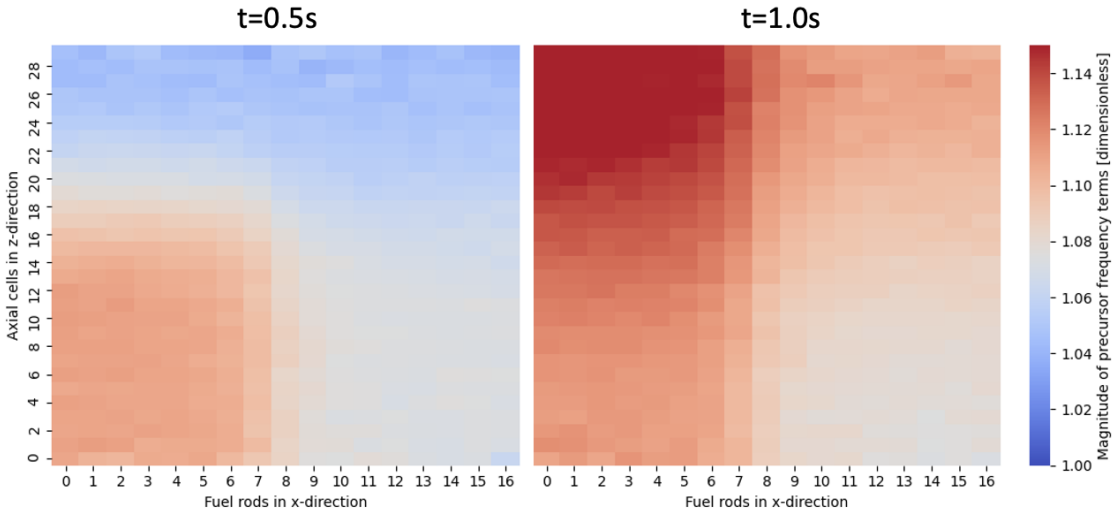

Figure 5-53: Change in precursor frequency terms ($\frac{\lambda}{\lambda+\omega^{D,b}}$) for group 1 delayed neutron precursors in the assembly geometry between t=0.5s (left) and t=1s (right) in axial view.

Despite the larger change in coolant temperature, the fact that the transient is isolated in the corner causes the overall power change of the system to be similar to the smaller inlet change transient, as shown in Figure 5-55. Therefore, the magnitude of the frequencies is not larger in this example than in the previous one. Nonetheless, the frequencies are shown to behave in a manner consistent with Section 3.4. Only the constraints of using a true multiphysics calculation as opposed to a prescribed transient are cause for the differences between the examples in Chapters 5 and 3.

The simplicity of the TD-TH solver constrains the geometry to a uniform rod type which can be improved in the future. The assumption that all rod channel temperatures are independent of each other does however allow each rod's time progression to be run in parallel, greatly cutting down on the overall time of this simulation. As it stands, 600 CPUs (broken down into 15 nodes, each with 40 threads) are used to complete the assembly size transient of 2 seconds long in 45 hours. The Monte Carlo

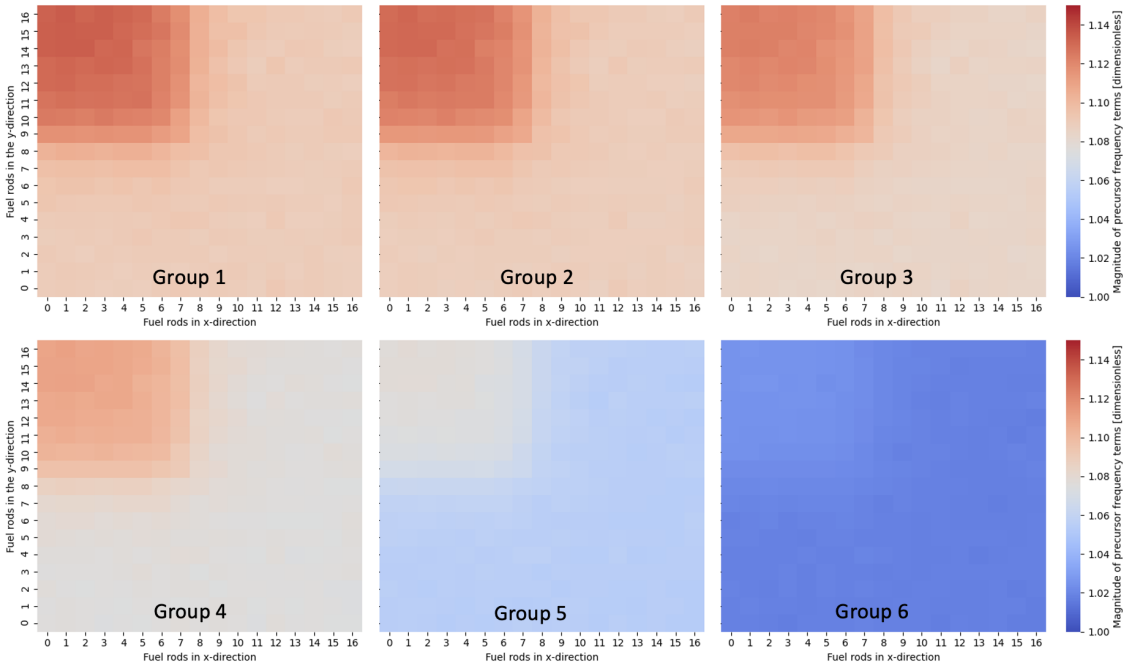

Figure 5-54: Change in precursor frequency terms ($\frac{\lambda}{\lambda+\omega^{D,b}}$) for groups 1-6 in the assembly geometry at t=1s in radial view at an axial cut at 1/2 the height of the assembly.

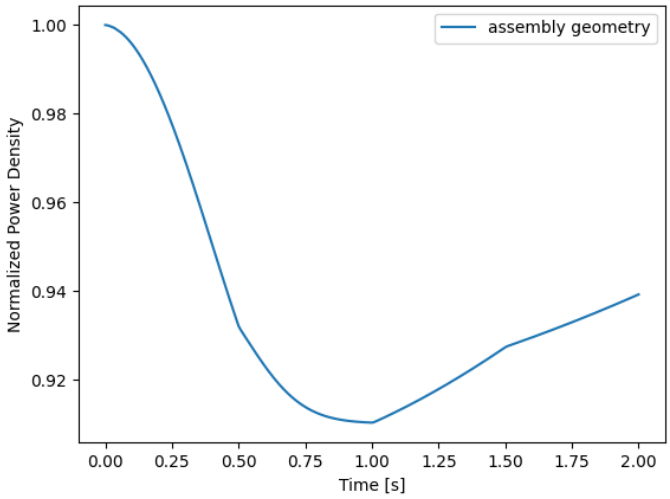

Figure 5-55: Change in normalized assembly power density during the quarter assembly inlet temperature change.

calculations are very well suited to large scale parallelization and efficiently use all 600 CPUs in parallel. Each OpenMC calculation requires approximately 2,400 CPU hours (for 1100 batches, of which 100 are inactive, with 5 million particles per batch),

of which 3 quarters is spent on transport and the rest on synchronizing fission banks. Each TD-TH/CMFD calculation that covers 200 inner time steps only requires 2 CPU hours but is weakly parallelized, using the "multiprocessing" module in Python to multithread a single loop of the TD-TH solver. This effectively only uses 40 of the 600 available CPUs. While this reduces the time to just a few minutes for each TD-TH/CMFD calculation, it could be further improved with MPI parallelization.

With a more sophisticated thermal solver that is more efficiently parallelized, includes adaptive time-stepping, and is more flexible with respect to different materials, this multiphysics calculation could be carried out on more significant problems. However, very high-fidelity time dependent thermal solvers, such as Computational Fluid Dynamics are well known to be a large computational expense which may negate savings otherwise achieved by the high-order/low-order approach to time dependent Monte Carlo neutronics. The work in this thesis rests on the assumption that Monte Carlo neutronics carry the greatest computational burden of the multiphysics procedure. Overall, the methods presented herein allow the use of Monte Carlo for time dependent multiphysics problems where it has previously been too expensive. The multi-scale (high-order/low-order) approach is the key feature of these methods. The coupled TD-TH/CMFD, while central to the numerical scheme is not novel in and of itself. Rather, the use of tally derivatives to enhance the communication between TD-TH and TD-CMFD is the most significant contribution to the multiphysics aspect of this work.

Chapter 6

Conclusions

6.1 Summary of thesis contributions

The goal of this work has been to demonstrate a time dependent, multiphysics, and multi-scale neutronics problem that harnesses the strengths of Monte Carlo neutron transport while mitigating its high computational cost and statistical uncertainty. While the development of time dependent Monte Carlo is theoretically straightforward, and is in fact a natural extension of steady-state Monte Carlo, a practical implementation that can span transients of multiple seconds long has presented challenges, both in run time and memory use. As a solution, an efficient high-order/low-order (HOLO) method called the omega method (the eigenvalue implementation of the frequency transform) was studied and used which allowed the time dependent solution of the neutron transport equation in Monte Carlo. This multi-scale and time dependent numerical scheme has great advantages for spatial resolution over other high-order/low-order methods, as shown in this thesis. The major contribution of this work is incorporating a multiphysics feedback mechanism to this scheme in the form of a time dependent thermal hydraulics solver. This code allows for a flow-initiated transient to alter the temperature distribution of the coolant in a reactor and subsequently change the temperature of surrounding materials, including fuel, and by extension changing power. Altogether, this thesis implements time dependent solvers for thermal hydraulics, neutron diffusion, and Monte Carlo. These work together in

multi-scale communication to efficiently calculate the feedback effects between temperature, density, and power for multiphysics solutions and have been demonstrated in a simplified reactor assembly geometry.

6.1.1 Time dependent

The Monte Carlo solver in this thesis is time dependent by the omega method, derived and implemented throughout Chapter 3. The frequencies are computed by the low-order time dependent Coarse Mesh Finite Difference (TD-CMFD) solver and passed as input to Monte Carlo where they approximate the time derivatives for flux and delayed neutron precursor (DNP) concentrations. Each neutron energy group has a prompt frequency, and each DNP group has a delayed frequency which is also spatially distributed. The DNPs are spatially homogenized in each mesh cell which significantly reduces computational memory requirements compared to the Dynamic Monte Carlo method.

TD-CMFD is an implicit solver of the transient diffusion equations, including prompt and delayed neutron terms, using the analytic solution for the precursor equation described in Chapter 3. Cross sections, diffusion coefficients, and delayed parameters must be tallied in Monte Carlo and used as inputs to the CMFD matrices. However, the HOLO implementation of Monte Carlo implies that these parameters are only tallied at the outer time steps. Therefore, an approximation for their evolution during the inner time steps is needed. A very basic linear-in-time interpolation of cross sections between outer time steps was found to be insufficient to fully resolve multiphysics transients in Chapter 4. Instead, selecting a parameter that follows the evolution of the transient, such as the density of the water in each cell, provided a smoother interpolation of cross sections in addition to a spatially varying interpolation so that not all cross sections have to change at the same rate. This method performed well in comparison to a reference solution with fine outer time steps. Another drawback of cross section interpolation is the need for tallies from the previous and forward outer Monte Carlo time steps, requiring both to be computed before any inner time steps can be solved. This was resolved by the implementation of tally derivatives,

which this thesis first demonstrates between static states in Chapter 4 and then successfully applies to transients in Chapter 5. The primary gain from transient tally derivatives is the ability to solve inner time steps before the forward outer Monte Carlo solution is known, enabling the use of the omega method during the first calculation of the forward outer step. Further usage of tally derivatives was explored to develop a higher order cross section interpolation scheme. Several possible approaches were implemented, including a technique that leverage tally derivatives from both ends of a time step.

The thermal hydraulics solver contains steady-state equations to find the initial temperature distribution of the system as well as time dependent equations (TD-TH), as detailed in Chapter 4. The TD-TH equations collapse back to the steady-state equations during a null transient. The TD-TH includes several approximations. Firstly, the heat transfer regime of steady-state is applied to the entire transient which contradicts recent thermal hydraulics research. However, for the purposes of demonstrating the numerical methods in this thesis, this is not expected to have an impact. Secondly, no cross flow between rod channels is modeled which isolates the temperatures of each channel from its neighbors. This is a physically unrealistic approximation which serves the purposes of this thesis but should be improved for true multiphysics demonstrations.

6.1.2 Multiphysics

The capability to run self-propagating transients, as opposed to fully prescribed transients, stems from the addition of the TD-TH solver to the neutronics solver, introduced in Chapter 4. This multiphysics approach allows feedback effects to guide the evolution of material properties. In the coolant, the dominant effect is from density, since a lower density of water molecules reduces the rate of neutron moderation needed for fission. In the fuel, the Doppler effect broadens the absorption resonances that capture neutrons and inhibit fission as the fuel temperature rises.

At the outer time steps, Monte Carlo responds to a change in temperature as the neutrons re-distribute according to the modified temperature dependent cross sec-

tions. At the inner time steps, TD-TH and TD-CMFD are coupled such that the TD-CMFD power change is a direct input to the TD-TH equations. Furthermore, the use of tally derivatives allows TD-TH to modify the cross sections in TD-CMFD, so that the power change can more closely reflect the feedback effects from the temperature and density changes. This is a major improvement to the TD-TH/CMFD coupling scheme, as without it, the TD-CMFD power cannot change before the forward outer Monte Carlo calculation is complete, as demonstrated in Chapter 5.

Incorporating multiphysics feedback significantly increases the computational cost of time dependent Monte Carlo. Not only is there an expense associated with the addition of thermal hydraulics, but the structure of the Monte Carlo calculation is also affected in such a way as to increase the time and memory usage. Namely, the need to assign independent temperatures and densities to every thermal hydraulics cell in the problem greatly increases the number of variables that need to be generated and updated in the neutronics calculation and constrains the coarseness of the tally mesh.

6.1.3 Multi-scale

A multi-scale approach is at the root of the HOLO method in this thesis which separates the flux into its shape and amplitude components. The basis of this separation is the assumption that amplitude changes more rapidly than shape, and can therefore be calculated at different intervals. The success of this approach is demonstrated throughout Chapter 3.

With this assumption, the Monte Carlo calculations for shape are carried out periodically at outer time steps. Throughout this thesis, 0.5 second intervals have been shown to be sufficient for the problems studied herein. The TD-CMFD calculations for amplitude need to be carried out on finer time steps. Given the extremely small time scale on which neutron diffusion takes place, the stability limits for explicit diffusion solvers are unfeasible. Therefore, TD-CMFD is carried out in an implicit solver which is inherently stable and allows for larger time steps (on the order of a hundredth of a second).

The time scale of the thermal hydraulics solver depends on both the flow rate of the coolant in the axial direction, and the speed of heat transfer through materials in the radial direction. As this calculation is carried out in an explicit solver, the stability limit must be taken into account as was done in Chapter 4, with time steps of 0.0025 seconds being sufficient for stability and accuracy. Since TD-TH and TD-CMFD are solved in tandem throughout the transient, the smaller time step from thermal hydraulics is imposed on both. However, future improvements to this work may be able to find greater efficiencies by de-coupling the time steps.

Furthermore, the axially-varying changes in coolant temperature that drive the transients presented in Chapter 4 reveal the additional challenge of a rapidly-changing shape. As described above, using information about the spatial progression of the transient allowed a more robust interpolation scheme at little added cost in Chapter 5.

Regarding spatial scales, axial thermal hydraulics solvers such as the one implemented in this thesis have previously been shown to be effective with mesh cells on the order of 10 cm in the axial direction. As for neutronics, pin-level accuracy is the goal of many full-core codes, so a pin-level neutronics mesh was employed in this work. However, there exist many methods to use larger spatial meshes to accelerate calculations. In fact, the neutronics HOLO methods used in this thesis are highly conducive to multi-scale spatial meshes. With a coarser tally mesh, important computational savings would be possible if material properties could be mapped to local regions for the thermal solver. However, these options were not explored in this work.

6.2 Future Work

6.2.1 Reactivity-driven transients

The TD-TH solver developed for this work and the specific methods that couple it into the neutronics HOLO method are limited to flow-driven transient applications. As written, the TD-TH solver takes as input flow rate, inlet temperature, and power, making those variables the key transient drivers. In a reactivity-driven transient, the

fuel temperature is expected to drive the transient, and the rate of change is much faster for the fuel temperature than for the coolant temperature. Future work will need to explore how the methods presented herein must be altered to accommodate those scale changes.

6.2.2 Further use of tally derivatives

In addition to the scale of the transient changing during reactivity-driven transients, the cross section interpolation will need to be revisited. This thesis used water density as a marker for the spatial progression of the transient, but the use of a more versatile method will be needed for other applications. In fact, the tally derivative based interpolation explored in Chapter 5 shows promise in this direction. Furthermore, these more sophisticated cross section interpolation schemes are potentially interesting for addressing rod cusping effects in control rod transients. Whereas in this work, methods were developed that greatly improved axially varying cross section changes, a control rod movement results in much larger cross section variations with large repercussions in power. Therefore, the methods developed in this thesis need to be tested under those more extreme conditions.

While this work has made the important demonstration of tally derivative use in a transient scenario, their rigorous analysis is limited primarily to projecting the TD-CMFD parameters throughout the inner time steps before the forward outer Monte Carlo solution is known. Further work is needed to fully understand how they can benefit transient multiphysics problems.

6.2.3 High fidelity thermal hydraulics

Finally, the methods in this thesis assume that Monte Carlo neutronics is the computational bottleneck of the calculation. However, if high fidelity thermal hydraulics are used, such as Computational Fluid Dynamics (CFD), this component would compete with Monte Carlo as the most costly part of the calculation. New methods would be needed to create more cost-effective coupling, although similarities might be re-

tained. For instance, CFD/MC coupling has been widely explored for steady-state applications. By extension, CFD/MC coupling at outer time steps may be feasible with low-order TD-TH and TD-CMFD coupling at inner time steps. This would greatly increase the cost of the outer time steps, but employ HOLO methods for both neutronics and thermal hydraulics, mirroring the overall approach of this thesis.

Appendix A

Description of the 2D LRA Transient Benchmark

The benchmark includes 2 energy group data and 2 delayed neutron precursor group data. Regions 1, 2, 3, and 4 in Figure A-1 are fuel regions with different cross sections. The rod withdrawal is modeled by a linear reduction in the thermal absorption cross section in region "R" for the first two seconds of the transient, see equation (A.1). Adiabatic heatup is modeled based on the change in temperature and power, see equations (A.2) and (A.3). Doppler feedback is simulated as a change in fast absorption cross section in all fuel regions, see equation (A.4). The two inner sides of the geometry have reflective boundary conditions, and the two outer sides of the core have zero flux boundary conditions.

The rod withdrawal is prescribed as:

$$\Sigma_{a2,R}(t) = \begin{cases} \Sigma^0_{a2,3} \cdot (1 - [0.0606184 \cdot t]) & \text{if } t \leq 2\text{s} \\ \Sigma^0_{a2,3} \cdot (1 - [0.0606184 \cdot 2]) & \text{if } t \geq 2\text{s} \end{cases} \tag{A.1}$$

The adiabatic heatup and power equations are:

$$\frac{\partial T(t,\vec{x})}{\partial t} = \alpha \sum_{g'=1}^{2} \Sigma_{f,g'} \Psi_{g'}(t,\vec{x}) \tag{A.2}$$

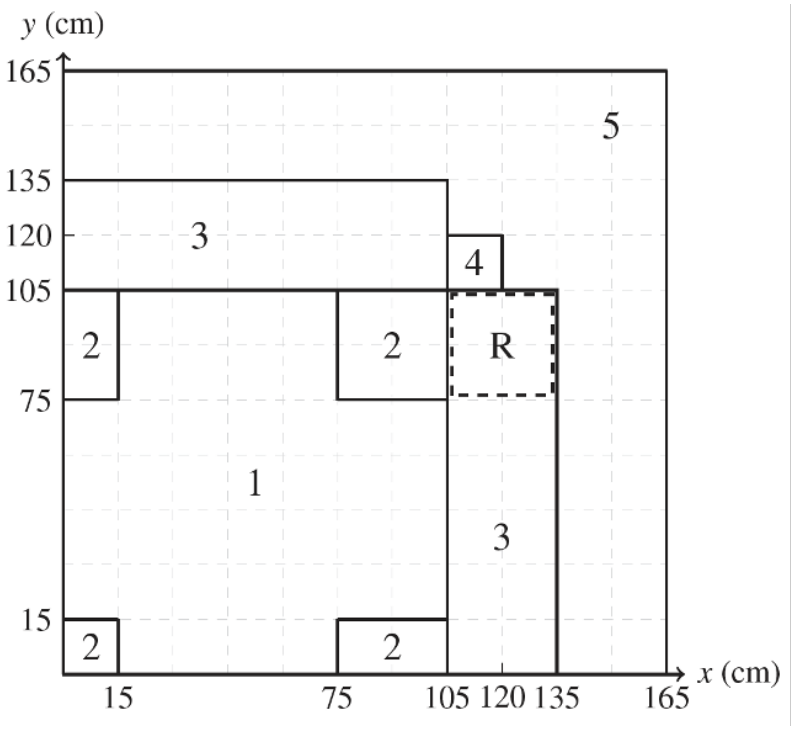

Figure A-1: 2D LRA geometry.

$$P(t,\vec{x}) = \kappa \sum_{g'=1}^{2} \Sigma_{f,g'} \Psi_{g'}(t,\vec{x}) \tag{A.3}$$

The Doppler feedback is modeled as:

$$\Sigma_{a,1}(T) = \Sigma_{a1}^{0}[a + \gamma(\sqrt{T} - \sqrt{T^0})] \tag{A.4}$$

where α, κ, γ are specified in the benchmark along with other relevant reactor-wide parameters. Note that most published solutions use a slightly modified benchmark whereby the Doppler feedback coefficient, γ is $3.034 \times 10^{-3}\ K^{-1/2}$ (instead of $2.034 \times 10^{-3}\ K^{-1/2}$) and the decay constant for the first delayed neutron precursor group is $0.0654\ s^{-1}$ (instead of $0.00654 s^{-1}$) [103]. These modified values are used in this paper for consistency with other published results.

Appendix B

Detailed derivation of the analytic precursor solution for the frequency transform method

The analytical solution to the precursor equation assumes that the time variation of the flux shape is linear over the time step, that all coefficients and frequencies are constant over the time step, and that β and λ are independent of fissioning group:

$$\frac{dC_i(t)}{dt} = -\lambda_i C_i(t) + e^{\omega t}\frac{\beta_i}{k_{crit}}\sum_{g'}\nu_{g'}\Sigma_{fg'}\psi_{g'}(t) \tag{B.1}$$

$$\psi_{g'}(t) = \psi_{g'}^n + \frac{t-t^n}{\Delta t}[\psi_{g'}^{n+1} - \psi_{g'}^n] \tag{B.2}$$

Substituting the linear shape assumption into the precursor equation with $t^n = 0$:

$$\frac{dC_i(t)}{dt} = -\lambda_i C_i(t) + e^{\omega t}\frac{\beta_i}{k_{crit}}\sum_{g'}\nu_{g'}\Sigma_{fg'}\left[\psi_{g'}^n + t\frac{\psi_{g'}^{n+1} - \psi_{g'}^n}{\Delta t}\right] \tag{B.3}$$

Grouping terms together into intermediate variables a and b for readability:

$$\frac{dC_i(t)}{dt} = -\lambda_i C_i(t) + e^{\omega t}\frac{\beta_i}{k_{crit}}\sum_{g'}\nu_{g'}\Sigma_{fg'}\psi_{g'}^n + te^{\omega t}\frac{\beta_i}{k_{crit}}\sum_{g'}\nu_{g'}\Sigma_{fg'}\left[\frac{\psi_{g'}^{n+1} - \psi_{g'}^n}{\Delta t}\right] \tag{B.4}$$

$$\frac{dC_i(t)}{dt} = -\lambda_i C_i(t) + ae^{\omega t} + bte^{\omega t} \tag{B.5}$$

Solving this ODE with the initial condition $C_i(0) = C_i^n$:

$$C_i(t) = C_i^n e^{-\lambda_i t} + \frac{e^{-\lambda_i t}}{\lambda_i + \omega}\left[ae^{(\lambda_i+\omega)t} - a + bte^{(\lambda_i+\omega)t} - b\frac{e^{(\lambda_i+\omega)t}}{\lambda_i+\omega} + \frac{b}{\lambda_i+\omega}\right] \tag{B.6}$$

Replacing a and b with their original expressions, factoring out the common terms $\frac{\beta_i}{k_{crit}}$ and $\nu_g \Sigma_{fg}$:

$$\begin{aligned} C_i(t) = C_i^n e^{-\lambda_i t} + \frac{\beta_i e^{-\lambda_i t}}{k_{crit}(\lambda_i+\omega)} \sum_{g'} \nu'_g \Sigma_{fg'} \Bigg[\psi_{g'}^n e^{(\lambda_i+\omega)t} - \\ \psi_{g'}^n + \frac{\psi_{g'}^{n+1} - \psi_{g'}^n}{\Delta t} te^{(\lambda_i+\omega)t} - \frac{\psi_{g'}^{n+1} - \psi_{g'}^n}{\Delta t}\frac{e^{(\lambda_i+\omega)t}}{\lambda_i+\omega} + \frac{\psi_{g'}^{n+1} - \psi_{g'}^n}{\Delta t}\frac{1}{\lambda_i+\omega}\Bigg] \end{aligned} \tag{B.7}$$

Grouping together terms multiplied by ψ_g^n and ψ_g^{n+1} respectively:

$$\begin{aligned} C_i(t) = C_i^n e^{-\lambda_i t} \\ + \frac{\beta_i e^{-\lambda_i t}}{k_{crit}(\lambda_i+\omega)} \sum_{g'} \nu_{g'} \Sigma_{fg'} \Bigg[\psi_{g'}^n \left(e^{(\lambda_i+\omega)t} - 1 - \frac{t}{\Delta t}e^{(\lambda_i+\omega)t} + \frac{e^{(\lambda_i+\omega)t}}{(\lambda_i+\omega)\Delta t} - \frac{1}{(\lambda_i+\omega)\Delta t}\right) \\ + \psi_{g'}^{n+1}\left(\frac{te^{(\lambda_i+\omega)t}}{\Delta t} - \frac{e^{(\lambda_i+\omega)t}}{(\lambda_i+\omega)\Delta t} + \frac{1}{(\lambda_i+\omega)\Delta t}\right)\Bigg] \end{aligned} \tag{B.8}$$

Multiplying through by $\frac{e^{(\lambda_i+\omega)t}}{e^{(\lambda_i+\omega)t}}$ to transform $e^{-\lambda_i t}$ into $e^{\omega t}$ as the common factor of the second term:

$$\begin{aligned} C_i(t) = C_i^n e^{-\lambda_i t} \\ + \frac{\beta_i e^{\omega t}}{k_{crit}(\lambda_i+\omega)} \sum_{g'} \nu_{g'} \Sigma_{fg'} \Bigg[\psi_{g'}^n \left(1 - e^{-(\lambda_i+\omega)t} - \frac{t}{\Delta t} + \frac{1}{(\lambda_i+\omega)\Delta t} - \frac{e^{-(\lambda_i+\omega)t}}{(\lambda_i+\omega)\Delta t}\right) \\ + \psi_{g'}^{n+1}\left(\frac{t}{\Delta t} - \frac{1}{(\lambda_i+\omega)\Delta t} + \frac{e^{-(\lambda_i+\omega)t}}{(\lambda_i+\omega)\Delta t}\right)\Bigg] \end{aligned} \tag{B.9}$$

Evaluating the expression at $C_i(\Delta t) = C^{n+1}$:

$$C_i^{n+1} = C_i^n e^{-\lambda_i \Delta t} + \frac{\beta_i e^{\omega \Delta t}}{k_{crit}(\lambda_i + \omega)} \sum_{g'} \nu_{g'} \Sigma_{fg'} \left[\psi_{g'}^n \left(1 - e^{-(\lambda_i+\omega)\Delta t} - 1 + \frac{1}{(\lambda_i+\omega)\Delta t} - \frac{e^{-(\lambda_i+\omega)\Delta t}}{(\lambda_i+\omega)\Delta t} \right) + \psi_{g'}^{n+1} \left(1 - \frac{1}{(\lambda_i+\omega)\Delta t} + \frac{e^{-(\lambda_i+\omega)\Delta t}}{(\lambda_i+\omega)\Delta t} \right) \right] \tag{B.10}$$

Re-arranging terms:

$$C_i^{n+1} = C_i^n e^{-\lambda_i \Delta t} + \frac{\beta_i e^{\omega \Delta t}}{k_{crit}(\lambda_i + \omega)} \sum_{g'} \nu_{g'} \Sigma_{fg'} \left[\psi_{g'}^n \left(- e^{-(\lambda_i+\omega)\Delta t} + \frac{1 - e^{-(\lambda_i+\omega)\Delta t}}{(\lambda_i+\omega)\Delta t} \right) + \psi_{g'}^{n+1} \left(1 - \frac{1 - e^{-(\lambda_i+\omega)\Delta t}}{(\lambda_i+\omega)\Delta t} \right) \right] \tag{B.11}$$

Appendix C

Detailed Description of Flux Factorization in the OpenMC Implementation of the Omega Method

In order to provide reproducibility of the work in this thesis, a detailed description of flux factorization, frequency definitions, and solution procedures for the High-Order/Low-Order implementation of the Omega Method in Monte Carlo is presented. Since this aspect of the work was previously done in another thesis, much of the contents of this appendix are taken directly from that reference by Shaner [76]. Quotation marks are used to indicate direct text references. This aim of this appendix is to provide clarification and highlight the differences between this thesis and the work implemented in [76].

The time dependent neutron transport equation is derived in [76] as:

$$\frac{1}{v}\frac{\partial \boldsymbol{\Psi}(r,E,\boldsymbol{\Omega},t)}{\partial t} = -\nabla\cdot\boldsymbol{\Omega}\boldsymbol{\Psi}(r,E,\boldsymbol{\Omega},t) - \Sigma_t(r,E,t)\boldsymbol{\Psi}(r,E,\boldsymbol{\Omega},t)$$
$$+\int_{4\pi} d^2\boldsymbol{\Omega}' \int_0^\infty dE'\nu_s\Sigma_s(r,E\leftarrow E',\boldsymbol{\Omega}\leftarrow\boldsymbol{\Omega}',t)\boldsymbol{\Psi}(r,E',\boldsymbol{\Omega}',t)$$
$$+\frac{1}{4\pi}\chi^P(r,E,t)\int_0^\infty dE'\nu_f\Sigma_f^P(r,E',t)\Phi(r,E',t) + \frac{1}{4\pi}\sum_{b=1}^{B}\chi^{D,b}(r,E,t)\lambda^b(r,t)C^b(r,t) \tag{C.1}$$

Coupled with the delayed neutron precursor balance equation:

$$\frac{\partial C^b(r,t)}{\partial t} = \int_0^\infty dE\nu_f\Sigma_f^{D,b}(r,E,t)\Phi(r,E,t) - \lambda^b(r,t)C^b(r,t) \tag{C.2}$$

where r is position, $\boldsymbol{\Omega}$ is direction, v is velocity, E is energy, t is time, $\boldsymbol{\Psi}$ is angular flux, Φ is scalar flux, Σ_t is the macroscopic total cross section, $\nu_s\Sigma_s$ is the macroscopic scattering cross section, $\nu_f\Sigma_f$ is the macroscopic fission cross section, B is the number of delayed neutron precursor groups, C is the delayed neutron precursor concentration, χ^P is the prompt neutron emission spectrum, χ^D is the delayed neutron emission spectrum, and λ is the delayed neutron precursor group decay constant.

The original implementation utilized four separate spatial meshes: a geometry mesh, a fine mesh, a coarse mesh, and a global mesh described as follows. "The geometry mesh contains the full spatial detail of the materials using constructive solid geometry to define region boundaries. The MC method will be solved on the geometry mesh over large time intervals (i.e. outer steps) and used to compute MGXS on the fine mesh. For consistency, this solve will be referred to as the MC or outer solve throughout. In all cases the fine mesh will use a pin-cell mesh in the radial direction in order to facilitate pin-cell delayed neutron precursor concentrations. The coarse mesh is a cartesian mesh overlaid on the geometry mesh that will be used to propagate the flux amplitude forward in time. Since delayed neutron precursors cannot migrate from cell to cell (assuming a solid fuel), the precursors in each cell can be propagated on the fine mesh level without incurring significant computational costs even for large problems. The TCMFD diffusion equation will be used to propagate the

amplitude forward over short time intervals (i.e. inner steps) on the coarse mesh. The cell-to-cell currents and diffusion coefficients will be tallied on the coarse mesh. For consistency, this solve will be referred to as the TCMFD or inner solve throughout. In order to obtain the fine mesh flux solution that comes out of the MC solve, a fine mesh shape is defined and updated after each MC solve. In between MC solves, the fine mesh shape is interpolated and used in condensing the MGXS down to the coarse mesh." Finally, "the global mesh is used to approximate some parameters that are either spatially flat or hard to implement with spatial dependence in a MC solve."

In this thesis, the coarse mesh and fine mesh are identical to each other to provide one-to-one communication between the thermal hydraulics solver and the coarse mesh finite difference diffusion solver on a pin-cell mesh.

The HOLO methods in this thesis rely on the separability of the flux into components that have varying dependence on space and time. The outer time steps are chosen such that a "MC solve is performed over long time intervals, Δt^s, to compute MGXS on the fine mesh and cell-to-cell currents on the coarse mesh. At the initial time step, the space, angle, and energy integrated MGXS are used to compute the fine mesh powers. The fine mesh powers are then normalized to a user input value and the delayed neutron precursor concentrations are computed. The shape-weighted MGXS on the fine mesh are collapsed down to the coarse mesh and interpolated on time intervals Δt^a. When MGXS and the flux shape are requested at a time point in between MC solves, these parameters are interpolated from the most recent MC solves."

In this thesis, Section 5.1.2 dives into the problems associated with the linear interpolation of the original implementation, which worked well when the transient was axially homogeneous. The issues arise when an axial heterogeneity is modeled.

As noted by Shaner in the context of control rod movements: “The attempt to apply simple models to interpolate MGXS and coupling terms between widely spaced shape function solves will result in an unphysical phenomena known as control rod cusping, whereby the reactivity and power profiles with be characterized by unphysical cusps. Much work has gone into resolving this problem for nodal diffusion and indirect

transport theory based transient codes that utilize low-order operator representations of the problem with spatially homogenized MGXS. In this work we have not chosen to investigate any procedures to correct for the cusping problem."

This thesis does not analyze reactivity-initiated transients such as control rod movement, but in the context of a relatively mild flow-initiated transient, this thesis addresses a similar concern due to axial variations.

For the inner time steps, the "flux amplitude and precursor concentration are then propagated forward over the Δt^a time interval. After both the inner and outer solves, the fine mesh powers are computed and checked for convergence to a user input value. The use of fine mesh cross sections allow for feedback to be conducted on the fine mesh during the inner solves. However, this is seen as a large effort in and of itself and is therefore recommended for future work."

This thesis brings this suggestion into the forefront by including feedback into this time dependent neutronics scheme throughout Chapters 4 and 5.

C.1 Frequency Transform of the Flux and Precursor Time-Derivatives

While Shaner's work refers to the Monte Carlo application of the frequency transform as the Frequency Transform Method, this thesis has chosen to use the alternative name of Omega Method as much as possible. This is to eliminate confusion between the frequency transform applications to eigenvalue problems (where Omega Method is more commonly used) and time-differencing problems (where Frequency Transform Method is more common). In quoting Shaner's text, the Frequency Transform Method, Omega Method, ω-mode method, or exponential transform all refer to the same methodology. As derived in Chapter 3,

$$\frac{1}{v}\frac{\partial \mathbf{\Psi}(r,E,\mathbf{\Omega},t)}{\partial t} = \frac{1}{v}\omega^P(r,E,\mathbf{\Omega},t)\cdot\mathbf{\Psi}(r,E,\mathbf{\Omega},t) \tag{C.3}$$

$$\frac{\partial C^b(r,t)}{\partial t} = \omega^{D,b}(r,t) \cdot C^b(r,t) \tag{C.4}$$

where $\omega^P(r, E, \mathbf{\Omega}, t)$ represents the flux (or prompt) frequency and $\omega^{D,b(r,t)}$ represents the precursor (or delayed) frequencies. Inserting the frequency transformed time-derivatives into the time-dependent neutron transport equation:

$$\begin{aligned}\frac{1}{v}\omega^P(r,E,\mathbf{\Omega},t)\cdot\mathbf{\Psi}(r,E,\mathbf{\Omega},t) &= -\nabla\cdot\mathbf{\Omega}\mathbf{\Psi}(r,E,\mathbf{\Omega},t) - \Sigma_t(r,E,t)\mathbf{\Psi}(r,E,\mathbf{\Omega},t) \\ &+ \int_{4\pi} d^2\mathbf{\Omega}' \int_0^\infty dE'\nu_s\Sigma_s(r,E\leftarrow E',\mathbf{\Omega}\leftarrow\mathbf{\Omega}',t)\mathbf{\Psi}(r,E',\mathbf{\Omega}',t) \\ +\frac{1}{4\pi}\frac{\chi^P(r,E,t)}{k_{eff}^{t0}}\int_0^\infty dE'\nu_f\Sigma_f^P(r,E',t)\Phi(r,E',t) &+ \frac{1}{4\pi}\sum_{b=1}^{B}\chi^{D,b}(r,E,t)\lambda^b(r,t)C^b(r,t)\end{aligned} \tag{C.5}$$

Coupled with the delayed neutron precursor balance equations:

$$\omega^{D,b}(r,t)\cdot C^b(r,t) = \frac{1}{k_{eff}^{t0}}\int_0^\infty dE\nu_f\Sigma_f^{D,b}(r,E,t)\Phi(r,E,t) - \lambda^b(r,t)C^b(r,t) \tag{C.6}$$

To distinguish this implementation of the Omega Method from the alpha-static eigenvalue method, it "should be made clear that the flux and precursor frequencies are known quantities that are *inserted* into the simulation when performing the MC solve, not values to be solved for."

C.2 Approximation of the Flux and Precursor Frequencies

The HOLO implementation requires a high level of consistency between the equations used for the shape and amplitude function solves. The following text clarifies how this is achieved. "When MC is used, consistency between the tallied scalar flux and the flux solution computed via nonlinear CMFD diffusion equation is only guaranteed in the infinite particle limit. [...] In the infinite particle limit, the tallied scalar flux and the nonlinear CMFD diffusion solution will converge, but this convergence is both

very slow and unnecessary. To reconstruct an approximation for the fine mesh flux between MC solves, a fine mesh shape function is defined as the difference between the MC solution and the amplitude at an outer time step. The fine mesh shape is integrated over both fine mesh cells and a coarse energy group structure. This flux factorization procedure separates the neutron flux into components that reside on the geometry, fine, and coarse meshes. These components allow for the full space, angle, and energy dependent flux to be reconstructed at any point in time."

In summary, the flux, shape, and amplitude are calculated as follows:

1. After the first Monte Carlo solve at $t = 0$, the CMFD matrix is filled with reaction rates (rather than cross sections as is traditionally done) as the product of tallied MGXS and tallied flux. The result of the CMFD calculation is called the amplitude which is a vector of values near unity (unity can be achieved only in the infinite particle limit).

2. The CMFD shape is defined as the tallied flux normalized to the user input power.

3. After subsequent Monte Carlo solves at $t \neq 0$, the shape is set to the tallied flux divided by the amplitude but retains its initial normalization. The amplitude, however, carries the changing power level which is incorporated in the flux when flux is defined as the product of shape and amplitude.

To further illustrate this workflow, Figures C-1 and C-2 are provided that show the shape at t=0 and the amplitude at the first inner time step (t=0.0025).

The frequencies at time t^a can then be approximated as:

$$\omega^{P,g,t^a} = \frac{1}{\Delta t^a} log \frac{\Phi^{g,t^a}}{\Phi^{g,t^{a-1}}} \tag{C.7}$$

$$\omega^{D,b,t^a} = \frac{1}{\Delta t^a} log \frac{C^{b,t^a}}{C^{b,t^{a-1}}} \tag{C.8}$$

where $\Phi^{g,t}$ is the space, angle, and energy integrated time dependent fine mesh flux.

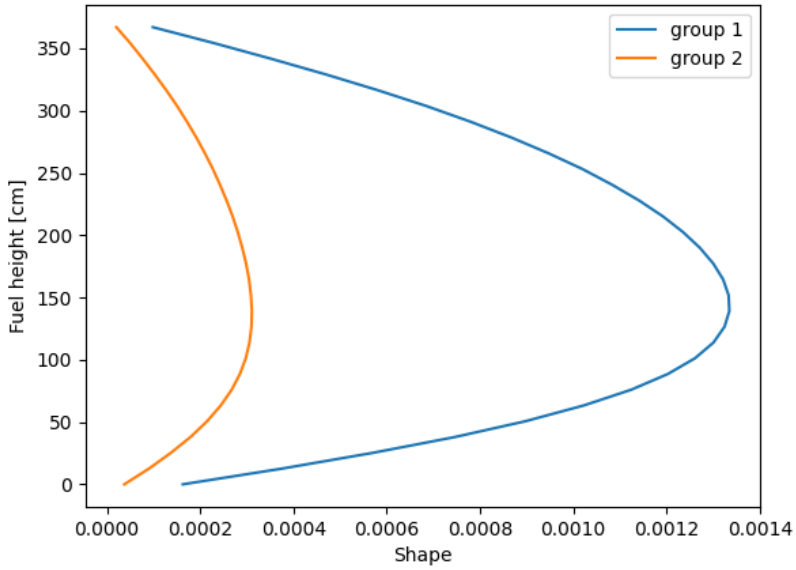

Figure C-1: Plot of initial shape at the first outer time step.

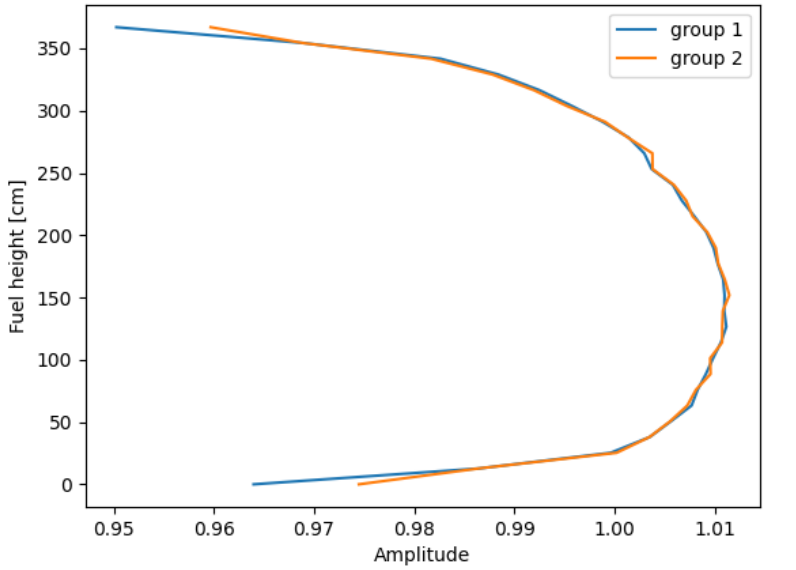

Figure C-2: Plot of initial amplitude at the first inner time step.

As Shaner showed to be practical and accurate, "we will approximate the flux frequency to be spatially flat [...] The final form of the time dependent neutron transport equation that will be solved with Monte Carlo is then:"

$$\frac{1}{v}\omega^{P,g,t}\mathbf{\Psi}(r,E,\mathbf{\Omega},t) = -\nabla\cdot\mathbf{\Omega}\mathbf{\Psi}(r,E,\mathbf{\Omega},t) - \Sigma_t(r,E,t)\mathbf{\Psi}(r,E,\mathbf{\Omega},t)$$
$$+\int_{4\pi} d^2\mathbf{\Omega}' \int_0^\infty dE'\nu_s\Sigma_s(r,E\leftarrow E',\mathbf{\Omega}\leftarrow\mathbf{\Omega}',t)\mathbf{\Psi}(r,E',\mathbf{\Omega}',t)$$
$$+\frac{1}{4\pi}\frac{\chi^P(r,E,t)}{k_{eff}^{t0}k_{eff}^t}\int_0^\infty dE'\nu_f\Sigma_f^P(r,E',t)\Phi(r,E',t) \tag{C.9}$$
$$+\frac{1}{4\pi}\sum_{b=1}^{B}\frac{\chi^{D,b}(r,E,t)}{k_{eff}^{t0}k_{eff}^t}\frac{\lambda^{b,t,r}}{\omega^{D,b,t,r}+\lambda^{b,t,r}}\int_0^\infty dE'\nu_f\Sigma_f^{D,b}(r,E',t)\Phi(r,E',t)$$

"To summarize, the following approximations have been applied [...]:

- The flux is assumed to be separable into components on the geometry mesh, fine mesh, and coarse mesh (Space-time factorization approximation).

- The flux and precursor concentrations are assumed to have an exponential frequency (Frequency Transform approximation).

- The flux frequency is approximated using the multi-group flux frequency on the global mesh.

- The precursor frequency is approximated using the precursor frequency on the fine mesh.

- It is assumed that the k eigenvalue will not significantly bias the flux distribution due to the incorporation of space and energy dependent flux and precursor frequencies."

Further detail on the amplitude function solve is provided by Shaner. "An amplitude function is used to maintain accuracy while increasing the time step size between MC solves." The factorized flux is inserted into the time dependent fine mesh balance equation and precursor balance equations. Then, the time-derivative terms must be approximated. The Improved Quasi Static approximation "uses current information about the shape function to approximate its derivative. Since the shape function at

the forward and previous MC solve time points will be known (or can be approximated), the IQS approximation will be applied in this work. The time-derivative of the shape at time t^a is then:"

$$\frac{\psi^{g,t^a}}{\partial t} = \frac{\psi^{g,t^s} - \psi^{g,t^{s-1}}}{\Delta t^s} \tag{C.10}$$

"The amplitude derivative is approximated using a backward difference approximation:"

$$\frac{\Phi^{g,t^a}}{\partial t} = \frac{\Phi^{g,t^a} - \Phi^{g,t^{a-1}}}{\Delta t^s} \tag{C.11}$$

"After each MC solve, the fine mesh shape is defined to be the cell-averaged tallied MC flux divided by the current cell-averaged amplitude. Therefore the coarse mesh flux will be identical to the coarse mesh flux tallies from the MC solve."

The final equations yield "a linear system of equations (i.e. Ax = b) where the left hand side is dependent on the amplitude at the previous time step (i.e. the b term) and the right hand side is dependent on the amplitude at the current time step (i.e. the x term). The cross sections, time-derivative terms, cell to cell coupling terms, and delayed neutron precursor decay terms that depend on the amplitude at the current time step combine to form the linear operator (i.e. the A term). This linear system of equations can them be solved in many different ways. In our implementation, we perform all MGXS data handling and propagation of the amplitude function in Python and make heavy use of the NumPy and SciPy packages. The linear system of equations is solved with SciPy spsolve(...) method, which uses the UMFPACK Unsymmetric MultiFrontal method for solving the Ax = b problem."

The very first Monte Carlo solve, which is a steady-state calculation at t=0 sets the stage for the rest of the calculation. "The initial solution is found by solving a MC eigenvalue problem to obtain the initial flux, MGXS on the fine mesh, and cell-to-cell currents on the coarse mesh. The initial flux shape is defined as the flux term. [...] The steady-state version of the time dependent nonlinear CMFD diffusion equation [...] is then solved to obtain the initial amplitude solution, which will converge to

unity in the infinite particle limit. [...] In this work, we apply the power method for the outer source iteration with the SciPy spsolve(...) method used to solve the inner linear system of equations. In all simulations in this work the RMS error in the cell-averaged fission source is converged to a residual of 1.e-6. [...] The eigenvalue from the low-order solve is needed to ensure the initial solution holds steady state. [...] [The] coarse mesh amplitude will not correspond to any real power level. Therefore, the initial flux shape will be normalized to a user defined initial core power. [...] The shape function following any time dependent MC solve is then defined as the normalized flux tallied from the MC solve divided by the amplitude."

C.3 Frequency Treatment in Monte Carlo

To incorporate the frequencies into Monte Carlo as input parameters, as mentioned above, the standard sampling technique needs to be modified. In order to do so, it is helpful to think of the flux frequency "as a prompt neutron hysteresis term as it accounts for the instantaneous change in the flux that is dictated by changes in the prompt neutron production rate or material properties. [...] The frequency should not be thought of as a reaction, but rather the response of the flux to time dependent changes in the actual reaction rates. [...] Monte Carlo is an event-based method that is formulated such that a reaction is randomly sampled [...] The frequency therefore should be input as an additional absorption (if positive) or production (if negative) reaction term. The absolute value of the frequency shall be added to the total cross section used to compute the distance to collision. During the collision, the absolute value of the frequency is used as a distributed absorber cross section. If the distributed absorber reaction is sampled, the neutron is either killed (if the frequency is positive) or a secondary neutron is born with the same weight, energy, and trajectory as the incident neutron (if the frequency is negative)."

C.4 Delayed Neutron Precursor Integration

For delayed neutron modeling, an alternative to Dynamic Monte Carlo has been introduced to save a significant amount of computation and memory. "The precursors will be propagated forward in time by directly integrating the space and energy integrated delayed neutron precursor balance equations assuming the fission source varies linearly over a time step. [...] Note that the decay constants, shape function, and delayed-nu-fission cross sections are at time steps t^a and t^{a-1} while these quantities are only computed during outer shape function solves at times t^s and t^{s-1}. These terms are approximated in between MC function solves using a linear interpolation."

Again, Section 5.1.2 dives into the problems associated with such a linear interpolation and suggests several possible alternatives.

C.5 Computing Coarse Mesh Diffusion Coefficients

A special point should be made about diffusion coefficients as there exist many approaches for computing them in Monte Carlo. The accuracy of diffusion coefficients will have the largest impact on cell-wise powers that are produced in the low-order solve according to Herman [31] who "evaluated the effect of different approximations on computing the diffusion coefficient for CMFD acceleration during the inactive cycles of MC. The most effective approach from Herman's work is a two-step procedure whereby the transport cross section is first computed on a fine energy group structure using the in-scatter approximation. The diffusion coefficient is then collapsed from the fine to coarse group structure to come up with coarse group diffusion coefficients for the diffusion solve." In this work, a 40 group CASMO structure is used for fine groups, and the 2 group CASMO structure is used for coarse groups.

These detailed descriptions of the algorithm bring to light a number of opportunities for improvement. Among these is the diffusion coefficient sampling in Monte Carlo which remains an area of active research, as well as the best methods for cross section interpolation over inner time steps which is discussed at length in this thesis.

Bibliography

[1] Kord Smith. An analytical nodal method for solving the two group, multi-dimensional, static and transient neutron diffusion equation. Master's thesis, Massachusetts Institute of Technology, Nuclear Science and Engineering, June 1979.

[2] B. Martin. "Challenges and Prospects for Whole-Core Monte Carlo Analysis". *Nuclear Engineering and Technology*, 44(2):151–160, March 2012.

[3] D. Kelly, T. Sutton, T. Trumbull, and P. Dobreff. "MC21 Monte Carlo Analysis of the Hoogenboom-Martin Full-Core PWR Benchmark Problem". In *Proc. PHYSOR (PHYSOR2010)*, Pittsburgh, PA, May 9–14, 2010. American Nuclear Society.

[4] T. Bahadir and S. Lindhal. "Studsvik's Next Generation Nodal Code Simulate5". In *Advances in Nuclear Fuel Management IV*, Hilton Head Island, SC, April 12 – 15, 2009. American Nuclear Society.

[5] D. Knott and A. Yamamoto. *Handbook of Nuclear Engineering*, chapter 9. Springer, 2010.

[6] D. Knott, B. Forssen, and M. Edenius. "CASMO-4 - a Fuel Assemby Burnup Program, Methodology Feedback". Technical Report STUDSVIKSOA-95/2, Studsvik of America, 1995.

[7] R. Sanchez, J. Mondot, Z. Stancovski, A. Cosic, and I. Zmijarevic. "APOLLO II: a User-oriented, Portable, Modular Code for Multi-group Transport Assembly Calculations". *Nuclear Science and Engineering*, 100:352, 1988.

[8] J. Casal, R. Stamm'ler, E. Villarion, and A. Ferri. "HELIOS: Geometric Capabilities of a New Fuel Assembly Program". In *International Topical Meeting on Advances in Mathematics, Computations, and Reactor Physics*, Pittsburgh, PA, April 28 – May 2, 1991.

[9] MPACT Team. *MPACT Theory Manual*. University of Michigan, Ann Arbor MI, 1.0 edition, October 2013.

[10] B. Kelley and E. Larsen. "A Consistent 2D/1D Approximation to the 3D Neutron Transport Equation". *Nuclear Engineering and Design*, 295:568, 2015.

[11] Z. Liu, L. Liang, C. Liangzhi, and W. Hongchun. "A 2D/1D Coupling Method for Transport Calculation". In *International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*, Jeju, Korea, April 16–20, 2017.

[12] X. Tang, Q. Li, X. Tu, W. Wu, and W. K. "Efficient Procedure for Radial MOC and Axial SN Coupled 3D Neutron Transport Calculation". In *International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*, Jeju, Korea, April 16–20, 2017.

[13] W. Wu, Q. Li, and K. Wang. "Verification of the 2D/1D Coupling 3D Transport Code TIGER with C5G7 Benchmarks". In *International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*, Jeju, Korea, April 16–20, 2017.

[14] John Tramm. *Development of The Random Ray Method of Neutral Particle Transport for High-Fidelity Nuclear Reactor Simulation*. PhD dissertation, Massachusetts Institute of Technology, Nuclear Science and Engineering, February 2018.

[15] Geoffrey Gunow. *Full Core 3D Neutron Transport Simulation Using the Method of Characteristics with Linear Sources*. PhD dissertation, Massachusetts Institute of Technology, Nuclear Science and Engineering, June 2018.

[16] Derek Gaston. *Parallel, Asynchronous Ray-Tracing for Scalable, 3D, Full-Core Method of Characteristics Neutron Transport on Unstructured Mesh*. PhD dissertation, Massachusetts Institute of Technology, Nuclear Science and Engineering, February 2020.

[17] K. Smith. "Nodal Method Storage Reduction by Nonlinear Iteration". In *Transactions of the American Nuclear Society*. Vol 44, American Nuclear Society, 1983.

[18] T. Newton, G. Hosking, L. Hutton, D. Powney, B. Turland, and T. Shuttleworth. "Developments Within WIMS 10". In *International Conference on the Physics of Reactors*, Interlaken, Switzerland, September 14–19, 2008.

[19] Lulu Li. A low order acceleration scheme for solving the neutron transport equation. Master's thesis, Massachusetts Institute of Technology, Nuclear Science and Engineering, September 2013.

[20] B. Kochunas, T. Downar, and Z. Liu. "Application of the SDD-CMFD Accelerations Method to Parallel 3-D MOC Transport". In *PHYSOR (PHYSOR2014)*, Kyoto, Japan, September 28 – October 3, 20144.

[21] Sterling Harper. *Tally Derivative Based Surrogate Models for Faster Monte Carlo Multiphysics*. PhD dissertation, Massachusetts Institute of Technology, Nuclear Science and Engineering, September 2020.

[22] J. Conlin, W. Ji, J. Lee, and W. Martin. “Pseudo Material Construct for Coupled Neutronic-Thermal-Hydraulic Analysis of VHTGR". *Transactions of the American Nuclear Society*, 92:225, 2005.

[23] P. Ducru, C. Josey, K. Dibert, V. Sobes, B. Forget, and K. Smith. “Kernel Reconstruction Methods for Doppler Broadening - Temperature Interpolation by Linear combination of Reference Cross Sections at Optimally Chosen Temperatures". *Journal of Computational Physics*, 335:535, 2017.

[24] W. Martin. "Implementation of On-the-Fly Doppler Broadening in MCNP for Multiphysics Simulation of Nuclear Reactors". Technical Report Project 10-897, DOE NEUP, November 2012.

[25] T. Viitanen and J. Leppänen. “Explicit Treatment of Thermal Motion in Continuous-Energy Monte Carlo Tracking Routines". *Nuclear Science and Engineering*, 171(2):165, 2012.

[26] T. Viitanen and J. Leppänen. “Target Motion Sampling Temperature Treatment Technique with Elevated Basis Cross-Section Temperatures". *Nuclear Science and Engineering*, 177(1):77, 2014.

[27] B. Forget, S. Xu, and K. Smith. “Direct Doppler Broadening in Monte Carlo Simulations Using the Multipole Representation". *Annals of Nuclear Energy*, 64:78, 2014.

[28] C. Josey, P. Ducru, B. Forget, and K. Smith. “Windowed Multipole for Cross Section Doppler Broadening". *Journal of Computational Physics*, 307:715–727, August 2016.

[29] M. Lee, H. Joo, D. Lee, and K. Smith. "Multigroup Monte Carlo Reactor Calculation with Coarse Mesh Finite Difference Formulation for Real Variance Reduction". In *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo (SNA+MC2010)*, Tokyo, Japan, October 17–21, 2010.

[30] K. Keady and E. Larsen. “Stability of Monte Carlo k-eigenvalue Simulations with CMFD Feedback". *Journal of Computational Physics*, 321:947–964, 2016.

[31] Bryan Herman. *Monte Carlo and Thermal Hydraulic Coupling Using Low-Order Nonlinear Diffusion Acceleration*. PhD dissertation, Massachusetts Institute of Technology, Nuclear Science and Engineering, September 2014.

[32] M. Kreher, B. Forget, and K. Smith. "Single-Batch Monte Carlo Multiphysics Coupling". In *Proc. International Conference on Mathematics and Computation (M&C2019)*, Portland, OR, August 25–29, 2019. American Nuclear Society.

[33] D. Gill, D. Griesheimer, and D. Aumiller. “Numerical Methods in Coupled Monte Carlo and Thermal-Hydraulic Calculations". *Nuclear Science and Engineering*, 185:194–205, February 2017.

[34] M. Lemaire, H. Lee, N. Tak, H. Lee, and D. Lee. "Monte Carlo/Thermal-Fluids Coupled Calculations for MHTGR-350MW Benchmark". In *Proc. International Conference on Mathematics and Computation (M&C2017)*, Jeju, Korea, April 16–20, 2017. American Nuclear Society.

[35] J. Dufek and W. Gudowski. "Stochastic Approximation for Monte Carlo Calculation of Steady-State Conditions in Thermal Reactors". *Nuclear Science and Engineering*, 152:274–283, 2006.

[36] N. Todreas and M. Kazimi. *Nuclear Systems Volume I Second Edition*, chapter 3. CRC Press USA, 2012.

[37] N. Todreas and M. Kazimi. *Nuclear Systems Volume I Second Edition*, chapter 13. CRC Press USA, 2012.

[38] N. Todreas and M. Kazimi. *Nuclear Systems Volume I Second Edition*, chapter 8. CRC Press USA, 2012.

[39] D. Hagrman and G. Reymann. *MATPRO A Handbook of Material Properties for Use in the Analysis of Light Water Reactor Fuel Rod Behavior*, chapter 2. Idaho National Laboratory, 1979.

[40] P. Van Uffelen, R. Konings, C. Vitanza, and J. Tulenko. *Handbook of Nuclear Engineering*, chapter 13. Springer, 2010.

[41] G. Grandi, K. Smith, Z. Xu, and J. Rhodes. "Effect of CASMO-5 Cross-Section Data and Doppler Temperature Definitions on LWR Reactivity Initiated Accidents". In *Proc. PHYSOR (PHYSOR2010)*, Pittsburgh, PA, May 9–14, 2010. American Nuclear Society.

[42] P. Romano, N. Horelik, B. Herman, A. Nelson, B. Forget, and K. Smith. "OpenMC: A State-of-the-Art Monte Carlo Code for Research and Development". *Annals of Nuclear Energy*, 82:90–97, August 2015.

[43] N. Norelik, B. Herman, B. Forget, and K. Smith. "Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS), v1.0.1.". In *Proc. International Conference on Mathematics and Computation (M&C2013)*, Sun Valley, ID, May 5–9, 2013. American Nuclear Society.

[44] N. Todreas and M. Kazimi. *Nuclear Systems Volume I Second Edition, Appendix K*. CRC Press USA, 2012.

[45] J. Yeh, T. Sung, and J. Shiue. "Developing a Tool for Simulating the Behavior of Axial Flux Difference during Power Reduction for Maanshan Pressurized Water Reactor Units". *Nuclear Technology*, 156(2):125–132, 2006.

[46] V. Rondinella and T. Wiss. "The High Burn-up Structure in Nuclear Fuel". *Materials Today*, 13(12):24–32, 2010.

[47] A. Lund, P. Romano, and A. Siegel. "Accelerating Source Convergence in Monte Carlo Calculations Using a Particle Ramp-Up Technique". In *Proc. International Conference on Mathematics and Computation (M&C2017)*, Jeju, Korea, April 16–20, 2017. American Nuclear Society.

[48] R. Gast and N. Candelore. "Monte Carlo Eigenfunction Strategies and Uncertainties". In *Proc. of NEACRP Meeting of a Monte Carlo Study Group, ANL-75-2*, Argonne, Illinois, July 1–3, 1974.

[49] X-5 Monte Carlo Team. "MCNP - a General Monte Carlo N-Particle Transport Code, Version 5, Volume 1: Overview and Theory". Technical Report LA-UR-03-1987, Los Alamos National Laboratory, 2008.

[50] J. Dufek and K. Tuttelberg. "Monte Carlo Criticality Calculations Accelerated by a Growing Neutron Population". *Annals of Nuclear Energy*, 94:16–21, 2016.

[51] Gerardo Grandi. *SIMULATE3-K Models & Methodology*. Studsvik Scandpower Inc, Newton MA, rev 7 edition, July 2011.

[52] K. Derstine. "DIF3D: a Code to Solve One-, Two-, and Three- Dimensional Diffusion Theory Problems". Technical Report ANL-82-64, Argonne National Laboratory, 1984.

[53] Y. Liu. "ANC: a Westinghouse Advanced Nodal Computer Code". Technical Report WCAP-10966-A-ADI, Westinghouse Electric Company, 1986.

[54] Bernard R. Bandini. *A Three-Dimensional Transient Neutronics Routine for the TRAC-PF1 Reactor Thermal Hydraulic Computer Code*. PhD dissertation, Pennsylvania State University, Department of Nuclear Engineering, May 1990.

[55] Y. Xu, T. Downar, D. Lee, and T. Kozlowski. *PARCS v2.6: U.S. NRC Core Neutronics Simulator*. Purdue University School of Nuclear Science and Engineering, West Lafayette, IN, November 2004.

[56] C. J. Permann, D. R. Gaston, D. Andrs, R. W. Carlsen, F. Kong, A. D. Lindsay, J. M. Miller, J. W. Peterson, A. E. Slaughter, R. H. Stogner, and R. C. Martineau. "MOOSE: Enabling Massively Parallel Multiphysics Simulation". *SoftwareX*, 11:100430, March 2020.

[57] A. Zhu, Y. Xu, A. Graham, M. Young, and T. Downar. "Transient Methods for Pin-Resolved Whole Core Transport using the 2D-1D Methodology in MPACT". In *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications and the Monte Carlo Method (SNA+MC)*, La Grange Park, IL, April 19–23, 2015. American Nuclear Society.

[58] A. Hoffman. *A Time-Dependent Method of Characteristics Formulation with Time Derivative Propagation*. PhD dissertation, University of Michigan, Nuclear Engineering and Radiological Sciences, 2013.

[59] J. Taylor. *The Development of a Three-Dimensional Nuclear Reactor Kinetics Methodology Based on the Method of Characteristics*. PhD dissertation, Pennsylvania State University, Nuclear Engineering, December 2007.

[60] T. Endo, Y. Ban, and A. Yamamoto. "A Unified Approach for Numerical Calculation of Space-Dependent Kinetic Equation". *Journal of Nuclear Science and Technology*, 49(5):496–515, 2012.

[61] J. Cho, K. Kim, C. Lee, H. Joo, W. Yang, T. Taiwo, and J. Thomas. "Transient Capability for a MOC-Based Whole Core Transport Code DeCart". In *Transactions of the American Nuclear Society*. Vol 92, American Nuclear Society, 2005.

[62] Samuel Shaner. Transient method of characteristics via the adiabatic, theta, and multi-grid amplitude function methods. Master's thesis, Massachusetts Institute of Technology, Nuclear Science and Engineering, June 2014.

[63] K. Tsujita, T. Endo, and A. Yamamoto. "Efficient Reduced Order Model Based on the Proper Orthogonal Decomposition for Time-Dependent MOC Calculations". *Journal of Nuclear Science and Technology*, 2022.

[64] Bart Sjenitzer. *The Dynamic Monte Carlo Method for Transient Analysis of Nuclear Reactors*. PhD dissertation, Delft University of Technology, Physics Engineering, July 2013.

[65] V. Valtavirta, M. Hessan, and J. Lappänen. "Delayed Neutron Emission Model for Time Dependent Simulations with the SERPENT 2 Monte Carlo Code — First Results". In *Proc. PHYSOR (PHYSOR2016)*, Sun Valley, ID, May 1–5, 2016. American Nuclear Society.

[66] D. Ferraro, M. Faucher, D. Mancusi, A. Zoia, V. Valtavirta, J. Lappänen, and V. Sanchez-Espinoza. "SERPENT and TRIPOLI-4 Transient Calculations Comparisons for Several Reactivity Insertion Scenarios in a 3D Minicore Benchmark". In *Proc. International Conference on Mathematics and Computation (M&C2019)*, Portland, OR, August 25–29, 2019. American Nuclear Society.

[67] M. Faucher, D. Mancusi, and A. Zoia. "New Kinetic Simulation Capabilities for TRIPOLI-4: Methods and Applications". *Annals of Nuclear Energy*, 120:74–88, May 2018.

[68] J. Hendricks and T. Booth. *MCNP Variance Reduction Overview*, volume 240 of *Monte-Carlo Methods and Applications in Neutronics, Photonics and Statistical Physics, Lecture Notes in Physics*. Springer, Berlin, Heidelberg, 10 January 1973.

[69] T. Booth. "A Weight (Charge) Conserving Importance-Weighted Comb for Monte Carlo". In *Proc. Radiation Protection and Shielding Division Topical Meeting*, Flamouth, MA, April, 1996. American Nuclear Society.

[70] M. A. Kreher and T. Trahan. "Accelerating Time-Dependent Monte Carlo Algorithms with Explicit Delayed Neutron Precursors". In *Proc. International Conference on Mathematics and Computation (M&C2021)*, virtual meeting, October 3–7, 2021. American Nuclear Society.

[71] D. Cullen. "Tart 2002: a Coupled Neutron-Photon, 3D, Combinatorial Geometry, Time Dependent Monte Carlo Transport Code". Technical Report UCRL-ID-126455, Lawrence Livermore National Laboratory, November 2002.

[72] M. Faucher, D. Mancusi, and A. Zoia. "Multi-Physics Transient Simulations with TRIPOLI-4". In *Proc. PHYSOR (PHYSOR2020)*, Virtual meeting, 2020. American Nuclear Society.

[73] K. Tsujita, T. Endo, and A. Yamamoto. "Application of the Multigrid Amplitude Function Method for Time-Dependent MOC Based on the Linear Source Approximation". *Journal of Nuclear Science and Technology*, 57(6):646–662, 2020.

[74] M. A. Kreher, K. Smith, and B. Forget. "Direct Comparison of High-Order/Low-Order Transient Methods on the 2D-LRA Benchmark". *Nuclear Science and Engineering*, 196(4):409–432, December 2021.

[75] T. Trahan. "A Quasi-Static Monte Carlo Algorithm for the Simulation of Sub-Prompt Critical Transients". *Annals of Nuclear Energy*, 127:257–267, May 2019.

[76] Samuel Shaner. *Development of High Fidelity Methods for 3D Monte Carlo Analysis of Nuclear Reactors*. PhD dissertation, Massachusetts Institute of Technology, Nuclear Science and Engineering, February 2018.

[77] M. Kreher, S. Shaner, B. Forget, and K. Smith. "Frequency Transform Method for Transient Analysis of Nuclear Reactors". *Nuclear Science and Engineering*, 174, 2022.

[78] T. Adams, S. Nolen, J. Sweezy, A. Zukaitis, J. Campbell, T. Goorley, S. Greene, and R. Aulwes. "Monte Carlo Application Toolkit (MCTAK)". *Annals of Nuclear Energy*, 82:41, August 2015.

[79] L. Jain, R. Karthikeyan, and U. Kannan. "Multi-Grid Acceleration Scheme for Neutron Transport Calculations Using Optimally Diffusive CMFD Method". In *Symposium on Advanced Sensors and Modeling Techniques for Nuclear Reactor Safety*, 2018.

[80] B. Chang, T. Manteuffel, S. McCormick, J. Ruge, and B. Sheehan. "Spatial Multigrid for Isotropic Neutron Transport". *SIAM Journal for Scientific Computing*, 29(5):1900, 2007.

[81] J. Yoon and H. Joo. “Two-Level Coarse Mesh Finite Difference Formulation with Multigroup Source Expansion Nodal Kernels". *Journal of Nuclear Science and Technology*, 45(7):668, 2008.

[82] Benchmark Problem Committee. “Argonne Code Center: Benchmark Problem Book". Technical Report ANL-7416 and Suppl. 2, Argonne National Laboratory, 1977.

[83] M. Abdo, R. Elzohery, and J. Robert. "Analysis of the LRA Reactor Benchmark Using Dynamic Mode Decomposition". In *Transactions of the American Nuclear Society*. Vol 119, American Nuclear Society, 2018.

[84] J. Banfield. *Semi-Implicit Direct Kinetics Methodology for Deterministic, Time-Dependent, Three-Dimensional, and Fine-Energy Neutron Transport Solutions*. PhD dissertation, University of Tennessee Knoxville, Nuclear Engineering, December 2013.

[85] Q. Wu, X. Peng, X. Tang, Y. Yu, Q. Li, and K. Wang. “Whole-Core Forward-Adjoint Neutron Transport Solutions with Coupled 2-D MOC and 1-D SN and Kinetics Parameter Calculation". *Progress in Nuclear Energy*, 108:310, 2018.

[86] J. Yasinsky and A. Henry. “Some Numerical Experiments Concerning Space-Time Reactor Kinetics Behavior". *Nuclear Science and Engineering*, 22(2):171, 1965.

[87] Z. Prince and J. Ragusa. “Multiphysics Reactor-Core Simulations Using the Improved Quasi-Static Method". *Annals of Nuclear Energy*, 125:186, 2019.

[88] A. Al-Mohy and N. Higham. “A New Scaling and Squaring Algorithm for the Matrix Exponential". *SIAM Journal on Matrix Analysis and Applications*, 31(3):970, 2009.

[89] D. Ferguson and K. Hansen. “Solution of the Space-Dependent Reactor Kinetics Equations in Three Dimensions". Technical Report MIT-2903-4, AEC Research and Development, 1971.

[90] Mario Ortega. *A Rayleigh Quotient Fixed Point Method for Criticality Eigenvalue Problems in Neutron Transport*. PhD dissertation, University of California Berkeley, Nuclear Engineering, 2019.

[91] T. Hill. “Efficient Methods for Time Absorption (α) Eigenvalue Calculations". Technical Report LA-9602-MS, Los Alamos National Laboratory, 1983.

[92] C. Josey and F. Brown. "A New Monte Carlo Alpha-Eigenvalue Estimator with Delayed Neutrons". In *Transactions of the American Nuclear Society*. Vol 118, American Nuclear Society, 2018.

[93] K. Ott and D. Meneley. “Accuracy of the Quasistatic Treatment of Spatial Reactor Kinetics". *Nuclear Science and Engineering*, 36(3):402, 1969.

[94] Zachary Prince. Improved quasi-static methods for time-dependent neutron diffusion and implementation in Rattlesnake. Master's thesis, Texas A&M University, Nuclear Engineering, May 2017.

[95] K. Ott. "Quasistatic Treatment of Spatial Phenomena in Reactor Dynamics". *Nuclear Science and Engineering*, 26(4):563, 1966.

[96] Y. Chao and A. Attard. "A Resolution of the Stiffness Problem of Reactor Kinetics". *Nuclear Science and Enuillaumineering*, 90(1):40–46, 1985.

[97] Y. Chao and P. Huang. "Theory and Performance of the Fast-Running Multidimensional Pressurized Water Reactor Kinetics Code, SPNOVA-K". *Nuclear Science and Engineering*, 103(3):415–419, 1989.

[98] J. Hou, K. Ivanov, V. Voyarinov, and P. Fomichenko. "OECD/NEA Benchmarks for Time-Dependent Neutron Transport Calculations without Spatial Homogenization". *Nuclear Engineering and Design*, 317:177–189, 2017.

[99] F. Chavagnat, R. Nop, N. Dorville, B. Phillips, and M. Bucci. "Single-Phase Heat Transfer Regimes in Forced Flow Conditions Under Exponential Heat Inputs". *International Journal of Heat and Mass Transfer*, 2021.

[100] Sterling Harper. Calculating reaction rate derivatives in monte carlo neutron transport. Master's thesis, Massachusetts Institute of Technology, Nuclear Science and Engineering, June 2016.

[101] Z. Liu, K. Smith, B. Forget, and J. Ortensi. "Cumulative Migration Method for Computing Rigorous Diffusion Coefficients and Transport Cross Sections from Monte Carlo". *Annals of Nuclear Energy*, 112:507–516, 2018.

[102] B. Herman, B. Forget, and K. Smith. "Improved Diffusion Coefficients from Monte Carlo Codes". In *International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*, Sun Valley, ID, May 5–9, 2013. American Nuclear Society.

[103] T. Sutton and B. Aviles. "Diffusion Theory Methods for Spatial Kinetics Calculations". *Progress in Nuclear Energy*, 30(2):119, 1996.