

Multi-Level Methods for Monte Carlo Reactor Kinetics

by

Evan S. Gonzalez

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in the University of Michigan
2023

Doctoral Committee:

Associate Professor Brian C. Kiedrowski, Chair
Dr. Gregory G. Davidson
Professor Krzysztof J. Fidkowski
Assistant Professor Brendan Kochunas

Evan S. Gonzalez
evangonzalez.mail@gmail.com
ORCID iD: 0000-0002-2626-0981

© Evan S. Gonzalez 2023

Dedication

To my parents, Angela and David, both of whom had to drop out of high school when I was born.

Acknowledgments

This work was funded by the Spent Fuel and Waste Disposition program from the Department of Energy – Office of Nuclear Energy (DOE-NE). Thank you to DOE-NE for their financial support.

Thank you to my advisor, Prof. Brian Kiedrowski. I have heard plenty of horror stories from other grad students about their advisors and am proud to say that I have no such stories of my own to share. Your knowledge and compassion are second to none.

Thank you to Greg Davidson at Oak Ridge National Laboratory for giving me a chance and advocating for my technical capabilities.

Thank you to all of the friends I made here at University of Michigan. There are too many to list here, but some that stick out are Rob Fonti, Simon Essig Aberg, Aaron Tumulak, and Kyle Beyer. Your kindness and humor did not go unnoticed.

Thank you to my parents and step-parents for their confidence in me. Thank you to my siblings, Ezra and Eli, for the long phone calls and sharing what's going in your lives with me. Thank you to little Victor for showing me all of your very cool toy trucks.

Lastly, and most importantly, thank you a thousand times over to my fiancée, Zereth. You were the only one who saw the late nights and talked me through them. From nearly the beginning to the bittersweet end.

To everyone I've interacted with over the last five and a half years, whether I mentioned you or not – Thank you for helping me. Thank you for believing in me.

Table of Contents

Dedication	ii
Acknowledgments	iii
List of Figures	vii
List of Tables	x
List of Appendices	xi
List of Abbreviations	xii
List of Algorithms	xiv
Abstract	xv
Chapter 1: Background & Introduction	1
1.1 Motivation	1
1.2 Literature Review	3
1.2.1 The Evolution of Reactor Kinetics	3
1.2.2 Parallelism in Neutron Transport	5
1.2.3 Parallel-in-Time Applications in Other Fields	6
1.2.4 Correlated Sampling for k -Eigenvalue Calculations	7
1.3 Unique Features of This Work	8
1.4 Dissertation Outline	9
Chapter 2: Transport Fundamentals	11
2.1 Boltzmann Equation	11
2.2 Neutron Transport Equation	13
2.2.1 Steady-State Neutron Transport Equation	20
2.2.2 k -Eigenvalue Problems	21
2.2.3 α -Eigenvalue Problems	22
2.3 Adjoint Neutron Transport Equation	23
2.4 Exact Point Kinetics Equations	26
2.4.1 Derivation for an Initially Critical Reactor	27
2.4.2 Finite Differencing for the Exact Point Kinetics Equations	31
2.4.2.1 Integrating the Precursor Equations	32
2.4.2.2 Prompt Reactivity Feedback	34
2.5 Doppler Broadening	37
2.6 Approximations to the Neutron Transport Equation	39
2.6.1 Multigroup Approximation	40
2.6.2 Angular Approximations	44
2.6.2.1 P_1 and Diffusion Approximations	45
2.6.2.2 SP_3 Approximation	49

2.7	Transient Fixed Source Problem	52
2.7.1	Diffusion Formulation	54
2.8	Coarse Mesh Finite Difference Equations	56
2.9	Deterministic Methods	61
2.10	Monte Carlo Method	62
2.10.1	Tallies	65
2.10.1.1	Generating Multigroup Cross Sections	67
2.10.2	Criticality Calculations	68
2.10.3	Non-analog Monte Carlo	69
2.11	Exnihilo Code Suite	70
2.12	Summary	71
Chapter 3:	Transient Multi-Level Method for Monte Carlo	72
3.1	Quasi-Static Methods	73
3.1.1	Predictor Corrector Quasi-Static Method	74
3.2	Transient Multi-Level Method	76
3.2.1	Transient Monte Carlo Solver	77
3.2.1.1	Shift Implementation	79
3.2.2	Monte Carlo Transport and CMFD Coupling	83
3.2.2.1	Macroscopic Cross Sections	84
3.2.3	CMFD and EPKE Coupling	86
3.2.4	Implementation Overview	87
3.3	C5G7 TD-3 Benchmark Results	89
3.3.1	PCQM Results	92
3.4	TML Results	93
3.5	Prompt Feedback Model	97
3.6	Dual-Purpose Canister Model	99
3.7	Issues with the CMFD Solver	103
3.8	Summary	104
Chapter 4:	Parallel-in-Time Acceleration for the TML	106
4.1	Parareal Algorithm	106
4.1.1	Serial Implementation for Point Kinetics	111
4.1.1.1	Slow Asymmetric Reactivity Ramp	112
4.1.1.2	Fast Control Rod Ejection	115
4.2	Serial Implementation for TML Subsolders	118
4.2.1	Introducing an Additional 1-Group CMFD Level	120
4.2.2	Coupling Between Monte Carlo Levels	125
4.3	Correlated Sampling	127
4.3.1	Derivation of the Direct and Indirect Perturbation Effects	128
4.3.2	Perturbed Fission Source Weighting	134
4.3.3	Extension of Perturbed Fission Source Weighting to Other Tally Ratios	135
4.3.4	Godiva Perturbations	139
4.3.5	Potential Use in a Multi-Level Scheme	142
4.3.5.1	Direct Effect	144
4.3.5.2	Indirect Effect on the Perturbed Random Walk	144
4.3.5.3	Indirect Effect on the Perturbed Fission Source	144
4.4	Summary	147
Chapter 5:	Conclusions and Future Work	148
5.1	Monte Carlo Transient Multi-Level Method	148
5.1.1	Future Work	149
5.2	Parallel-in-Time Acceleration of the TML Subsolders	150
5.2.1	Correlated Sampling as a potential Coarse-Estimate MC Solver	151

5.2.2	Future Work	151
References		153
Appendices		162

List of Figures

Figure 2.1	Particle spatial, angular, and energy phase space.	13
Figure 2.2	Depiction of the angular flux as an angular fluence rate density.	14
Figure 2.3	Doppler broadening effect on resonance shape as it appears to a neutron where $T_0 < T_1 < T_2$ and E_r is the center energy of the resonance.	39
Figure 2.4	The rapidly varying energy dependence of the fission, capture and elastic scattering interactions from U^{238}	43
Figure 2.5	Scalar flux of a reactor quarter-core discretized on both fine (left) and coarse (right) spatial meshes, where the coarse mesh might represent the fidelity of a CMFD discretization. Scalar flux units are $n \text{ cm}^{-2} \text{ s}^{-1}$	57
Figure 2.6	Example of a 2D CMFD spatial discretization.	58
Figure 2.7	Level-symmetric S_8 ordinates (left) shown for a single octant of the unit sphere and the order of sweeping for 2D cell-centered spatial mesh (right) for $\mu_m > 0, \eta_m > 0$ [43]. The nodes on the spatial discretization diagram are differentiated as $\blacksquare \rightarrow$ computed flux at cell interface, $\bullet \rightarrow$ known boundary flux, $\blacktriangle \rightarrow$ computed flux at cell center.	62
Figure 2.8	Distances to plane (left), sphere (center), and z-aligned cylinder (right). The characteristic equation of each surface is also shown below the surface.	64
Figure 3.1	Quasi-static time discretization scheme.	74
Figure 3.2	PCQM time discretization scheme.	75
Figure 3.3	Comparison of the IQM and PCQM algorithms.	76
Figure 3.4	TML time stepping scheme [13].	77
Figure 3.5	Simplified neutron history branching in transient fission source iteration scheme.	81
Figure 3.6	Transfer of source terms between fission source iterations and time steps.	82
Figure 3.7	Flow of information for coupling between the MC, CMFD, and EPKE levels.	87
Figure 3.8	South-east quadrant of the C5G7 reactor core configuration homogenized at the pin cell level.	90
Figure 3.9	Time-dependent changes in the average moderator density specified for the C5G7-TD3 exercises.	91
Figure 3.10	Dynamic reactivity and relative fission rate of the PCQM implementation.	93
Figure 3.11	Dynamic reactivity and relative fission rate of the PCQM implementation with stochastic noise removed for $t > 2$ s.	93
Figure 3.12	Dynamic reactivity and relative fission rate of the TML implementation with stochastic noise removed for $t > 2$ s.	94
Figure 3.13	Uncertainties from stochastic noise associated with dynamic reactivity and relative fission rate for TD3-4.	96
Figure 3.14	Time step convergence analysis for TD3-4 exercise.	97
Figure 3.15	C5G7 step reactivity insertion with simplified Doppler feedback.	98
Figure 3.16	Average weight of time source particles (before population combing).	99
Figure 3.17	Axial (left) and radial (right) cross-sectional views of the Dual-Purpose Canister (DPC) geometry modeled. This DPC configuration has 32 unique fuel assemblies, each with 18 axial segments.	100
Figure 3.18	Dynamic reactivity (left) and fractional core fission rate (right) for step reactivity insertion with DPC canister.	101

Figure 3.19	Radial fission profile (top) and percent differences in the fission profile from the previous MC time step t^{n-1} (bottom) integrated over axial length of core at select points in time throughout the DPC simulation. The normalizations are computed such that the radial fission distribution integrated over all of phase space at each individual time step is unity.	102
Figure 3.20	Axial fission profile (top) and percent differences from $t = 0$ s (bottom) integrated over radial length of core at select points in time throughout the DPC simulation. The normalizations are computed such that the axial fission distribution integrated over all of phase space at each individual time step is unity.	103
Figure 4.1	Parareal time discretization scheme.	108
Figure 4.2	Example of parareal algorithm updates to the solution $\mathbf{u}(t)$ from both the coarse ($\mathcal{G}$) and fine $\mathcal{F}$ propagators. The analytical solution is shown as the thin black line.	110
Figure 4.3	Parareal reactivity profile for asymmetric reactivity ramp with one delayed neutron precursor group.	113
Figure 4.4	Parareal power profile for asymmetric reactivity ramp with one delayed neutron precursor group.	113
Figure 4.5	Parareal delayed neutron precursor concentration profile for asymmetric reactivity ramp.	114
Figure 4.6	Convergence (left) and relative computational time (right) between serial parareal implementation and serial backward Euler implementation for reactivity ramp problem at various ratios of $\Delta t_{\text{coarse}}/\Delta t_{\text{fine}}$	115
Figure 4.7	Parareal reactivity profile for control rod ejection simulated with PARCS.	116
Figure 4.8	Parareal power profile for control rod ejection problem.	116
Figure 4.9	Parareal delayed neutron precursor concentration profiles for control rod ejection problem. The top-right and bottom plots show the difference between the precursor concentration profiles at k versus $k = 1$	117
Figure 4.10	Convergence (left) and relative computational time (right) between serial parareal implementation and serial backward Euler implementation for control rod ejection problem at various ratios of $\Delta t_{\text{coarse}}/\Delta t_{\text{fine}}$	117
Figure 4.11	Flow of information between unmodified TML levels. The circled numbers indicated the order in which the various solvers are evaluated throughout the transient.	119
Figure 4.12	Flow of information between parareal coupling of CMFD and EPKE levels. The circled numbers indicated the order in which the various solvers are evaluated throughout the transient.	119
Figure 4.13	Parareal fission rate with coupling between CMFD and EPKE levels.	120
Figure 4.14	Parareal flow of information for coupling between CMFD-7G and CMFD-1G levels with $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 5$	122
Figure 4.15	Fractional core fission rate for C5G7-TD3-4 problem with serial parareal between CMFD-7G and CMFD-1G levels and $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 5$	123
Figure 4.16	Fractional core fission rate for C5G7-TD3-4 problem with serial parareal coupling between CMFD-7G and CMFD-1G levels and $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 10$	123
Figure 4.17	Fractional core fission rate for C5G7 positive reactivity ramp with serial parareal coupling between CMFD-7G and CMFD-1G levels and $\Delta t_{\text{CMFD-7G}}/\Delta t_{\text{CMFD-1G}} = 10$	124
Figure 4.18	Projected speedup of parallel parareal coupling between CMFD-7G and CMFD-1G levels for various values of $\Delta t_{\text{CMFD-7G}}/\Delta t_{\text{CMFD-1G}}$. The left plot shows speedup if the CMFD-7G solution is additionally parallelized across energy groups, while the right shows speedup if no additional parallelization is used.	125
Figure 4.19	Proposed flow of information for coupling between two MC levels.	127
Figure 4.20	Simplified neutron history branching for correlated sampling. Changes from a standard Monte Carlo history are highlighted in blue text.	133

Figure 4.21	Simplified neutron history branching for correlated sampling. Changes for the non-perturbed fission source effect from a standard Monte Carlo history are highlighted in blue text. Changes for the perturbed fission source effect from a standard Monte Carlo history are highlighted in red text.	138
Figure 4.22	Godiva uniform (left) and central (right) perturbations where the dark gray regions correspond to the perturbed material density.	139
Figure 4.23	Correlated sampling results with perturbed fission source for Godiva configuration with various density perturbations applied uniformly to the entire Godiva geometry. The top plots describe the change in k_{eff} (left) and the change in the scattering reaction rate for the entire geometry relative the scalar flux (right). The bottom plots describe the change in the outer surface current out of the Godiva geometry relative to the scattering reaction rate for the entire geometry (left) and the change in the surface current out of an internal spherical surface with a 6 cm radius relative to the scattering reaction rate for the entire geometry (right).	140
Figure 4.24	Correlated sampling results with perturbed fission source for Godiva configuration with various density perturbations to the 6 cm inner Godiva core. The top plots describe the change in k_{eff} (left) and the change in the scattering reaction rate for the entire geometry relative to the scalar flux (right). The bottom plots describe the change in the outer surface current out of the Godiva geometry relative to the scattering reaction rate for the entire geometry (left) and the change in the surface current out of the 6 cm interface between the two Godiva regions relative to the scattering reaction rate for the entire geometry (right).	141
Figure 4.25	Correlated sampling results with perturbed fission source for pincell configuration with various moderator density perturbations. The top left plot describes the change in k_{eff} , the top right plot describes the change in the scattering reaction rate in the fuel relative the scalar flux in the fuel, and the bottom plot describes the change in the surface current out of the fuel relative to the scattering reaction rate in the fuel.	142
Figure 4.26	Correlated sampling results for C5G7-TD3-4 transient.	143
Figure 4.27	Transfer of source terms and perturbed fission source weights between fission source iterations and time steps for a time-propagated correlated sampling scheme.	146
Figure A.1	Quadratic Lagrange polynomial interpolation $P(t)$ of arbitrary function $p(t)$	164
Figure B.1	Asymmetric Reactivity Ramp	167

List of Tables

Table 1.1	A literature comparison of methods for solving reactor kinetics problems. *Dabaghi et al. solves the parabolic diffusion operator with Monte Carlo (not the hyperbolic transport operator). **This work only investigates a serial implementation of the parareal algorithm.	5
Table 3.1	Energy ranges for the multigroup cross sections specified by the C5G7 benchmark.	90
Table 3.2	Norm comparisons (given in %) to MPACT solution for both PCQM and TML implementations.	95
Table 3.3	CPU time comparison (hours) between the TD3 solution methods.	95
Table 3.4	Delayed neutron precursor decay constants and fractions for UO_2 given in the C5G7-TD benchmark [95].	101
Table 3.5	Delayed neutron fission spectra for UO_2 given in the C5G7-TD benchmark [95]. . .	101
Table 4.1	Six-group delayed neutron parameters used for control rod eject transient.	115
Table 4.2	Norm comparisons (given in %) to MPACT solution for parareal coupling between CMFD and EPKE levels.	120
Table 4.3	Summary of serial parareal problems simulated.	125

List of Appendices

Appendix A:	Temporal Integration of the Precursor Equations	162
Appendix B:	Analytic Solution for Asymmetric Reactivity Ramp	166

List of Abbreviations

CAD	Computer-Aided Design
CDF	Cumulative Distribution Function
CMFD	Coarse Mesh Finite Difference
CMR	Coarse Mesh Rebalance
CPU	Central Processing Unit
DOE-NE	Department of Energy – Office of Nuclear Energy
DPC	Dual-Purpose Canister
DSA	Diffusion Synthetic Acceleration
EPKE	Exact Point Kinetics Equations
GPU	Graphics Processing Unit
GMRES	Generalized Minimal Residual
IQM	Improved Quasi-Static Method
MC	Monte Carlo
MGM	Multigroup Matrix
MOC	Method of Characteristics
MOX	Mixed Oxide
MPACT	Michigan Parallel Characteristics Transport
ODE	Ordinary Differential Equation
ORNL	Oak Ridge National Laboratory
PARCS	Purdue Advanced Reactor Core Simulator
PCQM	Predictor-Corrector Quasi-static Method
PDE	Partial Differential Equation
PDF	Probability Density Function
PWR	Pressurized Water Reactor
S_N	Discrete Ordinates
SNL	Sandia National Laboratories
SP_N	Simplified P _N

TFSP Transient Fixed Source Problem

TML Transient Multi-Level

UO2 Uranium Dioxide

List of Algorithms

Algorithm 2.1	Finite differencing solution of the EPKEs with prompt reactivity feedback.	37
Algorithm 3.1	Improved quasi-static method iteration scheme.	75
Algorithm 3.2	Predictor corrector quasi-static method iteration scheme.	75
Algorithm 3.3	PCQM coupling between MC and CMFD levels in the TML.	84
Algorithm 3.4	PCQM coupling between CMFD and EPKE levels in the TML.	86
Algorithm 4.1	Parareal algorithm with coarse time propagator $\mathcal{G}$ and fine time propagator $\mathcal{F}$. . .	111
Algorithm 4.2	Routine for computing initial perturbed source weight for every fission source iteration.	136

Abstract

This dissertation develops methods to make time-dependent Monte Carlo transport calculations for reactor analysis more tractable. For the first time, the Transient Multi-Level (TML) method is applied to a Monte Carlo transport solver to offload some of the computational burden to lower-order Coarse Mesh Finite Difference (CMFD) and Exact Point Kinetics Equations (EPKE) solvers. The method is implemented in the production-level Monte Carlo code Shift and verified with select 2D transient events from the C5G7-TD reactor benchmark. It is shown that, as compared to other quasi-static methods, the TML reduces both the stochastic noise inherent to the Monte Carlo solver, and the number of Monte Carlo evaluations needed to simulate the transient. A step reactivity insertion for a 3D dual-purpose used nuclear fuel canister was also modeled with approximate fuel temperature Doppler broadening to stress the robustness of the method for large asymmetric configurations.

Two techniques for acceleration of the Monte Carlo TML implementation are also explored: parallel-in-time integration and correlated sampling for fission source perturbations. A serial implementation of the parareal algorithm was investigated as a novel acceleration scheme between TML levels. The method converged very quickly for all cases in less than five parareal iterations with projected speedup factors (upon a parallel implementation) ranging from 1.5-8 for the evaluated problems. For correlated sampling, a fission source weighting technique for estimating multiple k-eigenvalue perturbation tallies for a single random walk was extended to general Monte Carlo tallies and shown to be a good approximation for spatially-uniform perturbations. These perturbation tallies can then be used as input for the CMFD and EPKE solvers and serve as a coarse prediction of the propagated Monte Carlo solution.

Chapter 1

Background & Introduction

This dissertation develops methods to make time-dependent Monte Carlo transport calculations for reactor analysis more tractable than with existing methods. A hybrid quasi-static technique, the Transient Multi-Level (TML), is implemented with a Monte Carlo transport solver for the first time. The TML offloads some of the computational burden of the time propagation to lower-order Coarse Mesh Finite Difference (CMFD) and Exact Point Kinetics Equations (EPKEs) solvers and is shown to reduce both the stochastic noise and the number of Monte Carlo evaluations needed to simulate reactor transient as compared to other commonplace quasi-static methods. Approximate fuel temperature Doppler broadening is implemented to assess the applicability of the algorithm for practical problems, such as the used nuclear fuel canisters described in the next section.

The parallel-in-time “parareal” algorithm is explored as a novel acceleration scheme between various levels of the TML. The parareal algorithm (implemented in serial) is shown to converge within five outer iterations for all transients evaluated with projected speedup factors ranging from 1.5-8 for the evaluated problems. A correlated sampling technique is also explored as a way to generate Monte Carlo perturbation tallies that might be used in future work as input to the TML subsolvers in a parareal coupling between two Monte Carlo levels.

1.1 Motivation

This dissertation work was funded to support transient analyses of reactivity excursion accidents in used nuclear fuel repositories. In a broader context, this work contributes towards addressing the significant outstanding societal question of what to do with used radioactive fuel after it has served its useful lifetime in the reactor core (about three to six years in today’s commercial reactors). Upon removal from the core, used nuclear fuel is still very “hot”, generating significant amounts of heat and other forms of radiation via the radioactive decay of fission products. The current practice in the United States is to store the fuel

on-site in a pool of water immediately after it has been removed from the reactor core with enough spacing between fuel elements to prevent critical fission chains from initiating. The fuel is left in these pools until a significant fraction of the short-lived fission products have decayed away such that water cooling is no longer required, which is typically at least five years. Eventually, after a reactor has been operating for a few decades, the pools fill up and the cooled fuel needs to be moved again to be either reprocessed, or stored safely and permanently in a geological repository. However, because there is no agreed-upon national strategy in the United States for where to permanently store (or reprocess) used nuclear fuel, the current practice is to instead temporarily store the fuel on-site in large dry cask containers.

With the assumption that long-term consolidated geological storage will be the agreed-upon national strategy and that a site for a deep geological repository will be selected, engineering work has progressed to ensure a smooth transition between interim dry-cask storage and long-term indefinite storage. One proposed mechanism to maximize efficiency and minimize radiation exposure in transferring the fuel is to use a single Dual-Purpose Canister (DPC) for both interim storage and transport to the permanent storage site. In this way, radiation workers do not need to risk unnecessary radiation exposure by opening the canister to transfer the fuel. Although it is not their original design intent, DPCs are now being considered for expanded use as a container for permanent storage in the geological repository. This strategy raises some important questions: (1) How long will the structural integrity of the DPCs last? (2) What are the criticality consequences if the structural integrity of the DPCs fail? An ongoing collaborative effort between Sandia National Laboratories (SNL) and Oak Ridge National Laboratory (ORNL), and sponsored by the Department of Energy – Office of Nuclear Energy (DOE-NE) has attempted to address these questions by modeling the neutronics, heat transfer, geology, and chemistry of potential DPC criticality events [1], [2].

With respect to the neutronics, two primary criticality scenarios are considered: (1) “steady-state” criticality, where the structure of the DPC degrades over the course of tens of thousands of years and water slowly floods the canister¹, (2) “transient” criticality, where the DPC structure fails suddenly, water floods the canister, and a rapid transient proceeds over the course of seconds to hours. For the latter, high-fidelity time-dependent models are needed that can provide accurate estimates of local energy deposition and radionuclide inventory throughout the transient. These high-fidelity models can then be used to benchmark more efficient lower-fidelity models. This dissertation work was funded by the DOE-NE/SNL/ORNL collaboration and is predicated on making improvements to existing neutronics solvers that enable these transient capabilities at a practical computational expense.

Beyond DPC transient criticality events, the time-dependent behavior of neutrons in multiplying systems is of general interest to the reactor physics community, and particularly for advanced reactor design, for a

¹This includes the boroflex neutron absorbers inserted into the DPC to ensure subcriticality in flooding scenarios.

variety of reasons. For one, simulating standard operating transient scenarios can help inform engineering iterations of a new reactor design or changes to an existing reactor design. Typical transient scenarios might include control rod insertion/withdrawal to regulate reactor power, the buildup of xenon poison concentrations, or the transmutation of material compositions as neutrons interact in the reactor fuel (depletion). Provided that the solution is of a high-enough fidelity, computer simulations of these types of transients would save the engineering team the time, money, and trouble of conducting full-scale physical experiments.

Additionally, modeling atypical transient scenarios, i.e., accidents, is invaluable for informing reactor design and operation guidelines from a regulatory standpoint. For example, the Nuclear Regulatory Commission (the regulatory body responsible for ensuring the safe operation of nuclear reactors in the United States) certifies reactor designs in part by reviewing an applicant’s modeling of transient accident scenarios. The consequences of some nuclear reactor transient accidents could be catastrophic. Therefore, it is important to model them with a high degree of fidelity, precision and accuracy. Often times though, high-fidelity calculations require significant computational resources, which can be prohibitively costly and time-consuming for reactor design companies. In practice, high-fidelity models are used to benchmark more efficient lower-fidelity ones that are actually used in design calculations. This work makes progress towards achieving these high-fidelity benchmarking models by enabling transient analyses in a production-level Monte Carlo code.

1.2 Literature Review

The section gives a brief introduction to each of the topics discussed throughout this dissertation. The state of the literature is described to establish how this work fits in with and differs from existing research in the field of reactor kinetics.

1.2.1 The Evolution of Reactor Kinetics

The study of the time-dependent behavior of reactor systems, colloquially called *reactor kinetics* when feedback is excluded and *reactor dynamics* when feedback is included, has existed as a field since the 1940s. Early on, because of computational hardware constraints, reactor kinetics calculations were carried out by integrating out the spatial, energy and angular dependence of the underlying equations that govern neutron transport to a single “point” reactor model. In the 1950s, Henry and a series of colleagues observed that loosely-coupled reactor physics problems could be solved with sufficient accuracy by factorizing the neutron flux into the product of a time-dependent amplitude function solved by point kinetics on a fine time scale and a shape function with weak time-dependence solved by neutron diffusion on a coarse time scale [3]–[5]. Ott and Meneley improved the accuracy of the method by introducing updates to the shape function with an

iterative procedure to develop the Improved Quasi-Static Method (IQM) [6].

Gehin’s 1992 dissertation introduced the Predictor-Corrector Quasi-static Method (PCQM), which avoided the costly iterative procedure to determine the shape function by updating the entire neutron flux on the coarse time scale as a “predictor” and correcting the solution with the fine time-resolution amplitude function [7]. The method received little attention at the time of publication but was re-investigated by Dulla et al. in the mid-2000’s [8], [9]. In their analysis, Dulla et al. showed the PCQM to be well-suited for implementations requiring high-fidelity solvers because it typically requires fewer evaluations on the coarse time scale than the IQM. On the other hand, several studies over the last few years have suggested that the IQM might actually be better-suited for multiphysics calculations because it allows for iterative communication between the neutronics and thermal hydraulics solvers [10]–[12]. Between the two, the clearly-superior technique has yet to be established, as both methods have started to see transport-level applications in recent years.

Zhu improved the speed and accuracy of the PCQM in the deterministic Michigan Parallel Characteristics Transport (MPACT) code by introducing an intermediate time-discretization [13]. In what Zhu calls the TML method, the CMFD equations are used to propagate the amplitude function forward in time at the intermediate time scale to preserve some spatial and energy dependence in between transport solutions. Shen showed further improvement by introducing a fourth, energy-integrated, diffusion level to the TML [14]. Shen additionally characterized Lie and Strang operator splitting for multiphysics schemes that might be relevant for future applications of this work.

With huge strides in computational capacity, the previous decade has also seen several reactor kinetics implementations that make use of Monte Carlo transport solvers. Sjenitzer and Hoogenboom showed that explicitly tracking the inventory of delayed neutron precursors produces very accurate results, but is prohibitively expensive for practical applications [15], [16]. Hackemack et al. implemented the PCQM with fixed-source Monte Carlo transport at the coarse time step but found that the performance was limited by long fission chains [17]. Jo et al. solved the issue of non-terminating fission chains by solving the Monte Carlo step with fission source iteration [18]. Shaner developed a frequency transform method in 2018 (based on a deterministic technique developed by Ban et al. [19] that uses time-dependent CMFD equations (in a distinct formulation from those developed by Zhu) to propagate the space- and energy-dependent frequencies forward in time so that they can inform time-dependent updates to the Monte Carlo solution [20]. Kreher and Forget are currently working to incorporate thermal-hydraulic feedback into Shaner’s frequency transform method [21].

Table 1.1 shows a comparison of characteristics of the reactor kinetics solvers implemented over the last several decades. The topics presented in this dissertation are in the bottom row. The work presented in subsequent chapters uses a modified version of the transient fission source iteration routine developed by Jo

Table 1.1: A literature comparison of methods for solving reactor kinetics problems. *Dabaghi et al. solves the parabolic diffusion operator with Monte Carlo (not the hyperbolic transport operator). **This work only investigates a serial implementation of the parareal algorithm.

Reference	Year	Implementation Characteristics							
		Quasi-static	IQM	PCQM	Diffusion	Deterministic Transport	Monte Carlo Transport	Multi-fidelity	Parallel-in-Time
Henry [3]	1958	✓			✓				
Henry and Curlee [4]	1958	✓			✓				
Yasinsky and Henry [5]	1965	✓			✓				
Ott and Meneley [6]	1969	✓	✓		✓				
Gehin [7]	1992	✓		✓	✓				
Dulla et al. [9]	2008	✓		✓	✓				
Sjenitzer and Hoogenboom [15]	2013						✓		
Hackemack et al. [17]	2013	✓		✓			✓		
Baudron et al. [23]	2014				✓				✓
Sjenitzer and Hoogenboom [16]	2015						✓		
Zhu [13]	2015	✓		✓		✓		✓	
Jo et al. [18]	2016	✓		✓			✓		
Patricot et al. [10]	2017	✓	✓		✓			✓	
Shaner [20]	2018	✓		✓			✓	✓	
Prince and Ragusa [11]	2019	✓	✓			✓		✓	
Shen et al. [14]	2021	✓		✓		✓		✓	
Kooreman and Griesheimer [12]	2021	✓		✓			✓		
Kreher and Forget [21]	2022	✓		✓			✓	✓	
Dabaghi et al. [24]	2023						✓*		✓
This work	2022	✓		✓			✓	✓	✓**

et al. to enable transient multi-level calculations in the Shift Monte Carlo solver maintained at ORNL [22]. Until now, the TML has only been implemented in the MPACT deterministic solver, using the Method of Characteristics (MOC) for the transport level. Further, this dissertation investigates the use of parallel-in-time integration techniques to accelerate the propagation of TML subsolvers.

1.2.2 Parallelism in Neutron Transport

Parallelism has been a critical tool for computational neutron transport for several decades. Parallelism was introduced to deterministic solvers across the energy domain by Wienke and Hiromoto in 1986 [25], the angular domain by Azmy in 1988 [26], and the spatial domain by Yavuz and Larsen in 1989 [27]. Today massively-parallel deterministic solvers across all of these domains of phase space are relatively saturated for

Central Processing Units (CPUs) and most new research avenues for parallelization is transitioning towards Graphics Processing Units (GPUs).

For Monte Carlo solvers, the dominant parallelism technique across particle histories was introduced by Martin et al. in 1987 [28]. Spatial decomposition algorithms were studied by Brunner et al. in 2006 [29]. Shift, the Monte Carlo transport solver used throughout this work, is one of the few codes to implement spatial decomposition methods [30]. The popularity of hybrid computing architectures has placed renewed attention to vectorized techniques developed by Brown and Martin in 1984 [31] that parallelize across the next event to be processed in a Monte Carlo particle history [32]–[34].

Parallelism across the temporal domain remains understudied in the neutron transport community. Baudron et al. studied the parareal algorithm applied to a neutron diffusion solver with a rediscrretized temporal domain in 2014 [23], [35] but no study has been performed to test its applicability to the neutron transport equation. In a soon-to-be-published journal article, Dabaghi et al. propose a hybrid Galerkin/Monte Carlo parareal application for 1D Brownian motion, which is the random process behind diffusion [24]. In all of these previous works, the unknown solution is the neutron flux distribution with full dependence on the relevant phase space.

In the acceleration technique proposed in Chapter 4, the unknown solution is instead the amplitude function that comes from a quasi-static flux factorization and is by definition either strongly or exclusively dependent on time. The parareal coupling is then applied between various levels of a multi-level scheme, where each level is characterized by a different fidelity of approximation to the underlying physics. To the author’s knowledge, this is a unique application of the parareal algorithm.

1.2.3 Parallel-in-Time Applications in Other Fields

Parallel-in-time algorithms have received increasing attention over the last two decades as performance gains from parallel-in-space algorithms begin to saturate and the number of available processing units available on supercomputing architectures continues to increase. The most well-studied of these is the *parareal* algorithm, introduced by Lions et al. in 2001 [36]. The parareal algorithm is a predictor-corrector technique that uses a coarse time step with a computationally-inexpensive propagation, and a fine time step with a computationally-expensive propagation to iteratively converge towards the solution with fine time step resolution. The algorithm typically provides significant speedup if the number of outer parareal iterations is kept low. While the algorithm has received limited attention outside of the applied math community, engineering applications have been seen for select Partial Differential Equations (PDEs).

Despite the neutron transport equation being burdened with the hyperbolic streaming term, lessons

can still be learned from applications to other fields. Parallel-in-time applications to the 2D Navier-Stokes equation by Guiber and Trumeur in 2007 [37], and Newton’s equations of motion for molecular dynamics by Bylaska et al. in 2013 [38] both found the algorithm to have limited applicability to engineering systems of realistic complexity because of expensive integrators on the fine time scale. Additionally, a 2014 application to the 3D heat equation by Speck et al. cautioned that it is unclear how to optimally allocate processors when using parallelism in additional domains (space) [39]. For this work, the quasi-static approximation allows for the decoupling of the spatial domain from temporal domain between the CMFD and EPKE levels of the TML method. This EPKE amplitude solution can then be accelerated as a system of coupled Ordinary Differential Equations (ODEs) as opposed to a system of coupled PDEs.

Recent developments in the applied math community over the last decade have studied a generalized parallel-in-time algorithm, Multigrid Reduction in Time (MGRIT) with multiple time discretization levels, and shown convergence analysis for nonlinear hyperbolic PDEs [40], [41]. The parareal algorithm can be interpreted as a two-level MGRIT formulation, though it has been shown that adding multiple time discretization levels only sometimes leads to faster algorithms [42]. To the author’s knowledge, the two-level parareal algorithm is the only parallel-in-time algorithm studied in the neutron transport literature and should be considered as a potential avenue for future research.

1.2.4 Correlated Sampling for k -Eigenvalue Calculations

In Chapter 4, a correlated sampling technique is proposed as a mechanism for generating coarse estimates of Monte Carlo tallies propagated forward in time that can be used as input for the lower-order solvers in the TML. Correlated sampling is a Monte Carlo-specific technique, extended from general perturbation theory, for simulating N_{pert} small perturbations to the parameters that define a system, such as material densities, nuclear data, or geometry features, with the same random sequence used to simulate the unperturbed system [43], [44]. By using the same random sequence, $N_{\text{pert}} + 1$ correlated Monte Carlo estimates can be obtained for the cost of one.

Correlated sampling is well-established for fixed-source calculations, but work is still being done to generalize the technique for eigenvalue calculations, where the indirect perturbation effect on the fission source distribution must be accounted for across fission generations [45]. Nakagawa et al. and Kitada et al. both proposed methods for tracking indirect adjustment factors across fission generations [46] [47], but the fluctuations in the propagated adjustment factors increase rapidly with increasing generations. This led Kitada et al. to use spatially-averaged propagated adjustment factors, but the proper spatial resolution is problem-specific and difficult to pinpoint.

Nagaya and Mori developed a technique for estimating the perturbed fission source effect by passing a normalized initial weight to fission source particles between generations, which reduced the fluctuations of the propagated adjustment factors [48]. Their technique, however, was only applied and tested for k -eigenvalue tallies and not for other, more general tallies of interest during an eigenvalue calculation. In this work, reaction rate, surface current, and point kinetics tallies are all of interest to be used as input for both the CMFD and EPKE subsolvers of the TML. Nagaya and Mori’s technique is expanded for general Monte Carlo tallies in Section 4.3 and implemented in an in-house Monte Carlo solver. A roadmap for future production-level implementations to be used as Monte Carlo perturbation estimates in parallel-in-time schemes is proposed at the end of Chapter 4.

1.3 Unique Features of This Work

Several features of this dissertation work expand upon existing numerical methods and enter regimes not previously explored. Together, they help to pave the way towards making high-fidelity transient Monte Carlo analyses of DPC and reactor systems more tractable and hopefully serve as a roadmap for future production-level implementations. These features include:

- The foundation that this dissertation work is built upon is an implementation of the TML method using a Monte Carlo solver as the highest-fidelity (in the spatial, angular, and energy domains) solver in the nested PCQM scheme. To date, the TML method has only seen one implementation, where the highest-fidelity solver was a deterministic MOC solver. The implementation presented in Chapter 3 is therefore unique within the neutron transport literature. Limitations of the TML method that are specific to the Monte Carlo implementation are encountered and discussed as well.
- The calculations performed for this work push the limits of what similar works developing transient Monte Carlo methods have analyzed. The majority of research works performing transient Monte Carlo calculations evaluate their methods for 1D, 2D, or small 3D geometries with axial symmetry and low dominance ratios. At the end of Chapter 3, a large DPC geometry with asymmetry in the fuel composition for both the radial and axial profiles, is simulated a Monte Carlo TML implementation. While this calculation is performed with multigroup physics, it represents a much larger and more complex system than many of those analyzed in previous works.
- The transient Monte Carlo method used in this work is implemented on a frozen research branch of the production-level Monte Carlo code, Shift [22]. Shift is designed primarily for high-performance computing applications and supports a wide array of reactor physics features, including coupling

interfaces with multi-physics solvers. Many of the transient Monte Carlo works described in Section 1.2.1 are implemented in research-oriented codes with limited physics features and high-performance computing scalability. While the implementation used in this work was not subject to rigorous code-review and exists on a static branch roughly two and a half years behind the most recent git commit on the main development branch, it lays a roadmap for future implementations for similar performance-oriented codes. Hopefully, valuable lessons can be learned from the existing infrastructure for advancing the k -eigenvalue calculation forward in time, capturing and processing state information at time census events, modifying time-dependent material densities and interfacing between the various levels of low-fidelity TML subsolvers.

- To date, parareal algorithms have been almost exclusively limited to deterministic solvers because the statistical uncertainties inherent to stochastic solvers make iterative convergence difficult and their computational expense restricts their use as the coarse time step propagator. This work implements the parareal algorithm to accelerate the propagation of subsolvers of hybrid stochastic/deterministic calculations with multigrid time discretizations.
- With respect to reactor kinetics, a standard parareal implementation has been applied to time-dependent neutron diffusion propagators, where the difference in fidelity between the fine and coarse propagators was a rediscritization of the time step [23], [35]. The unknown solution of the referenced studies was the scalar flux with full positional, energy and time dependence. This work investigates using the parareal algorithm to accelerate subsolvers of a nested PCQM implementation. Therefore, the unknown solution is instead the amplitude function, assumed to have strong temporal dependence, of the level being accelerated. In this way, parareal algorithms are now used to enable transport-fidelity calculations.
- Correlated sampling techniques have existed in the neutron transport literature for several decades. In more recent decades, significant attention has been devoted to developing perturbation techniques that can be applied to fission source iteration problems to estimate changes in the k -eigenvalue given perturbations to some problem parameter. To the author’s knowledge, these techniques have not been generalized for other Monte Carlo tallies, until now. Although incomplete, this work also speculates on what modifications to the generalized correlated sampling technique would look like to estimate an indirect time-propagated effect on the perturbed tally estimates.

1.4 Dissertation Outline

The remainder of this dissertation can be summarized as follows:

- Chapter 2 gives a foundation for neutron transport fundamentals and defines the terms present in the equations that govern the process of neutron transport. Specific formulations and approximations made to the neutron transport equation that are relevant to understanding this dissertation work are derived. Finally, a brief overview of the types of numerical techniques used to solve these equations are discussed.
- Chapter 3 discusses the state-of-the-art techniques used today for evaluating reactor transients. One of these techniques, the TML method, is implemented for the first time with the production-level Monte Carlo solver for multigroup physics. The implementation details are discussed and the method is verified via the C5G7-TD3 transient benchmark against an MOC solver that also uses the TML to propagate the solution forward in time. A pseudo Doppler broadening effect is implemented in the code to test the implementation’s ability to handle prompt feedback-driven transients. Finally, the step change in reactivity is modeled for a large 3D DPC showing that the method is feasible for practical transient criticality applications. Difficulties with the CMFD solver of the TML method that are unique, and potentially limiting, to the Monte Carlo implementation are also discussed.
- Chapter 4 discusses the parareal algorithm and pioneers its use as a mechanism to accelerate propagations of the TML subsolvers. Unfortunately, the parareal algorithm is only implemented in serial, but projected speedups are given. The implementation is verified with a coupling between two EPKE solvers with rediscritized time domains. The serial implementation was then migrated to the Shift codebase, where it was studied as an acceleration scheme between two CMFD levels. At the end of the chapter, a correlated sampling technique is proposed to extend the parareal coupling between two Monte Carlo levels. The proposed technique generalizes a method for estimating perturbation for static k_{eff} tallies to arbitrary static tallies. While special consideration for time-propagated perturbations was not developed, speculations on the required modifications to the correlated sampling algorithm are given.
- Chapter 5 provides a summary of this dissertation work and postulates potential avenues of research to build upon the results discussed throughout the dissertation.

Chapter 2

Transport Fundamentals

In this chapter, the equations governing the time-dependent behavior of neutrons in a multiplying system will be derived. The equations take various forms depending on the specific system being analyzed and the approximations used to solve them. First, a heuristic derivation of the time-dependent neutron transport equation is presented. Next, the adjoint neutron transport equation and various approximations to the neutron transport equation are derived. These include the multigroup approximation, the P_N/SP_N approximations, and the diffusion approximation. Of particular interest to this dissertation are derivations of the Coarse Mesh Finite Difference (CMFD) equations (Section 2.8) and the Exact Point Kinetics Equations (EPKEs) (Section 2.4), as they will be relied upon heavily in the multi-level method discussed in Chapter 3. Finally, a brief overview of stochastic and deterministic methods is given.

2.1 Boltzmann Equation

The distribution of particles $N(\mathbf{x}, \mathbf{p}, t)$ in a given system can be described by the Boltzmann transport equation. The Boltzmann equation gives the probability that a particle occupies a given volume dV about position $\mathbf{x}$ and has momentum nearly equal to a given momentum vector $\mathbf{p}$. The Boltzmann equation was originally developed as the foundation for the kinetic theory of gases, where the properties of dilute gases are defined by analyzing the elementary collision processes between pairs of molecules. In modern literature, the Boltzmann equation has been used to describe kinematic processes such as the time-dependent changes in quantities in a system such as energy, charge, or number of particles. In its general form, the Boltzmann equation is a nonlinear integro-differential equation, meaning that in many contexts, the equation suffers from problems of existence and uniqueness of solutions. Fortunately, transporting neutrons isn't as complicated as "say" transporting charged particles in a 6000 K plasma on the surface of the sun.

In the neutron transport context, several simplifying assumptions can be made that retain most of the accuracy of the equation for reactor physics applications. These assumptions are such that the resulting

equation is linear and therefore much less difficult to solve. These assumptions include:

- Acceleration forces, such as gravity, are negligible for the problems being considered.
- The locations of neutron-nucleus interactions are assumed to be randomly located within a Poisson distribution and behave as well-defined isolated two-body interactions.
- Statistical fluctuations in the radiation population can be ignored as the radiation density is sufficiently high. This assumption is particularly applicable within nuclear power reactors, where neutron populations are on the order of $\sim 10^{15}$ and deviations in the neutron population are relatively small compared to the mean population. The applicability of this assumption is less sound in the case of weak fission neutron sources, where fission chains are well-separated and there is considerable variation in the length of fission chains.
- Material distributions are independent of time on the time scale of neutron interactions.
- Material distributions are random and isotropic such that there is no preferred direction of scattering and neutron cross sections are independent of the incident direction.
- Neutrons interact almost exclusively with the background materials and not with other neutrons. This is a well-founded assumption, as even in high neutron population environments (such as a nuclear power reactor), the atomic density of the transport medium is $\sim 10^{22}$ and far exceeds the neutron populations.
- The neutron reaction rate density is sufficiently small that the isotopic composition of the transport medium is not appreciably changed. An exception to this assumption are problems where long-enough time scales for fuel burnup need to be considered.
- Neutrons are treated as points, meaning that their free motion between interactions can be described by classical mechanics. Consequently, neutron properties that require wave-like or quantum mechanical consideration, such as spin and polarization, are assumed to influence only interaction cross sections. Additionally, these cross sections are assumed to be quantities for the purposes of neutron transport theory.

In addition, it is commonplace in the neutron transport literature to substitute the momentum vector with a unit direction vector representing the neutron's direction of flight $\hat{\Omega}$ and the neutron's energy-dependent non-relativistic speed $v(E) = \sqrt{(2E/m)}$, where m is the mass of the neutron. The direction of flight is a function of two independent variables: θ describing the angle the neutron is traveling relative to the polar axis $\hat{\mathbf{i}}$, and γ describing the azimuthal angle. In practice, θ is represented by the variable $\mu = \cos \theta$ so that

the individual component definitions of $\hat{\Omega}$ are written as

$$\begin{aligned}\hat{\Omega} &= \Omega_x \hat{i} + \Omega_y \hat{j} + \Omega_z \hat{k} \\ &= \mu \hat{i} + \left(\sqrt{1 - \mu^2} \cos \gamma \right) \hat{j} + \left(\sqrt{1 - \mu^2} \sin \gamma \right) \hat{k}.\end{aligned}\tag{2.1}$$

Thus, the phase space of the neutron probability distribution is dependent on seven independent dimensions: three in space, two in direction, one in energy, and one in time. The spatial, angular, and energy phase-space for a given point in time t is depicted in Fig. 2.1.

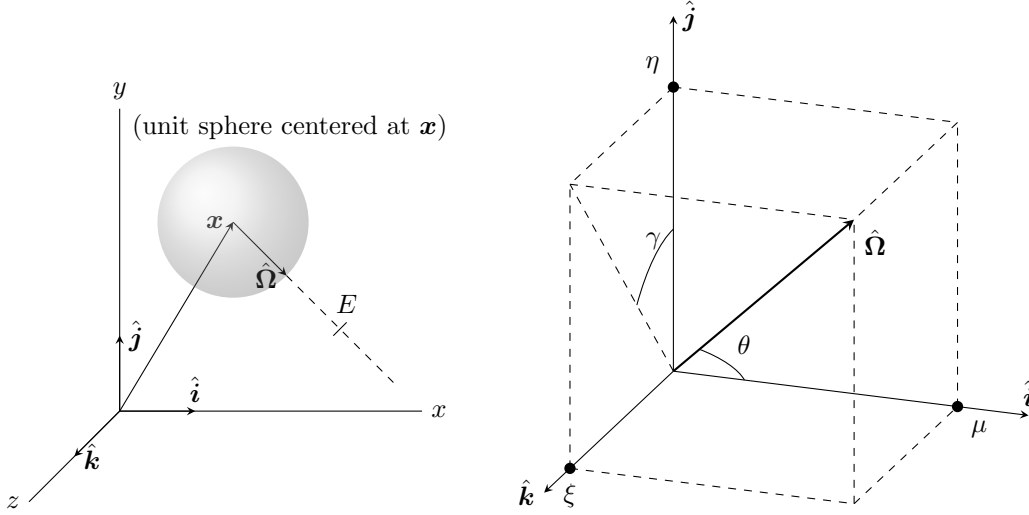


Figure 2.1: Particle spatial, angular, and energy phase space.

2.2 Neutron Transport Equation

The probability distribution of neutrons is commonly referred to as the angular neutron density $N(\mathbf{x}, \hat{\Omega}, E, t)$. In words, this translates to the probability that at time t , neutrons are located within a differential volume dV about position $\mathbf{x}$, traveling within a solid angle $d\Omega$ about the direction $\hat{\Omega}$, with energies between E and $E + dE$. As $N(\mathbf{x}, \hat{\Omega}, E, t)$ is a *probability distribution* it can be described in words as

$$\begin{aligned}N(\mathbf{x}, \hat{\Omega}, E, t) dV d\Omega dE &= \text{the expected number of neutrons in } dV d\Omega dE \\ &\text{about } (\mathbf{x}, \hat{\Omega}, E) \text{ at time } t.\end{aligned}\tag{2.2}$$

In the neutron transport context, the unknown quantity of interest is the angular flux $\psi(\mathbf{x}, \hat{\Omega}, E, t)$, which

is a product of the neutron speed v and the neutron probability distribution,

$$\psi(\mathbf{x}, \hat{\Omega}, E, t) = v(E) N(\mathbf{x}, \hat{\Omega}, E, t). \quad (2.3)$$

The angular flux has the units $[\text{cm}^{-2} \text{ MeV}^{-1} \text{ s}^{-1}]$ and is commonly thought of as having two physical interpretations. The first is as an angular fluence rate density. In plainer words, this translates to the rate at which particles pass through a differential surface area dA oriented normal to the particle's direction of flight per unit energy, per unit solid angle, per unit area. The second comes by considering the definition in words

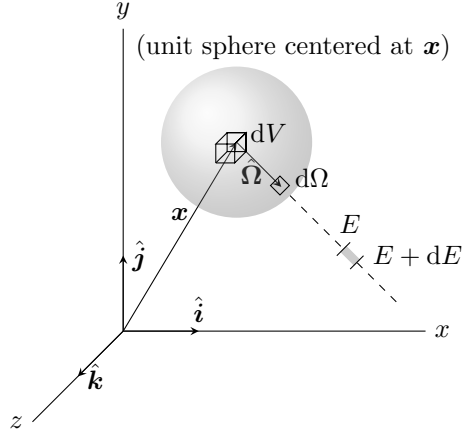


Figure 2.2: Depiction of the angular flux as an angular fluence rate density.

from Eq. (2.2) and applying it to Eq. (2.3):

$$\begin{aligned} \psi(\mathbf{x}, \hat{\Omega}, E, t) dV d\Omega dE dt &= v(E) \left[N(\mathbf{x}, \hat{\Omega}, E, t) dV d\Omega dE \right] dt \\ &= [v(E) dt] \left[N(\mathbf{x}, \hat{\Omega}, E, t) dV d\Omega dE \right] \\ &= [\text{distance (path length) traveled by one neutron in time } dt] \\ &\quad \times \left[\text{expected number of neutrons in } dV d\Omega dE \text{ about } (\mathbf{x}, \hat{\Omega}, E) \right] \\ &= \text{total path length traveled by neutrons in } dV d\Omega dE \\ &\quad \text{about } (\mathbf{x}, \hat{\Omega}, E) \text{ during time increment } dt \text{ about } t. \end{aligned} \quad (2.4)$$

Therefore, the angular flux is also referred to as a path length rate density per unit energy, per unit solid angle, per unit volume. This interpretation becomes particularly useful when calculating reaction rates within a given volume of interest.

The neutron transport equation describes the statistical behavior of the neutron distribution in the system

by accounting for rates of gains and losses within a differential subset of phase space. When multiplied by the volume, solid angle, and energy range of the differential phase space, the neutron transport equation describes a rate balance such that the accumulation of particles between two points in time t and $t + \Delta t$ is summarized as

$$\text{rate of accumulation} = \underbrace{\text{rate streaming in} - \text{rate streaming out}}_{-\text{rate of net leakage}} - \text{rate consumed} + \text{rate generated}. \quad (2.5)$$

Integrating over the entire phase space, the mathematical expressions of the terms in Eq. (2.5) are given as

- *rate of accumulation*: The rate of accumulation is simply the difference in the neutron density at times $t + \Delta t$ and t divided by the difference in time Δt

$$\text{rate of accumulation} = \int_0^\infty \int_V \int_{4\pi} \frac{[N(\mathbf{x}, \hat{\Omega}, E, t + \Delta t) - N(\mathbf{x}, \hat{\Omega}, E, t)]}{\Delta t} d\Omega dV dE, \quad (2.6a)$$

where V is the system volume. By taking the limit as $\Delta t \rightarrow 0$ and multiplying the numerator and denominator by the velocity $v(E)$ Eq. (2.6a) becomes

$$\begin{aligned} \text{rate of accumulation} &= \int_0^\infty \int_V \int_{4\pi} \frac{1}{v(E)} \frac{v(E) \partial N(\mathbf{x}, \hat{\Omega}, E, t)}{\partial t} d\Omega dV dE \\ &= \int_0^\infty \int_V \int_{4\pi} \frac{1}{v(E)} \frac{\partial \psi(\mathbf{x}, \hat{\Omega}, E, t)}{\partial t} d\Omega dV dE. \end{aligned} \quad (2.6b)$$

- *rate of net leakage*: As shown in Eq. (2.5), the rate of net leakage out of the differential phase space is simply the difference in the rate of streaming into the differential control volume minus the rate of streaming out of the differential control volume. Alternatively, since one of the interpretations of the angular flux is as a fluence rate density through a differential surface area dA with unit normal vector $\hat{\mathbf{n}}$, the net rate of leakage from the differential control volume is equivalently expressed as the rate at which particles in $dV d\Omega dE$ about $(\mathbf{x}, \hat{\Omega}, E)$ at time t stream through dA . This is written in math as the product $(\hat{\Omega} \cdot \hat{\mathbf{n}}) \psi(\mathbf{x}, \hat{\Omega}, E, t) d\Omega dA dE$. The net rate of leakage integrated over all of phase space is then

$$\begin{aligned} \text{rate of net leakage} &= \text{rate streaming in} - \text{rate streaming out} \\ &= \int_0^\infty \int_{\partial V} \int_{4\pi} (\hat{\Omega} \cdot \hat{\mathbf{n}}) \psi(\mathbf{x}, \hat{\Omega}, E, t) d\Omega dA dE, \end{aligned}$$

where ∂V is the system boundary. The divergence theorem can be applied to the net leakage term so

that it can be rewritten in terms of a volume integral

$$\text{rate of net leakage} = \int_0^\infty \int_V \int_{4\pi} \hat{\Omega} \cdot \nabla \psi(\mathbf{x}, \hat{\Omega}, E, t) \, d\Omega \, dV \, dE. \quad (2.6c)$$

- *rate consumed*: The rate at which neutrons are consumed within the control volume can be thought of as the rate at which neutrons undergo any collision. If a neutron re-emerges from a collision within the same differential phase space through scattering or fission, it will be treated appropriately as a source term. Equation (2.4) describes the angular flux in words as the rate density at which path length ℓ is generated within the differential volume. Let $\Sigma_t(\mathbf{x}, E, t)$ be the macroscopic total cross section, which describes the number of interactions per unit distance traveled (or path length generated) by a particle such that

$$\begin{aligned} \Sigma_t(\mathbf{x}, E, t) \, d\ell &= \text{the probability that a neutron at } (\mathbf{x}, E, t), \\ &\text{traveling distance } d\ell, \text{ will collide with a nucleus.} \end{aligned} \quad (2.6d)$$

Since $\psi(\mathbf{x}, \hat{\Omega}, E, t)$ has an interpretation as a path length rate density, the rate at which neutrons are consumed can then be written as the product between macroscopic cross section, angular flux, and differential phase space

$$\text{rate consumed} = \int_0^\infty \int_V \int_{4\pi} \Sigma_t(\mathbf{x}, E, t) \psi(\mathbf{x}, \hat{\Omega}, E, t) \, d\Omega \, dV \, dE. \quad (2.6e)$$

- *rate generated*: Let $S(\mathbf{x}, \hat{\Omega}, E, t)$ describe the source rate density accounting for all neutrons “born” in $dV \, d\Omega \, dE$ about $(\mathbf{x}, \hat{\Omega}, E)$ at time t whether they result from scattering, prompt fission, delayed fission, or internal source terms. The rate at which neutrons are generated can then be described as

$$\text{rate generated} = \int_0^\infty \int_V \int_{4\pi} S(\mathbf{x}, \hat{\Omega}, E, t) \, d\Omega \, dV \, dE, \quad (2.6f)$$

By substituting the definitions from Eqs. (2.6) into Eq. (2.5), rearranging terms, and removing the integrals, the neutron transport equation thus far is expressed as

$$\frac{1}{v(E)} \frac{\partial \psi}{\partial t}(\mathbf{x}, \hat{\Omega}, E, t) + \hat{\Omega} \cdot \nabla \psi(\mathbf{x}, \hat{\Omega}, E, t) + \Sigma_t(\mathbf{x}, E, t) \psi(\mathbf{x}, \hat{\Omega}, E, t) = S(\mathbf{x}, \hat{\Omega}, E, t). \quad (2.7)$$

The source density term is further expanded as

$$S(\mathbf{x}, \hat{\Omega}, E, t) = \text{in-scattering} + \text{prompt fission} + \text{delayed fission} + \text{internal source}, \quad (2.8)$$

where these individual source terms are further defined as:

- *in-scattering*: Suppose that a neutron with incident energy E' and direction vector $\hat{\Omega}'$ emerges from a scattering event with a new energy E and direction vector $\hat{\Omega}$. The double differential macroscopic scattering cross section $\Sigma_s(\mathbf{x}, \hat{\Omega}' \cdot \hat{\Omega}, E' \rightarrow E, t)$ is defined such that

$$\begin{aligned} \Sigma_s(\mathbf{x}, \hat{\Omega}' \cdot \hat{\Omega}, E' \rightarrow E, t) \, d\ell \, d\Omega \, dE &= \text{the probability that a neutron with incident direction } \hat{\Omega}', \\ &\text{incident energy } E', \text{ traveling distance } d\ell, \\ &\text{will scatter into } d\Omega \, dE \text{ about } (\hat{\Omega}, E) \text{ at time } t. \end{aligned} \quad (2.9)$$

The in-scattering term can then be derived by integrating the in-scattering reaction rate over all incident directions and energies:

$$\text{in-scattering} = \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{x}, \hat{\Omega}' \cdot \hat{\Omega}, E' \rightarrow E, t) \, \psi(\mathbf{x}, \hat{\Omega}', E', t) \, d\Omega' \, dE'. \quad (2.10)$$

Note that $\Sigma_s(\mathbf{x}, \hat{\Omega}' \cdot \hat{\Omega}, E' \rightarrow E, t)$ is only dependent on one directional variable $\hat{\Omega}' \cdot \hat{\Omega}$ as opposed to the more complicated term $\hat{\Omega}' \rightarrow \hat{\Omega}$ that is dependent on four direction variables. This is a result of one of the assumptions discussed at the beginning of this chapter for restricting the Boltzmann equation to neutron transport: material distributions are random and isotropic. Thus, neutron scattering has the important property of being equiprobable in the azimuthal direction, or “rotationally invariant”. The outgoing direction of flight is therefore only dependent on the cosine of the scattering angle $\mu_0 = \hat{\Omega}' \cdot \hat{\Omega}$ between the incoming and outgoing directions.

- *prompt fission*: When a neutron of energy E initiates a fission event, $\nu(\mathbf{x}, E, t)$ are released on average. Of these $\nu(\mathbf{x}, E, t)$ neutrons, a fraction of them $\nu(\mathbf{x}, E, t) [1 - \beta(\mathbf{x}, E, t)]$ are “prompt” and emitted within 1×10^{-15} s of the fission event. The remaining fraction $\nu(\mathbf{x}, E, t) \beta(\mathbf{x}, E, t)$ are “delayed” and emitted anywhere from 0.1 s to 60 s after the fission event. The numerical stiffness introduced by the difference in the timescales of emission between prompt and delayed neutrons is one of the confounding factors that make time-dependent neutron transport problems difficult to solve. It is assumed that fission source neutrons, prompt or delayed, are born with a known function of energy, χ . For prompt

neutrons, this function is defined be the isotope-specific prompt fission spectrum $\chi_p(\mathbf{x}, E, t)$. Fission neutrons are typically assumed to be emitted isotropically. Let $\Sigma_f(\mathbf{x}, E, t)$ be the macroscopic fission cross section defined such that

$$\Sigma_f(\mathbf{x}, E, t) d\ell = \text{the probability that a neutron at } (\mathbf{x}, E, t), \text{ traveling distance } d\ell, \text{ will cause a fission.} \quad (2.11)$$

The prompt fission source term can then be defined as

$$\text{prompt fission} = \frac{1}{4\pi} \chi_p(\mathbf{x}, E, t) \int_0^\infty \int_{4\pi} [1 - \beta(\mathbf{x}, E', t)] \nu_f \Sigma_f(\mathbf{x}, E', t) \psi(\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE'. \quad (2.12)$$

- *delayed fission*: Delayed neutrons are emitted isotropically from the radioactive decay of certain unstable daughter nuclei that are produced during fission events. These unstable daughter nuclei are often grouped into $J = 6$ to $J = 8$ distinct “delayed neutron precursor” groups classified by their radioactive decay constants λ_j , where $1 \leq j \leq J$. Each delayed neutron precursor group also has a corresponding fission spectrum $\chi_{d,j}(E)$ and decay fraction, $\beta_j(\mathbf{x}, E, t)$, where $\beta(\mathbf{x}, E, t) = \sum_j^J \beta_j(\mathbf{x}, E, t)$. Let the spatial concentration of delayed neutron precursors for group j be $C_j(\mathbf{x}, t)$. The delayed fission source term can then be defined as

$$\text{delayed fission} = \frac{1}{4\pi} \sum_{j=1}^J \frac{\chi_{d,j}(E)}{4\pi} \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t). \quad (2.13)$$

The delayed neutron precursor balance equation connects the generation and consumption of delayed neutron precursors and is described by

$$\frac{\partial C_j}{\partial t}(\mathbf{x}, t) = \int_0^\infty \int_{4\pi} \beta_j(\mathbf{x}, E', t) \nu_f \Sigma_f(\mathbf{x}, E', t) \psi(\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE' - \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t). \quad (2.14)$$

The first term in Eq. (2.14) corresponds to the direct production of the delayed neutron precursor daughter nuclei from fission while the second term corresponds to the radioactive decay of these daughter nuclei.

- *internal source*: Internal sources are typically isotopes producing neutrons via radioactive decay and are often emitted isotropically. It is standard practice to include a $\frac{1}{4\pi}$ multiplicative factor so that integration over the spatial, directional, and energy domains gives the physical interpretation of the

number of neutrons emitted from the internal source into $d\Omega dV dE$ about $(\mathbf{x}, E)$ at time t

$$\text{internal source} = \frac{1}{4\pi} Q(\mathbf{x}, E, t) . \quad (2.15)$$

- There are other neutron-nucleus interactions that produce other sub-atomic particles, such as protons (n,p), alpha (n, α), or photons (n, γ). There are also neutron-nucleus interactions that produce multiple neutrons such as (n,2n), (n,3n), etc. With the exception of (n, γ) interactions, these events are generally rare and usually caused by neutrons outside of the energy range of those produced in commercial nuclear reactors. These types of interactions are therefore ignored throughout this dissertation for convenience.

The transport equation from Eq. (2.7) is finally written as the coupled system of $J + 1$ integro-differential equations defined for $\mathbf{x} \in V$, $\hat{\Omega} \in 4\pi$, $0 \leq E < \infty$, $0 \leq t$:

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial \psi}{\partial t}(\mathbf{x}, \hat{\Omega}, E, t) + \nabla \cdot \hat{\Omega} \psi(\mathbf{x}, \hat{\Omega}, E, t) + \Sigma_t(\mathbf{x}, E, t) \psi(\mathbf{x}, \hat{\Omega}, E, t) = \\ \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{x}, \hat{\Omega} \cdot \hat{\Omega}', E' \rightarrow E, t) \psi(\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE' \\ + \frac{1}{4\pi} \chi_p(\mathbf{x}, E, t) \int_0^\infty \int_{4\pi} [1 - \beta(\mathbf{x}, E', t)] \nu \Sigma_f(\mathbf{x}, E', t) \psi(\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE' \\ + \frac{1}{4\pi} \sum_{j=1}^J \chi_{d,j}(\mathbf{x}, E, t) \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t) + \frac{1}{4\pi} Q(\mathbf{x}, \hat{\Omega}, E, t) , \end{aligned} \quad (2.16a)$$

$$\frac{\partial C_j}{\partial t}(\mathbf{x}, t) = \int_0^\infty \int_{4\pi} \beta_j(\mathbf{x}, E', t) \nu_f \Sigma_f(\mathbf{x}, E', t) \psi(\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE' - \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t) . \quad (2.16b)$$

In order to solve Eq. (2.16a), both boundary and initial conditions must be specified. For the boundary condition, the angular flux is constrained to be some known function ψ^b along the boundary ∂V for all incoming directions

$$\psi(\mathbf{x}, \hat{\Omega}, E, t) = \psi^b(\mathbf{x}, \hat{\Omega}, E, t) , \quad \mathbf{x} \in \partial V , \quad \hat{\Omega} \cdot \hat{\mathbf{n}} < 0 , \quad 0 \leq E < \infty , \quad 0 < t . \quad (2.16c)$$

The initial condition constrains the flux to be some known function ψ^0 of all space, directions and energy at $t = 0$

$$\psi(\mathbf{x}, \hat{\Omega}, E, 0) = \psi^0(\mathbf{x}, \hat{\Omega}, E) , \quad \mathbf{x} \in V , \quad \hat{\Omega} \in 4\pi , \quad 0 \leq E < \infty . \quad (2.16d)$$

Equation (2.16a) is called the neutron transport equation with time dependence and sometimes referred to as the *linear Boltzmann equation*. In conjunction with the J balance equations for the delayed neutron precursor densities defined in Eq. (2.16b), the boundary constraints defined in Eq. (2.16c), and the initial

condition defined in Eq. (2.16d), the angular flux can be solved for using any appropriate numerical technique of choice. Eqs. (2.16) are the foundation for which the topic of reactor kinetics and this dissertation are built upon. Problems written in the general form of Eqs. (2.16) are referred to as *fixed-source* problems because their numerical solution involves imposing a known (or “fixed”) neutron source that is independent of the neutron flux on a transport medium. This is not the case for the eigenvalue calculations discussed in Sec. 2.2.2 and Sec. 2.2.3.

2.2.1 Steady-State Neutron Transport Equation

In steady-state problems, the time derivatives in Eqs. (2.16) are set to zero. The precursor densities can then be directly solved for from Eq. (2.16b)

$$\lambda_j C_j = \int_0^\infty \int_{4\pi} \beta_j(\mathbf{x}, E') \nu \Sigma_j(E') \psi(\mathbf{x}, \hat{\Omega}', E') d\Omega' dE'. \quad (2.17)$$

Equation (2.17) can then be substituted into Eq. (2.16a) to give

$$\begin{aligned} \nabla \cdot \hat{\Omega} \psi(\mathbf{x}, \hat{\Omega}, E) + \Sigma_t(\mathbf{x}, E) \psi(\mathbf{x}, \hat{\Omega}, E) = & \\ & \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{x}, \hat{\Omega} \cdot \hat{\Omega}', E' \rightarrow E) \psi(\mathbf{x}, \hat{\Omega}', E') d\Omega' dE' \\ & + \frac{1}{4\pi} \chi_p(\mathbf{x}, E) \int_0^\infty \int_{4\pi} [1 - \beta(\mathbf{x}, E')] \nu \Sigma_f(\mathbf{x}, E') \psi(\mathbf{x}, \hat{\Omega}', E') d\Omega' dE' \\ & + \frac{1}{4\pi} \sum_{j=1}^J \chi_{d,j}(E) \int_0^\infty \int_{4\pi} \beta_j(\mathbf{x}, E) \nu \Sigma_f(\mathbf{x}, E') \psi(\mathbf{x}, \hat{\Omega}', E') d\Omega' dE' \\ & + \frac{1}{4\pi} Q(\mathbf{x}, \hat{\Omega}, E), \quad \mathbf{x} \in V, \quad \hat{\Omega} \in 4\pi, \quad 0 < E < \infty, \end{aligned} \quad (2.18a)$$

$$\psi(\mathbf{x}, \hat{\Omega}, E) = \psi^b(\mathbf{x}, \hat{\Omega}, E), \quad \mathbf{x} \in \partial V, \quad \hat{\Omega} \cdot \hat{n} < 0, \quad 0 \leq E < \infty. \quad (2.18b)$$

Equation (2.18) is exact given that the steady state conditions are met, but a simplifying approximation is typically made to combine the prompt and delayed fission source terms. The β and β_j terms are all sometimes weighted by some space and energy-dependent function $\Psi(\mathbf{x}, E)$ to define

$$\bar{\beta}(\mathbf{x}) \equiv \frac{\int_0^\infty \beta(\mathbf{x}, E') \nu \Sigma_f(\mathbf{x}, E') \Psi(\mathbf{x}, E') dE'}{\int_0^\infty \nu \Sigma_f(\mathbf{x}, E') \Psi(\mathbf{x}, E') dE'}, \quad (2.19a)$$

$$\bar{\beta}_j(\mathbf{x}) \equiv \frac{\int_0^\infty \beta_j(\mathbf{x}, E') \nu \Sigma_f(\mathbf{x}, E') \Psi(\mathbf{x}, E') dE'}{\int_0^\infty \nu \Sigma_f(\mathbf{x}, E') \Psi(\mathbf{x}, E') dE'}, \quad (2.19b)$$

where $\Psi(\mathbf{x}, E)$ is ideally representative of the space and energy-dependent flux shape of the system. This

flux-weighting to remove energy dependence is a technique discussed further in Sec. 2.6.1. A cumulative fission spectrum $\chi(\mathbf{x}, E)$ is also defined to remove the explicit appearance of the $\bar{\beta}$ and $\bar{\beta}_j$ terms in Eq. (2.18a)

$$\chi(\mathbf{x}, E) \equiv (1 - \bar{\beta}(\mathbf{x})) \chi_p(\mathbf{x}, E) + \sum_{j=1}^J \bar{\beta}_j(\mathbf{x}) \chi_j(\mathbf{x}, E). \quad (2.20)$$

Using the definitions written in Eqs. (2.19) and Eq. (2.20), the commonly used form of the steady-state transport equation is

$$\begin{aligned} \nabla \cdot \hat{\Omega} \psi(\mathbf{x}, \hat{\Omega}, E) + \Sigma_t(\mathbf{x}, E) \psi(\mathbf{x}, \hat{\Omega}, E) = & \\ & \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{x}, \hat{\Omega} \cdot \hat{\Omega}', E' \rightarrow E) \psi(\mathbf{x}, \hat{\Omega}', E') d\Omega' dE' \\ & + \frac{1}{4\pi} \chi(\mathbf{x}, E) \int_0^\infty \int_{4\pi} \nu_f \Sigma_f(\mathbf{x}, E') \psi(\mathbf{x}, \hat{\Omega}', E') d\Omega' dE' \\ & + \frac{1}{4\pi} Q(\mathbf{x}, \hat{\Omega}, E), \quad \mathbf{x} \in V, \quad \hat{\Omega} \in 4\pi, \quad 0 < E < \infty, \end{aligned} \quad (2.21a)$$

$$\psi(\mathbf{x}, \hat{\Omega}, E) = \psi^b(\mathbf{x}, \hat{\Omega}, E), \quad \mathbf{x} \in \partial V, \quad \hat{\Omega} \cdot \hat{n} < 0, \quad 0 < E < \infty. \quad (2.21b)$$

Until the last 20 years or so, the vast majority of reactor physics calculations were limited to solving Eq. (2.21) due to limitations in computing power. Even to date, high-fidelity time-dependent solutions of Eq. (2.16) for practical reactor problems are only tractable on world-class supercomputers.

2.2.2 k -Eigenvalue Problems

In steady-state reactor calculations it is common to see the following form of Eq. (2.21) where the internal source Q and the boundary source ψ^b are set to zero and the fission source is multiplied by a constant factor $1/k_{\text{eff}}$

$$\begin{aligned} \nabla \cdot \hat{\Omega} \psi(\mathbf{x}, \hat{\Omega}, E) - \Sigma_t(\mathbf{x}, E) \psi(\mathbf{x}, \hat{\Omega}, E) = & \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{x}, \hat{\Omega} \cdot \hat{\Omega}', E' \rightarrow E) \psi(\mathbf{x}, \hat{\Omega}', E') d\Omega' dE' \\ & + \frac{1}{4\pi} \frac{\chi(\mathbf{x}, E)}{k_{\text{eff}}} \int_0^\infty \int_{4\pi} \nu \Sigma_f(\mathbf{x}, E') \psi(\mathbf{x}, \hat{\Omega}', E') d\Omega' dE', \\ & \mathbf{x} \in V, \quad \hat{\Omega} \in 4\pi, \quad 0 < E < \infty, \end{aligned} \quad (2.22a)$$

$$\psi(\mathbf{x}, \hat{\Omega}, E) = 0, \quad \mathbf{x} \in \partial V, \quad \hat{\Omega} \cdot \hat{n} < 0, \quad 0 < E < \infty. \quad (2.22b)$$

While these problems always have the null solution, $\psi = 0$, the intended solution is to find the largest value of k_{eff} such that a nonzero ψ exists. The k_{eff} here is often referred to as the criticality eigenvalue

and the resulting ψ is referred to as the eigenfunction or “fundamental mode”. k -Eigenvalue problems are formulated to assess the balance between the production of neutrons from fission and the loss of neutrons from capture and leakage. By adjusting the magnitude of k_{eff} , it is possible to force this balance. If $k_{\text{eff}} < 1$, the system is said to be “subcritical” and the fission source must be increased for a steady state solution to exist. Conversely, if $k_{\text{eff}} > 1$, the system is said to be “supercritical” and the fission source must be decreased for the steady state solution to exist. If $k_{\text{eff}} = 1$, the system is said to be “critical” and in a state where capture and leakage exactly balance fission. For many reactor kinetics applications, it is convenient to replace k_{eff} with reactivity ρ , where

$$\frac{1}{k_{\text{eff}}} = 1 - \rho, \quad (2.23)$$

such that the system is subcritical if $\rho < 0$, critical if $\rho = 0$, and supercritical if $\rho > 0$.

2.2.3 α -Eigenvalue Problems

Similarly to the k -eigenvalue formulation, for α -eigenvalue problems the internal source Q and boundary source ψ^b are set to zero. However, in the α -eigenvalue formulation the initial condition from Eq. (2.16d) is additionally ignored. Further, the time-dependence of angular flux is assumed to be separable from the other phase dimensions

$$\psi(\mathbf{x}, \hat{\Omega}, E, t) = \Psi(\mathbf{x}, \hat{\Omega}, E) e^{\alpha t}, \quad (2.24)$$

where α is now referred to as the α -eigenvalue. Eq. (2.24) is then used to formulate the problem

$$\begin{aligned} \nabla \cdot \hat{\Omega} \Psi(\mathbf{x}, \hat{\Omega}, E) - \left[\Sigma_t(\mathbf{x}, E) + \frac{\alpha}{v} \right] \Psi(\mathbf{x}, \hat{\Omega}, E) = \\ \int_0^\infty \int_{4\pi} \Sigma_s(\mathbf{x}, \hat{\Omega} \cdot \hat{\Omega}', E' \rightarrow E) \Psi(\mathbf{x}, \hat{\Omega}', E') \, d\Omega' \, dE' \\ + \frac{\chi(\mathbf{x}, E)}{4\pi} \int_0^\infty \int_{4\pi} \nu_f \Sigma_f(\mathbf{x}, E') \Psi(\mathbf{x}, \hat{\Omega}', E') \, d\Omega' \, dE', \\ \mathbf{x} \in V, \quad \hat{\Omega} \in 4\pi, \quad 0 < E < \infty, \end{aligned} \quad (2.25a)$$

$$\Psi(\mathbf{x}, \hat{\Omega}, E) = 0, \quad \mathbf{x} \in \partial V, \quad \hat{\Omega} \cdot \hat{n} < 0, \quad 0 < E < \infty. \quad (2.25b)$$

As with the k -eigenvalue problems, the state of the reactor can be described by the eigenvalue such that the system is subcritical when $\alpha < 0$, critical when $\alpha = 1$ and supercritical when $\alpha > 0$. Additionally, when both the k -eigenvalue or α -eigenvalue problems are critical, their eigenfunctions yield the true neutron flux of the critical system within a multiplicative constant. While the k and α -eigenvalues are similar in these regards, they are not the same. When the k -eigenvalue problem is not critical, its eigenfunction has no physical

interpretation. The α -eigenvalue on the other hand has a physical interpretation as the time-independent shape of the neutron flux independent of time.

When the system is critical or supercritical, the bracketed term on the left-hand side of Eq. (2.25a), $[\Sigma_t(\mathbf{x}, E) + \frac{\alpha}{v}] \geq 0$ for all energies. The $\frac{\alpha}{v}$ term can be mathematically treated as an additional absorption reaction and it is relatively easy to develop algorithms for solving Eq. (2.25a). In the subcritical case, the $\frac{\alpha}{v}$ term is no longer positive, making the calculation of α and Ψ more problematic because of the negative macroscopic cross section that arises if $\frac{\alpha}{v} > \Sigma_t(\mathbf{x}, E)$.

2.3 Adjoint Neutron Transport Equation

In Sec. 2.2, the formulation of the neutron transport equation was focused on the *forward* version of the transport process. The forward transport process, in a simplistic explanation in words, follows a particle that originates at some source and is transported through the host media until it interacts within some specified region of interest, such as a radiation detector, and contributes to some specified quantity of interest. It is also useful to define the transport process in reverse, starting from some detector region of interest and following the interaction and streaming events backwards towards the originating source. This formulation is used to define the *adjoint* neutron transport equation. The adjoint solution $\psi^\dagger$ is often thought of as an importance weighting function for the forward solution, i.e., it describes how important a particular region of phase space is for a neutron to interact in a particular region of interest. The adjoint solution is a key component for many neutron transport numerical solution techniques, including perturbation theory [49], [50], Monte Carlo variance reduction [51]–[54], and point reactor kinetics [3], [49], [55].

The mathematical relationship used to define the relationship between a function f and its an adjoint function $f^\dagger$ is that for that some linear operator $\mathbf{L}$ the following inner products are satisfied¹

$$\langle f^\dagger \mathbf{L} f \rangle = \langle f \mathbf{L}^\dagger f^\dagger \rangle . \quad (2.26)$$

By convention, the bracket notation $\langle \cdot \rangle$ signifies integration over all the independent variables of the domain. For neutron transport this would normally mean integration over all seven dimensions of the particle phase space. For this dissertation, unless otherwise stated, the bracket notation neglects integration over the temporal domain and instead indicates integration over only the spatial, directional and energy dimensions of phase space

$$\langle f \rangle = \int_0^\infty \int_V \int_{4\pi} f \, d\Omega \, dV \, dE . \quad (2.27)$$

¹This relationship is only exact if the boundary source is zero, otherwise a surface integral term appears. To reduce the complexity of the derivation, the boundary source is assumed to be zero.

Let Eq. (2.16a) be written in operator notation with the $\frac{1}{4\pi}$ factor in front of the external source Q not shown for convenience

$$\frac{1}{v} \frac{\partial \psi}{\partial t} (\mathbf{x}, \hat{\Omega}, E, t) + \mathbf{M} \psi (\mathbf{x}, \hat{\Omega}, E, t) = \mathbf{F} \psi (\mathbf{x}, \hat{\Omega}, E, t) + Q (\mathbf{x}, \hat{\Omega}, E, t), \quad (2.28)$$

where $\mathbf{M}$ is the migration operator accounting for the streaming, collision, and scattering terms, and $\mathbf{F}$ is the fission operator and consists of both the prompt and delayed fission terms. These operators are defined as

$$\begin{aligned} \mathbf{M} \psi (\mathbf{x}, \hat{\Omega}, E, t) = & \nabla \cdot \hat{\Omega} \psi (\mathbf{x}, \hat{\Omega}, E, t) + \Sigma_t (\mathbf{x}, E, t) \psi (\mathbf{x}, \hat{\Omega}, E, t) - \\ & \int_0^\infty \int_{4\pi} \Sigma_s (\mathbf{x}, \hat{\Omega} \cdot \hat{\Omega}', E' \rightarrow E, t) \psi (\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE', \end{aligned} \quad (2.29)$$

$$\begin{aligned} \mathbf{F} \psi (\mathbf{x}, \hat{\Omega}, E, t) = & \frac{1}{4\pi} \chi_p (\mathbf{x}, E, t) \int_0^\infty \int_{4\pi} [1 - \beta (\mathbf{x}, E', t)] \nu \Sigma_f (\mathbf{x}, E', t) \psi (\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE' + \\ & \frac{1}{4\pi} \sum_{j=1}^J \chi_{d,j} (\mathbf{x}, E, t) \lambda_j (\mathbf{x}, E, t) C_j (\mathbf{x}, t). \end{aligned} \quad (2.30)$$

The corresponding adjoint transport equation can then be written in operator notation as

$$-\frac{1}{v} \frac{\partial \psi^\dagger}{\partial t} (\mathbf{x}, \hat{\Omega}, E, t) + \mathbf{M}^\dagger \psi^\dagger (\mathbf{x}, \hat{\Omega}, E, t) = \mathbf{F}^\dagger \psi^\dagger (\mathbf{x}, \hat{\Omega}, E, t) + Q^\dagger (\mathbf{x}, E, t), \quad (2.31)$$

where the adjoint migration and fission operators are defined as

$$\begin{aligned} \mathbf{M}^\dagger \psi^\dagger (\mathbf{x}, \hat{\Omega}, E, t) = & \nabla \cdot \hat{\Omega} \psi^\dagger (\mathbf{x}, \hat{\Omega}, E, t) + \Sigma_t (\mathbf{x}, E, t) \psi^\dagger (\mathbf{x}, \hat{\Omega}, E, t) - \\ & \int_0^\infty \int_{4\pi} \Sigma_s (\mathbf{x}, \hat{\Omega}' \cdot \hat{\Omega}, E \rightarrow E', t) \psi^\dagger (\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE', \end{aligned} \quad (2.32)$$

$$\begin{aligned} \mathbf{F}^\dagger \psi^\dagger (\mathbf{x}, \hat{\Omega}, E, t) = & \frac{1}{4\pi} [1 - \beta (\mathbf{x}, E, t)] \nu \Sigma_f (\mathbf{x}, E, t) \int_0^\infty \int_{4\pi} \chi_p (\mathbf{x}, E', t) \psi^\dagger (\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE' + \\ & \frac{1}{4\pi} \sum_{j=1}^J \beta_j (\mathbf{x}, E, t) \nu \Sigma_f (\mathbf{x}, E, t) \lambda_j (\mathbf{x}, E, t) C_j (\mathbf{x}, t). \end{aligned} \quad (2.33)$$

Recall that the adjoint transport equation can be physically interpreted as reversing the forward transport process. Translated into math, a few modifications have been made to individual terms. First, a negative sign has been placed in front of the time derivative and streaming terms, signifying that both time and particle streaming are happening in reverse. Second, the scattering integral now describes neutrons with incident energy E and incident direction $\hat{\Omega}$ scattering into $d\Omega' dE'$ about $(\hat{\Omega}', E')$. Third, in the fission terms the energy integration is switched such that the energy spectra χ_p and $\chi_{d,j}$ are replaced with $[1 - \beta] \nu \Sigma_f$ and $\beta \nu \Sigma_f$, respectively. Finally, in order for Eq. (2.31) to have a non-zero solution, an adjoint source $Q^\dagger$ must be

specified. There are several choices for the adjoint source and its selection depends on the problem-specific quantity of interest that has been defined. For example, if the quantity of interest in the forward calculation has been defined as a reaction rate R in a given volume of interest V_D at time t

$$R(t) = \int_0^\infty \int_{V_d} \int_{4\pi} g(\mathbf{x}, \hat{\Omega}, E, t) \psi(\mathbf{x}, \hat{\Omega}, E, t) d\Omega dV dE, \quad (2.34)$$

then the corresponding adjoint source would be

$$Q^\dagger(\mathbf{x}, \hat{\Omega}, E, t) \equiv g(\mathbf{x}, \hat{\Omega}, E, t). \quad (2.35)$$

By defining the adjoint source as in Eq. (2.35), the adjoint flux represents the probability that a (forward) neutron born at $(\mathbf{x}, \hat{\Omega}, E, t)$ will experience some interaction defined by detector response function $g(\mathbf{x}, \hat{\Omega}, E, t)$ in V_D at time t . The forward and adjoint sources are then related by the reaction rate

$$R(t) = \langle \psi^\dagger Q \rangle = \langle \psi Q^\dagger \rangle. \quad (2.36)$$

In order to solve the forward transport equation, both boundary and initial conditions must be specified. Analogously, for the adjoint transport equation to be solved, boundary and terminating conditions must be specified. Again, the boundary condition is assumed to be zero here. There are many choices for the adjoint terminating condition, but one that is particularly useful for understanding the physical interpretation of the adjoint flux as an iterated fission probability is defined so that $\psi^\dagger$ is unity at terminating time $t = T$ so that the neutron population can be uniformly counted by integrating over all of phase space. The iterated fission probability interpretation of the adjoint flux is particularly useful for calculating adjoint-weighted point kinetics parameters with the Monte Carlo method [56].²

One important feature of the adjoint flux is that when it is used to weight the forward neutron density, the resulting inner product is invariant with time. This is shown by multiplying Eq. (2.28) by $\psi^\dagger$, multiplying

²The Transient Multi-Level (TML) method described in Ch. 3 has use for such adjoint-weighted Monte Carlo (MC) tallies, but their implementation was beyond the scope of this dissertation work.

Eq. (2.31) by ψ , integrating both equations over phase space and subtracting the two resulting equations

$$\left(\left\langle \psi^\dagger \frac{1}{v} \frac{\partial \psi}{\partial t} \right\rangle + \left\langle \psi \frac{1}{v} \frac{\partial \psi^\dagger}{\partial t} \right\rangle \right) - \left(\langle \psi^\dagger \mathbf{M} \psi \rangle - \underbrace{\langle \psi \mathbf{M}^\dagger \psi^\dagger \rangle}_{\langle \psi^\dagger \mathbf{M} \psi \rangle} \right) = \left(\langle \psi^\dagger \mathbf{F} \psi \rangle - \underbrace{\langle \psi \mathbf{F}^\dagger \psi^\dagger \rangle}_{\langle \psi^\dagger \mathbf{F} \psi \rangle} \right) + \left(\langle \psi^\dagger Q \rangle - \underbrace{\langle \psi Q^\dagger \rangle}_{\langle \psi^\dagger Q \rangle} \right). \quad (2.37)$$

In Eq. (2.37), the migration, fission, and external source terms cancel after applying the adjoint identity defined in Eq. (2.26). The time derivative terms can be rewritten using the chain rule to arrive at

$$\frac{\partial}{\partial t} \left\langle \psi^\dagger \frac{1}{v} \psi \right\rangle = 0. \quad (2.38)$$

Thus, the adjoint-weighted neutron density integrated over phase space is said to be invariant with time. The specific numerical techniques used in this dissertation work require the solution to the eigenvalue formulation of the adjoint neutron transport equation, which neglects time dependence and zeros out the adjoint source term and boundary conditions

$$\mathbf{M}^\dagger \psi^\dagger(\mathbf{x}, \hat{\Omega}, E) = \frac{1}{k^\dagger} \mathbf{F}^\dagger \psi^\dagger(\mathbf{x}, \hat{\Omega}, E). \quad (2.39)$$

The adjoint eigenfunction is now interpreted as the importance function of a neutron contributing to the eigenvalue. A neutron that thermalizes and is subsequently absorbed by a fissile isotope is very important to the reactor because it is likely to cause fission and contribute to the eigenvalue of the system by continuing the chain reaction. Conversely, a neutron that leaks from the reactor is not important because it can no longer contribute to fission chain reactions. Likewise, a neutron that is captured by some non-fissile isotope is not important for the same reason.

2.4 Exact Point Kinetics Equations

Henry[3] first derived the EPKE in 1958, and their use (or approximations to them) has been the fundamental basis for the vast majority of reactor kinetics and dynamics calculations since. The qualifier, “exact”, refers to Eqs. (2.55) being derived directly from Eqs. (2.16) with full phase-space dependence and without relying on some fundamental approximation such as separability of variables. The derivation of the EPKEs begins by factorizing the neutron flux into the product of two functions: an amplitude function that is dependent

exclusively on time $p(t)$, and a shape function $\Psi(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t)$ with full phase-space dependence

$$\psi(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) = p(t) \Psi(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) . \quad (2.40)$$

Note that Eq. (2.40) is a factorization and not a separation of variables. As such, it is not an approximation, but rather a decomposition of the angular flux from one function into two functions. As a consequence, a second equation must be introduced to make the factorization unique. This second equation can also be used to shift most of the time dependence to the amplitude function by holding the integral over the spatial, directional, and energy phase space constant over time to constrain the time variation of the magnitude of the shape function. Additionally, it is convenient to select a weighting function for the angular flux that minimizes error in the reactivity when integrated over phase space. The logical choice for this weighting function, otherwise known as the importance function, is the adjoint fundamental mode eigenfunction $\psi_0^\dagger$. Weighting the partial time derivative of the neutron density by $\psi_0^\dagger$, factorizing the flux as in Eq. (2.40), integrating over $\mathbf{x} \in V$, $\hat{\boldsymbol{\Omega}} \in 4\pi$, $0 \leq E < \infty$, and applying the chain rule gives

$$\left\langle \psi_0^\dagger \frac{1}{v} \frac{\partial \psi}{\partial t} \right\rangle = \frac{dp(t)}{dt} \left\langle \psi_0^\dagger \frac{1}{v} \Psi \right\rangle + p(t) \frac{d}{dt} \left\langle \psi_0^\dagger \frac{1}{v} \Psi \right\rangle . \quad (2.41)$$

Finally, applying the condition that the inner product of the shape function is held constant in time gives

$$\left\langle \psi_0^\dagger \frac{1}{v} \Psi \right\rangle = \text{constant} = K_0 , \quad (2.42)$$

and it follows then that the second time derivative on the right-hand side of Eq. (2.41) is zero. With the amplitude, shape and constraint functions now defined, they can be substituted into the appropriate transport equation for an initially-critical reactor to define the EPKEs.

2.4.1 Derivation for an Initially Critical Reactor

Let Eqs. (2.16a) and (2.16b) be represented in operator notation with the internal source Q omitted

$$\frac{1}{v} \frac{\partial \psi}{\partial t}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) = (\mathbf{F}_p - \mathbf{M}) \psi(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) + S_d(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) , \quad (2.43a)$$

$$\frac{\partial C_j}{\partial t}(\mathbf{x}, t) = \frac{1}{\chi_{d,j}(\mathbf{x}, E, t)} \mathbf{F}_{d,j} \psi(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) - \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t) . \quad (2.43b)$$

The migration operator $\mathbf{M}$ was defined previously in Eq. (2.29). The prompt fission operator $\mathbf{F}_p$, delayed fission operator $\mathbf{F}_d$, and delayed neutron source S_d are written as

$$\mathbf{F}_p \psi(\mathbf{x}, \hat{\Omega}, E, t) = \frac{1}{4\pi} \chi_p(\mathbf{x}, E, t) \int_0^\infty \int_{4\pi} [1 - \beta(\mathbf{x}, E', t)] \nu \Sigma_f(\mathbf{x}, E', t) \psi(\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE', \quad (2.44)$$

$$\mathbf{F}_{d,j} \psi(\mathbf{x}, \hat{\Omega}, E, t) = \chi_{d,j}(\mathbf{x}, E, t) \int_0^\infty \int_{4\pi} \beta_j(\mathbf{x}, E', t) \nu \Sigma_f(\mathbf{x}, E', t) \psi(\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE', \quad (2.45)$$

$$\mathbf{F}_d \psi(\mathbf{x}, \hat{\Omega}, E, t) = \sum_{j=1}^J \mathbf{F}_{d,j} \psi(\mathbf{x}, \hat{\Omega}, E, t), \quad (2.46)$$

$$S_{d,j}(\mathbf{x}, \hat{\Omega}, E, t) = \frac{1}{4\pi} \chi_{d,j}(\mathbf{x}, E, t) \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t), \quad (2.47)$$

$$S_d(\mathbf{x}, \hat{\Omega}, E, t) = \sum_{j=1}^J S_{d,j}(\mathbf{x}, \hat{\Omega}, E, t). \quad (2.48)$$

For the initial and boundary conditions, as the title of this subsection suggests, assume that the reactor is critical and in steady state for all times $t \leq 0$. The boundary condition is then zero. The initial condition at time $t = 0$ is

$$\begin{aligned} 0 &= (\mathbf{F}_{p0} - \mathbf{M}_0) \psi_0(\mathbf{x}, \hat{\Omega}, E) + \mathbf{F}_{d0} \psi_0(\mathbf{x}, \hat{\Omega}, E) \\ &= (\mathbf{F}_0 - \mathbf{M}_0) \psi_0(\mathbf{x}, \hat{\Omega}, E), \end{aligned} \quad (2.49)$$

where the fission operator is defined as

$$\mathbf{F} \psi(\mathbf{x}, \hat{\Omega}, E, t) = \mathbf{F}_p \psi(\mathbf{x}, \hat{\Omega}, E, t) + \mathbf{F}_d \psi(\mathbf{x}, \hat{\Omega}, E, t), \quad (2.50)$$

and the subscript 0 denotes $t = 0$. Note that this definition of $\mathbf{F}$ is not the same as that defined in Eq. (2.30). In order to introduce the concept of reactivity, a measure of the off-criticality is introduced by adding and subtracting the operator $\mathbf{F}_d$ within the parentheses of Eq. (2.43a)

$$\begin{aligned} \frac{1}{v} \frac{\partial \psi}{\partial t}(\mathbf{x}, \hat{\Omega}, E, t) &= (\mathbf{F}_p + \mathbf{F}_d - \mathbf{M} - \mathbf{F}_d) \psi(\mathbf{x}, \hat{\Omega}, E, t) + S_d(\mathbf{x}, \hat{\Omega}, E, t) \\ &= (\mathbf{F} - \mathbf{M} - \mathbf{F}_d) \psi(\mathbf{x}, \hat{\Omega}, E, t) + S_d(\mathbf{x}, \hat{\Omega}, E, t) \\ &= (\mathbf{F} - \mathbf{M}) \psi(\mathbf{x}, \hat{\Omega}, E, t) - \mathbf{F}_d \psi(\mathbf{x}, \hat{\Omega}, E, t) + S_d(\mathbf{x}, \hat{\Omega}, E, t). \end{aligned} \quad (2.51)$$

The quantity $\mathbf{F}_d \psi$ is not the actual delayed neutron source at time t . Instead, it is physically interpreted as the *quasi-steady-state delayed neutron source*, or the source of delayed neutrons that would be produced in a steady-state reactor with fission cross sections and neutron flux as they exist at time t . The accumulation of

delayed neutrons from all previous times into a delayed neutron source term at time t is instead described by S_d . The derivation continues by weighting Eq. (2.51) by $\psi_0^\dagger$, weighting Eq. (2.43b) by $\psi_0^\dagger \chi_{d,j}$, and integrating over phase space

$$\left\langle \psi_0^\dagger \frac{1}{v} \frac{\partial \psi}{\partial t} \right\rangle = \left\langle \psi_0^\dagger (\mathbf{F} - \mathbf{M}) \psi \right\rangle - \left\langle \psi_0^\dagger \mathbf{F}_d \psi \right\rangle + \left\langle \psi_0^\dagger S_d (\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) \right\rangle, \quad (2.52a)$$

$$\frac{d}{dt} \left\langle \psi_0^\dagger \chi_{d,j} C_j \right\rangle = \left\langle \psi_0^\dagger \mathbf{F}_{d,j} \psi \right\rangle - \left\langle \psi_0^\dagger \chi_{d,j} \lambda_j C_j \right\rangle. \quad (2.52b)$$

where the initial adjoint flux is the solution of the adjoint problem to Eq. (2.49)

$$(\mathbf{F}_0^\dagger - \mathbf{M}_0^\dagger) \psi_0^\dagger (\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) = 0. \quad (2.52c)$$

The flux factorization from Eq. (2.40) is now introduced into Eq. (2.52a) and divided by $\left\langle \psi_0^\dagger \mathbf{F} \Psi \right\rangle$ to arrive at

$$\frac{dp(t)}{dt} \frac{\left\langle \psi_0^\dagger \frac{1}{v} \Psi \right\rangle}{\left\langle \psi_0^\dagger \mathbf{F} \Psi \right\rangle} = \frac{\left\langle \psi_0^\dagger (\mathbf{F} - \mathbf{M}) \Psi \right\rangle}{\left\langle \psi_0^\dagger \mathbf{F} \Psi \right\rangle} p(t) - \frac{\left\langle \psi_0^\dagger \mathbf{F}_d \Psi \right\rangle}{\left\langle \psi_0^\dagger \mathbf{F} \Psi \right\rangle} p(t) + \frac{\left\langle \psi_0^\dagger S_d (\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) \right\rangle}{\left\langle \psi_0^\dagger \mathbf{F} \Psi \right\rangle}. \quad (2.53a)$$

The corresponding precursor equations are derived similarly introducing Eq. (2.40) into Eq. (2.52b) and dividing by $\left\langle \psi_0^\dagger \mathbf{F}_0 \Psi_0 \right\rangle$

$$\frac{d}{dt} \frac{\left\langle \psi_0^\dagger \chi_{d,j} C_j \right\rangle}{\left\langle \psi_0^\dagger \mathbf{F}_0 \Psi_0 \right\rangle} = \frac{\left\langle \psi_0^\dagger \mathbf{F}_{d,j} \Psi \right\rangle}{\left\langle \psi_0^\dagger \mathbf{F}_0 \Psi_0 \right\rangle} p(t) - \frac{\left\langle \psi_0^\dagger \chi_{d,j} \lambda_j C_j \right\rangle}{\left\langle \psi_0^\dagger \mathbf{F}_0 \Psi_0 \right\rangle}. \quad (2.53b)$$

Let the exact point kinetics parameters be defined as

$$\Lambda(t) = \frac{\langle \psi_0^\dagger \frac{1}{v} \Psi \rangle}{\langle \psi_0^\dagger \mathbf{F} \Psi \rangle} = \frac{K_0}{\langle \psi_0^\dagger \mathbf{F} \Psi \rangle}, \quad (2.54a)$$

$$\Lambda_0 = \frac{K_0}{\langle \psi_0^\dagger \mathbf{F}_0 \Psi_0 \rangle}, \quad (2.54b)$$

$$\rho(t) = \frac{\langle \psi_0^\dagger (\mathbf{F} - \mathbf{M}) \Psi \rangle}{\langle \psi_0^\dagger \mathbf{F} \Psi \rangle}, \quad (2.54c)$$

$$\beta^{\text{eff}}(t) = \frac{\langle \psi_0^\dagger \mathbf{F}_d \Psi \rangle}{\langle \psi_0^\dagger \mathbf{F} \Psi \rangle} = \sum_j \beta_j^{\text{eff}}(t), \quad (2.54d)$$

$$\beta_j^{\text{eff}}(t) = \frac{\langle \psi_0^\dagger \mathbf{F}_{d,j} \Psi \rangle}{\langle \psi_0^\dagger \mathbf{F} \Psi \rangle}, \quad (2.54e)$$

$$\lambda_j(t) = \frac{\langle \psi_0^\dagger \lambda_j \rangle}{\langle \psi_0^\dagger \mathbf{F} \Psi \rangle}, \quad (2.54f)$$

$$\zeta_j(t) = \frac{\langle \psi_0^\dagger \chi_{d,j} C_j \rangle}{\langle \psi_0^\dagger \mathbf{F}_0 \Psi_0 \rangle}. \quad (2.54g)$$

In Eqs. (2.54), $\Lambda(t)$ is the adjoint-weighted, or “effective”, neutron generation time. In simpler words, Λ describes the average time it takes for a neutron to replace itself through fission. The term $\rho(t)$ is the “dynamic” reactivity with time dependence. Since the derivation of Eq. (2.54c) assumed that the initial state of the reactor was critical, it is equivalent to the definition of the “ k -eigenvalue”, “static”, or “steady-state” reactivity in Eq. (2.23) at time $t = 0$. The $\beta_j^{\text{eff}}(t)$, $\lambda_j(t)$, and $\zeta_j(t)$ terms are the effective delayed neutron fractions, decay constants, and densities, respectively, for precursor group j . Substituting Eqs. (2.54) into Eqs. (2.53) gives the form of the EPKEs used in this dissertation work

$$\frac{dp(t)}{dt} = \frac{\rho(t) - \beta(t)}{\Lambda(t)} + \frac{1}{\Lambda_0} \sum_j \lambda_j(t) \zeta_j(t), \quad (2.55a)$$

$$\frac{d\zeta_j(t)}{dt} = \frac{\Lambda_0}{\Lambda(t)} \beta_j^{\text{eff}}(t) p(t) - \lambda_j(t) \zeta_j(t). \quad (2.55b)$$

The same initial condition from Eq. (2.49) is now expressed as

$$\zeta_j(0) = \frac{\beta_j^{\text{eff}}(0)}{\lambda_j} p(0). \quad (2.55c)$$

The EPKEs are very sensitive to the accuracy of the dynamic reactivity $\rho(t)$, which is highly dependent on

the accuracy of the flux and adjoint flux at the initial state. As these calculations can be very computationally expensive, it is common to see approximations introduced to reformulate the EPKEs. One class of these approximations, *quasi-static methods* is discussed further in Ch. 3.

2.4.2 Finite Differencing for the Exact Point Kinetics Equations

Equations (2.55a) and (2.55b) are a coupled system of linear (if reactivity feedback is neglected) or non-linear (if reactivity feedback is included) Ordinary Differential Equations (ODEs) and can be rewritten as the matrix system

$$\frac{d}{dt} \begin{bmatrix} p(t) \\ \zeta_1(t) \\ \zeta_2(t) \\ \vdots \\ \zeta_J(t) \end{bmatrix} = \begin{bmatrix} \frac{\rho(t) - \beta^{\text{eff}}(t)}{\Lambda(t)} & \frac{\lambda_1(t)}{\Lambda(0)} & \frac{\lambda_2(t)}{\Lambda(0)} & \cdots & \frac{\lambda_J(t)}{\Lambda(0)} \\ \frac{\Lambda(0)}{\Lambda(t)} \beta_1(t) & -\lambda_1(t) & & & \\ \frac{\Lambda(0)}{\Lambda(t)} \beta_2(t) & & -\lambda_2(t) & & \\ \vdots & & & \ddots & \\ \frac{\Lambda(0)}{\Lambda(t)} \beta_J(t) & & & & -\lambda_J(t) \end{bmatrix} \begin{bmatrix} p(t) \\ \zeta_1(t) \\ \zeta_2(t) \\ \vdots \\ \zeta_J(t) \end{bmatrix}. \quad (2.56)$$

Solving Eq. (2.56) is relatively straightforward with well-established numerical methods, such as the Newton-Raphson method. However, in most practical applications it is ideal to avoid EPKE matrix inversions as they can be costly due to the numerical stiffness inherent to prompt and delayed neutron timescales or nonlinearities associated with reactor feedback mechanisms. Numerous techniques have been proposed to avoid these issues that make use of Taylor series expansions [57], Runge-Kutta methods [58], Backward Differentiation Formula methods [59], and Padé approximations [60]. Spectral Deferred Correction has also recently been applied to solve the EPKEs [61] and is being investigated for parallel-in-time applications. The following differencing of the EPKEs relies on an exponential transform [62] of the flux amplitude function $p(t)$ and its corresponding derivative

$$p(t) = e^{\alpha(t)t} \hat{p}(t), \quad (2.57)$$

$$\frac{dp(t)}{dt} = e^{\alpha(t)t} \frac{d\hat{p}(t)}{dt} + \alpha(t) e^{\alpha(t)t} \hat{p}(t). \quad (2.58)$$

where $\hat{p}(t)$ is ideally a small modulation on the assumed behavior of $e^{\alpha(t)t}$ and $\alpha(t)$ is a time-dependent frequency. Substituting Eq. (2.58) into Eq. (2.55a) gives

$$\frac{d\hat{p}(t)}{dt} = \left(\frac{\rho(t) - \beta^{\text{eff}}(t)}{\Lambda(t)} - \alpha(t) \right) \hat{p}(t) + \frac{e^{-\alpha(t)t}}{\Lambda(0)} \sum_j \lambda_j \zeta_j(t). \quad (2.59)$$

Reformulating Eq. (2.56) with Eq. (2.59) gives a system of equations that is less susceptible to numerical

truncation error (though not necessarily more stable) [62]. Finally, finite differencing is applied to Eq. (2.59) with θ -weighting [63]

$$e^{-\alpha^n \Delta t^n} \hat{p}^n - \hat{p}^{n-1} = \theta \Delta t^n e^{-\alpha^n \Delta t^n} \left[\left(\frac{\rho^n - \beta^{\text{eff},n}}{\Lambda^n} - \alpha^n \right) \hat{p}^n + \frac{1}{\Lambda_0} \sum_j \lambda_j^n \zeta_j^n \right] \\ + (1 - \theta) \Delta t^{n-1} \left[\left(\frac{\rho^{n-1} - \beta^{\text{eff},n-1}}{\Lambda^{n-1}} - \alpha^{n-1} \right) \hat{p}^{n-1} + \frac{1}{\Lambda_0} \sum_j \lambda_j^{n-1} \zeta_j^{n-1} \right], \quad (2.60)$$

where n is discrete time index and $0 \leq \theta \leq 1$ can be selected to yield more familiar temporal differencing schemes:

- $\theta = 0 \rightarrow$ forward Euler (explicit) with 1st order accuracy,
- $\theta = 1 \rightarrow$ backward Euler (implicit) with 1st order accuracy,
- $\theta = 0.5 \rightarrow$ improved Euler with 2nd order accuracy.

2.4.2.1 Integrating the Precursor Equations

The precursor equations can be integrated over the time step $t^{n-1} \leq t \leq t^n$ after approximating delayed neutron production as having a quadratic time dependence. This integration eliminates the ζ_j ODEs and avoids having to invert the point kinetics matrix in Eq. (2.56). To integrate the precursor equations, first multiply Eq. (2.55b) by the integrating factor $e^{\lambda_j t}$ and recognize that the left-hand side can be rearranged to resemble the chain rule for derivatives

$$\frac{d\zeta_j(t)}{dt} e^{\lambda_j t} + \lambda_j \zeta_j(t) e^{\lambda_j t} = \frac{\Lambda_0}{\Lambda(t)} \beta_j^{\text{eff}}(t) p(t) e^{\lambda_j t} \\ \frac{d}{dt} (\zeta_j(t) e^{\lambda_j t}) = \frac{\Lambda_0}{\Lambda(t)} \beta_j^{\text{eff}}(t) p(t) e^{\lambda_j t}. \quad (2.61)$$

Let the right-hand side of Eq. (2.61) be condensed by defining

$$G(t) \equiv \frac{\Lambda(0)}{\Lambda(t)} \beta^{\text{eff}}(t) p(t), \quad (2.62)$$

and letting $G(t)$ be approximated with a quadratic Lagrange polynomial interpolation (see Appendix A). The precursor equations can then be integrated using Simpson's rule with the dummy variable $\tilde{t} = t - t^{n-1}$.

This changes the integration interval from $t^{n-1} \leq t \leq t^n$ to $0 < \tilde{t} < \Delta t^n$ to arrive at

$$\begin{aligned} E(\lambda_j^n) \zeta_j^n - \zeta_j^{n-1} = & G^n \frac{\kappa_2(\lambda_j^n) + \gamma \Delta t^n \kappa_1(\lambda_j^n)}{(1+\gamma)(\Delta t^n)^2} + G^{n-1} \left[\kappa_0(\lambda_j^n) - \frac{\kappa_2(\lambda_j^n) + (\gamma-1) \Delta t^n \kappa_1(\lambda_j^n)}{\gamma (\Delta t^n)^2} \right] \\ & + G^{n-2} \frac{\kappa_2(\lambda_j^n) - \Delta t^n \kappa_1(\lambda_j^n)}{(1+\gamma) \gamma (\Delta t^n)^2}, \end{aligned} \quad (2.63)$$

where

$$E(\lambda) \equiv e^{\lambda \Delta t^n}, \quad (2.64a)$$

$$\kappa_0(\lambda) \equiv \int_0^{\Delta t^n} e^{\lambda t} dt = \frac{E(\lambda) - 1}{\lambda}, \quad (2.64b)$$

$$\kappa_1(\lambda) \equiv \int_0^{\Delta t^n} t e^{\lambda t} dt = \frac{\Delta t^n E(\lambda) - \kappa_0(\lambda)}{\lambda}, \quad (2.64c)$$

$$\kappa_2(\lambda) \equiv \int_0^{\Delta t^n} t^2 e^{\lambda t} dt = \frac{(\Delta t^n)^2 E(\lambda) - 2\kappa_1(\lambda)}{\lambda}, \quad (2.64d)$$

$$\gamma \equiv \frac{\Delta t^{n-1}}{\Delta t^n}. \quad (2.64e)$$

Equation (2.63) can now be solved for the precursor concentration at time index n

$$\zeta_j^n = \hat{p}^n \Omega_j^n + \hat{\zeta}_j^n, \quad (2.65)$$

where

$$\Omega_j^n \equiv w_j^n \Lambda_0 \frac{\beta_j^{\text{eff},n} \kappa_2(\lambda_j^n) + \gamma \Delta t^n \kappa_1(\lambda_j^n)}{\Lambda^n (1+\gamma)(\Delta t^n)^2}, \quad (2.66a)$$

$$\begin{aligned} \hat{\zeta}_j^n \equiv & w_j^n \zeta_j^{n-1} + w_j^n \Lambda_0 \hat{p}^{n-1} \frac{\beta_j^{\text{eff},n-1}}{\Lambda^{n-1}} \left[\kappa_0(\lambda_j^n) - \frac{\kappa_2(\lambda_j^n) + (\gamma-1) \Delta t^n \kappa_1(\lambda_j^n)}{\gamma (\Delta t^n)^2} \right] \\ & + w_j^n \Lambda_0 \hat{p}^{n-2} \frac{\beta_j^{\text{eff},n-2}}{\Lambda^{n-2}} \frac{\kappa_2(\lambda_j^n) - \Delta t^n \kappa_1(\lambda_j^n)}{(1+\gamma) \gamma (\Delta t^n)^2}, \end{aligned} \quad (2.66b)$$

$$w_j^n \equiv \frac{1}{E(\lambda_j^n)}. \quad (2.66c)$$

The λ_j -weighted sum of Eq. (2.65) over all precursor groups is then

$$\sum_j \lambda_j^n \zeta_j^n = \hat{p}^n \sum_j \lambda_j^n \Omega_j^n + \sum_j \lambda_j \hat{\zeta}_j^{n-1} = \tau^n \hat{p}^n + \hat{S}_d^n, \quad (2.67)$$

where

$$\tau^n \equiv \sum_j \lambda_j^n \Omega_j^n, \quad (2.68a)$$

$$\hat{S}_d^n \equiv \sum_j \lambda_j^n \hat{\zeta}_j^n, \quad (2.68b)$$

$$S_d^{n-1} \equiv \sum_j \lambda_j^{n-1} \zeta_j^{n-1}. \quad (2.68c)$$

Substituting Eq. (2.67) into Eq. (2.60) and solving for $\hat{p}^n$ gives

$$\hat{p}^n = \frac{e^{\alpha^n \Delta t^n} \left\{ \hat{p}^{n-1} + (1 - \theta) \Delta t^n \left[\left(\frac{\rho^{n-1} - \beta^{\text{eff}, n-1}}{\Lambda^{n-1}} - \alpha^n \right) \hat{p}^{n-1} + \frac{S_d^{n-1}}{\Lambda_0} \right] \right\} + \theta \Delta t^n \frac{\hat{S}_d^n}{\Lambda_0}}{1 - \theta \Delta t^n \left(\frac{\rho^n - \beta^{\text{eff}, n}}{\Lambda^n} - \alpha^n + \frac{\tau^n}{\Lambda_0} \right)}. \quad (2.69)$$

After advancing time index n , a new exponential transform frequency is estimated as

$$\alpha^n = \frac{1}{\Delta t^{n-1}} \ln \frac{\hat{p}^{n-1}}{\hat{p}^{n-2}}. \quad (2.70)$$

However, it is necessary to check that the transformed solution is more linear than the original solution. If this is the case, then the transformation parameter α^n is accepted. Otherwise, the solution is re-evaluated without the transformation ($\alpha^n = 0$). This condition is written as

$$\left| \hat{p}^n - e^{\alpha^n \Delta t^n} \hat{p}^{n-1} \right| \leq \left| \hat{p}^n - \hat{p}^{n-1} - (\hat{p}^{n-1} - \hat{p}^{n-2}) / \gamma \right|. \quad (2.71)$$

2.4.2.2 Prompt Reactivity Feedback

Up until this point, the reactivity profile $\rho(t)$ was assumed to be given by a prescribed transient and the feedback mechanisms present in a physical reactor were neglected. Most feedback mechanisms require spatial and energy-dependent solutions of the neutron distribution to be modeled properly, which is clearly beyond the scope of what point kinetics can provide. However, prompt reactivity feedback is dependent almost entirely on energy deposition in the fuel (see Sec. 2.5) during the transient, and several models have been developed to include this feedback in point kinetics formulations. The simplest feedback model for estimating the energy deposited in the fuel during a transient method neglects heat transfer entirely. The method is approximate and referred to as the *linear energy model* or named after its authors, Fuchs [64], Nordheim [65], and Hansen [66]. The linear energy model assumes that the heat flux out of the cylindrical fuel element is proportional to the negative gradient of the temperature distribution. The model additionally assumes that

the temperature distribution of any heat generated in the fuel during the transient event only changes in amplitude, not shape. As a result, during a rapid transient, the magnitude of the temperature distribution is increased without any significant change in the shape of the distribution. Thus, the temperature gradient and heat flux out of the fuel rod are maintained throughout the transient event, i.e., no heat transfer. For a completely adiabatic fuel power increment

$$\rho_d(t) = \gamma_d \int_0^t [H(t') - H(0)] dt', \quad (2.72)$$

where

- $\rho_d(t)$ is the feedback reactivity and the full reactivity profile $\rho(t)$ is written as a sum of the imposed reactivity $\rho_{\text{imp}}(t)$ and the feedback reactivity

$$\rho(t) = \rho_d(t) + \rho_{\text{imp}}(t). \quad (2.73)$$

- $H(t) = f_{\text{fp}} \hat{p}(t)$ is the core full power, which is the product of the core amplitude and full power normalization factor f_{fp} .
- $\gamma_d = \frac{H \cdot (1 \text{ s})}{mc_p} \frac{\partial \rho}{\partial T_f} \leq 0$ is the fuel temperature coefficient representing the ratio of energy absorbed by the fuel in one second to the heat capacity (the product of the mass, m , and specific heat, c_p) of the fuel multiplied by the differential change in the reactivity per change in fuel temperature T_f .

The adiabatic approximation can be improved by treating the heat release out of the fuel with a single time constant $\lambda_H \geq 0$

$$\rho_d(t) = \gamma_d \int_0^t [H(t') - H(0)] e^{-\lambda_H(t-t')} dt'. \quad (2.74)$$

Integrating Eq. (2.74) over a single time step with variable substitution $\tilde{t}$ over the range $0 \leq \tilde{t} \leq \Delta t^n$ to discretize the feedback reactivity gives

$$\begin{aligned} \rho_d^n &= \rho_d^{n-1} + \gamma_d e^{-\lambda_H \Delta t^n} \int_0^{\Delta t^n} [H(t') - H(0)] e^{\lambda_H t'} dt' \\ &= \rho_d^{n-1} + \frac{\gamma_d}{E(\lambda_H)} \left[-H^0 \kappa_0(\lambda_H) + \int_0^{\Delta t^n} H(t') e^{\lambda_H t'} dt' \right]. \end{aligned} \quad (2.75)$$

Using another quadratic approximation for the temporal behavior of $H(t)$ reduces the integral in Eq. (2.75)

to

$$\begin{aligned}
\int_0^{\Delta t^n} H(\tilde{t}) e^{\lambda_H \tilde{t}} d\tilde{t} &\approx H^n \frac{\kappa_2(\lambda_H) + \kappa_1(\lambda_H) \gamma \Delta t^n}{(1 + \gamma) (\Delta t^n)^2} \\
&+ H^{n-1} \left[\kappa_0(\lambda_H) - \frac{\kappa_2(\lambda_H) - (\gamma - 1) \Delta t^n \kappa_1(\lambda_H)}{\gamma (\Delta t^n)^2} \right] \\
&+ H^{n-2} \frac{\kappa_2(\lambda_H) - \kappa_1(\lambda_H) \Delta t^n}{(1 + \gamma) \gamma (\Delta t^n)^2} .
\end{aligned} \tag{2.76}$$

Substituting Eqs. (2.76) and (2.75) into Eq. (2.73) and rearranging for ρ^n gives

$$\rho^n = a_1 \hat{p}^n + b_1 , \tag{2.77}$$

where

$$a_1 = \frac{\gamma f_{\text{fp}}}{E(\lambda_H)} \frac{\kappa_2(\lambda_H) + \kappa_1(\lambda_H) \gamma \Delta t^n}{(1 + \gamma) (\Delta t^n)^2} , \tag{2.78a}$$

$$\begin{aligned}
b_1 = \rho_{\text{imp}}^n + \rho_{\text{d}}^{n-1} + \frac{\gamma f_{\text{fp}}}{E(\lambda_H)} &\left\{ -\hat{p}^0 \kappa_0(\lambda_H) \right. \\
&+ \hat{p}^{n-1} \left[\kappa_0(\lambda_H) - \frac{\kappa_2(\lambda_H) + (\gamma - 1) \Delta t^n \kappa_1(\lambda_H)}{\gamma (\Delta t^n)^2} \right] \\
&\left. + \hat{p}^{n-2} \frac{\kappa_2(\lambda_H) - \kappa_1(\lambda_H) \Delta t^n}{(1 + \gamma) \gamma (\Delta t^n)^2} \right\} .
\end{aligned} \tag{2.78b}$$

Finally, substituting Eq. (2.77) into Eq. (2.69) gives a quadratic (hence the “non-linearity” from the feedback) expression for the the amplitude

$$a (\hat{p}^n)^2 + b \hat{p}^n + c = 0 , \tag{2.79}$$

where

$$a = \frac{\theta \Delta t^n a_1}{\Lambda^n} , \tag{2.80a}$$

$$b = \left[\theta \Delta t^n \left(\frac{b_1 - \beta^{\text{eff},n}}{\Lambda^n} - \alpha^n + \frac{\tau^n}{\Lambda_0} \right) - 1 \right] , \tag{2.80b}$$

$$c = \theta \Delta t^n \frac{1}{\Lambda_0} \hat{S}_d^n + e^{\alpha^n \Delta t^n} \left\{ (1 - \theta) \Delta t^n \left[\left(\frac{\rho^{n-1} - \beta^{\text{eff},n-1}}{\Lambda^{n-1}} - \alpha^n \right) \hat{p}^{n-1} + \frac{S_d^{n-1}}{\Lambda_0} \right] + \hat{p}^{n-1} \right\} . \tag{2.80c}$$

The flux amplitude $\hat{p}^n$ can then be solved using the quadratic formula as

$$\hat{p}^n = \begin{cases} \frac{-b - \sqrt{b^2 - 4ac}}{2a}, & a < 0 \\ \frac{c}{-b}, & a = 0. \end{cases} \quad (2.81)$$

Note that for $a = 0$, Eq. (2.81) is equivalent to Eq. (2.69). Algorithm 2.1 describes the final EPKE solution scheme used. The calculation marches through successive time steps and accumulates the integrated delayed neutron precursor contributions. After the amplitude has been estimated with the exponential transform in line 19, its linearity is checked and re-evaluated without the exponential transform if necessary.

Algorithm 2.1 Finite differencing solution of the EPKEs with prompt reactivity feedback.

```

1:  $\hat{p}^0 \leftarrow 1; \rho^0 \leftarrow \rho_{\text{imp}}^0$ 
2: for all  $j$  do
3:    $\zeta_j^0 \leftarrow \text{Eq. (2.55c)}$ 
4: end for
5: for all  $n > 0$  do
6:   if  $n == 1$  then
7:      $\hat{p}^{n-2} \leftarrow 0; \rho^{n-2} \leftarrow 0;$ 
8:     for all  $j$  do
9:        $\zeta_j^{n-2} \leftarrow 0;$ 
10:    end for
11:  end if
12:  for all  $j$  do
13:     $w_j^n \leftarrow \text{Eq. (2.66c)}; \Omega_j^n \leftarrow \text{Eq. (2.66a)}; \hat{\zeta}_j^n \leftarrow \text{Eq. (2.66b)}$ 
14:  end for
15:   $\tau^n, \hat{S}_d^n, S_d^{n-1} \leftarrow \text{Eqs. (2.68)}$ 
16:   $a_1, b_1 \leftarrow \text{Eqs. (2.78)}$ 
17:   $\alpha^n \leftarrow \text{Eq. (2.70)}$ 
18:   $a, b, c \leftarrow \text{Eqs. (2.80)}$ 
19:   $\hat{p}^n \leftarrow \text{Eq. (2.81)}$ 
20:  if Eq. (2.71) is not True then
21:     $\alpha^n \leftarrow 0$ 
22:     $a, b, c \leftarrow \text{Eqs. (2.80)}$ 
23:     $\hat{p}^n \leftarrow \text{Eq. (2.81)}$ 
24:  end if
25:   $\rho^n \leftarrow \text{Eq. (2.77)}$ 
26:  for all  $j$  do
27:     $\zeta_j^n \leftarrow \text{Eq. (2.65)};$ 
28:  end for
29: end for

```

2.5 Doppler Broadening

There are several feedback mechanisms that result from changes to the moderator temperature, pressure, and phase state and introduce positive or negative reactivity to the reactor system. There are also reactivity

changes introduced as the isotopic concentration changes via nuclear transmutations during reactor operation. While these feedback mechanisms are important and should be considered as avenues of future work, they occur on relatively long timescales and require the neutronics solvers to be integrated with thermal hydraulics or depletion solvers, respectively. Coupling with additional external solvers beyond the various neutronics solvers already being used within the TML was beyond the scope of this dissertation and discussed as future work in Chapter 5. Instead, the analysis done in this dissertation work focuses exclusively on changes in reactivity associated with changes in the fuel temperature.

The rise of the fuel temperature during reactor operation comes from the energy released during fission reactions (~ 200 MeV of recoverable energy per fission). A small amount of the released energy ($\sim 3\%$) appears as the kinetic energy of prompt neutrons, most of which is deposited in the coolant through scattering reactions. Additional energy ($\sim 11\%$) appears in the form of γ rays emitted from neutron radiative capture. Another ~ 21 MeV is attributed to the radioactive decay of unstable fission products in the form of β particles, antineutrinos, or γ rays. Most of this energy though is not recoverable or occurs on “delayed” timescales that are not considered for prompt feedback effects. The majority of the energy released in the fission process (about 83%) appears as the kinetic energy of the fission fragments. These fission fragments emerge as highly ionized atoms and lose most of their energy after traveling about $100\mu\text{m}$ in the fuel. The ion pairs that are produced by these fission products quickly dissipate their energy after a negligibly short time $\ll \Lambda$ within a very small volume. The superposition of 1×10^{15} to 1×10^{20} fission tracks per second (depending on the size of the reactor) with rapid heat transfer over very short distances between tracks then leads to a smooth spatial distribution of the heat production and temperature increases within the reactor core.

The neutron cross sections within the system are dependent on the relative velocity between the neutron and the target nucleus. Since the target nuclei are in continual motion due to their thermal lattice vibrations, incident neutrons appear to have a continuous spread in energy. This continuous spread in energy affects the shape of resonances. Higher temperatures translate to more rapid lattice vibrations that effectively broaden the range of energies that neutrons may be resonantly absorbed within the fuel. The resonances become shorter and wider than when the target nuclei are at rest, although the total area under the resonance remains essentially constant (see Fig. 2.3).

The exact Doppler broadening equation developed by Cullen [67] in 1976 is well-known

$$\sigma_{\text{Doppler}}(y) = \frac{1}{\sqrt{\pi}y^2} \int_0^\infty \sigma_0(x) x^2 \left[e^{-(x-y)^2} - e^{-(x+y)^2} \right] dx, \quad (2.82)$$

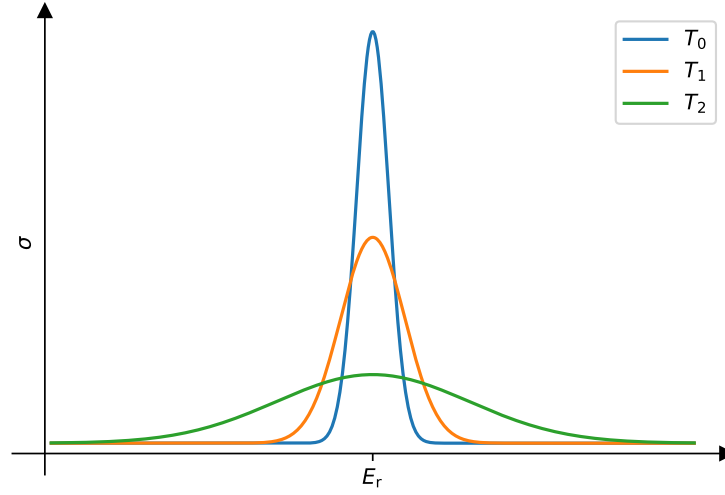


Figure 2.3: Doppler broadening effect on resonance shape as it appears to a neutron where $T_0 < T_1 < T_2$ and E_r is the center energy of the resonance.

where

$$y^2 = \alpha E = \beta v (E)^2, \quad (2.83a)$$

$$x^2 = \alpha E_r = \beta v_r^2 (E, T), \quad (2.83b)$$

$$\alpha = \frac{A}{k (T_{\text{Doppler}} - T_0)}, \quad (2.83c)$$

$$\beta = \frac{M}{2k (T_{\text{Doppler}} - T_0)}. \quad (2.83d)$$

$A = M/m$ is the atomic weight ratio between the mass of the target nucleus M and the mass of the neutron m . E_r is the energy of the resonance, and v_r is the relative velocity between the target nucleus and the neutron. σ_{Doppler} and T_{Doppler} are the cross section and temperature, respectively, of the Doppler broadening state. T_0 is the temperature at the initial state.

2.6 Approximations to the Neutron Transport Equation

Despite the neutron transport equation already being a simplified version of the general Boltzmann transport process, it is still very difficult to solve for many practical problems. As a result, the transport literature has seen dozens of attempts to simplify the neutron transport equation. The nature of this dissertation work is centered around combining multiple solution techniques of the neutron transport equation to find a balance

between computational speed and accuracy. As such, a handful of approximations to the neutron transport equation are used to generate solutions with different physical fidelities.

The subsections that follow discuss these approximations to the extent that is necessary and useful for context in subsequent chapters. They are not intended to be an exhaustive derivation of all of the mainstream approximations to the neutron transport equation. Sections 2.6.1 and 2.6.2.1 are necessary to derive the multigroup CMFD equations in Section 2.8 that makeup a key solver within the TML discussed in Chapter 3. Section 2.6.2.2 is necessary to derive the equations that are used to calculate the adjoint solution used to generate point kinetics parameters used in the TML.

2.6.1 Multigroup Approximation

Many numerical techniques for solving the neutron transport equation require the entire phase space to be discretized, i.e. deterministic methods (see Sec. 2.9). One of the most important and consequential of these discretizations is that of the energy domain. In order to discretize the energy domain, the *multigroup approximation* is introduced. The multigroup approximation assumes that the energy dependence of the angular flux is a separable function in space and energy such that

$$\psi(\mathbf{x}, \hat{\Omega}, E, t) \approx \Psi(\mathbf{x}, E) f(\mathbf{x}, \hat{\Omega}, t). \quad (2.84)$$

In Eq. (2.84), $\Psi(\mathbf{x}, E)$ is a space and energy-dependent neutron spectrum used to weight energy-dependent material properties over energy group g defined by the range the range $E \in [E_g, E_{g-1})$. There are generally G energy groups defined and are arranged by convention that smallest energy bound $E_{\min}$ corresponds to the lower bound of the largest group index $g = G$ and the largest energy bound $E_{\max}$ corresponds to the upper bound of the smallest group index $g = 0$

$$0 \leq E_G = E_{\min} \leq E_{G-1} < \cdots < E_g < E_{g-1} < \cdots < E_1 < E_0 = E_{\max} < \infty. \quad (2.85)$$

In practice, the number of groups and precise group bounds that are specified is application-dependent. Many reactor physics applications use as few as two energy groups to represent “fast” high-energy neutrons and “thermal” low-energy neutrons. Other applications, such as neutron shielding design, require tens to hundreds of energy groups to produce accurate calculations and meet safety design standards. The selection of the energy bounds themselves can sometimes vary from problem to problem, as the evaluations of energy-dependent material properties are dependent on the weighting spectrum $\Psi(\mathbf{x}, E)$. By weighting energy-dependent material properties by $\Psi(\mathbf{x}, E)$ and integrating over the energy range $E \in [E_g, E_{g-1})$ the

resulting multigroup material properties are defined as

$$\frac{1}{v_g} \equiv \frac{\int_{E_g}^{E_{g-1}} \frac{1}{v(E)} \Psi(\mathbf{x}, E) \, dE}{\int_{E_g}^{E_{g-1}} \Psi(\mathbf{x}, E) \, dE} \quad (2.86a)$$

$$\Sigma_{t,g}(\mathbf{x}, t) \equiv \frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\mathbf{x}, E, t) \Psi(\mathbf{x}, E) \, dE}{\int_{E_g}^{E_{g-1}} \Psi(\mathbf{x}, E) \, dE}, \quad (2.86b)$$

$$\nu \Sigma_{f,g}(\mathbf{x}, t) \equiv \frac{\int_{E_g}^{E_{g-1}} \nu \Sigma_f(\mathbf{x}, E, t) \Psi(\mathbf{x}, E) \, dE}{\int_{E_g}^{E_{g-1}} \Psi(\mathbf{x}, E) \, dE}, \quad (2.86c)$$

$$\Sigma_{s,g' \rightarrow g}(\mathbf{x}, \hat{\Omega}' \cdot \hat{\Omega}, t) \equiv \frac{\int_{E_g}^{E_{g-1}} \int_{E_{g'}}^{E_{g'-1}} \Sigma_s(\mathbf{x}, E', \hat{\Omega}' \cdot \hat{\Omega}, t) \Psi(\mathbf{x}, E) \, dE' \, dE}{\int_{E_g}^{E_{g-1}} \Psi(\mathbf{x}, E) \, dE}, \quad (2.86d)$$

$$\Sigma_{s,g}(\mathbf{x}) \equiv \sum_{g'=1}^G \int_{4\pi} \Sigma_{s,g' \rightarrow g}(\mathbf{x}, \hat{\Omega}' \cdot \hat{\Omega}, t) \, d\Omega. \quad (2.86e)$$

The delayed neutron fractions will be handled specially so that they are weighted over the entire energy domain and can be pulled out of the fission integrals. Recall their definitions from Eqs. (2.19)

$$\bar{\beta}(\mathbf{x}, t) \equiv \frac{\int_0^\infty \beta(\mathbf{x}, E', t) \nu \Sigma_f(\mathbf{x}, E', t) \Psi(\mathbf{x}, E') \, dE'}{\int_0^\infty \nu \Sigma_f(\mathbf{x}, E', t) \Psi(\mathbf{x}, E') \, dE'}, \quad (2.87a)$$

$$\bar{\beta}_j(\mathbf{x}, t) \equiv \frac{\int_0^\infty \beta_j(\mathbf{x}, E', t) \nu \Sigma_f(\mathbf{x}, E', t) \Psi(\mathbf{x}, E') \, dE'}{\int_0^\infty \nu \Sigma_f(\mathbf{x}, E', t) \Psi(\mathbf{x}, E') \, dE'}. \quad (2.87b)$$

The flux, internal source, and fission spectra do not require special weighting and are simply integrated over energy group g to define

$$\psi_g(\mathbf{x}, \hat{\Omega}, t) \equiv \int_{E_g}^{E_{g-1}} \psi(\mathbf{x}, \hat{\Omega}, E, t) \, dE, \quad (2.88a)$$

$$Q_g(\mathbf{x}, \hat{\Omega}) \equiv \int_{E_g}^{E_{g-1}} Q(\mathbf{x}, \hat{\Omega}, E) \, dE, \quad (2.88b)$$

$$\chi_g(\mathbf{x}) \equiv \int_{E_g}^{E_{g-1}} \chi(\mathbf{x}, E) \, dE. \quad (2.88c)$$

Integrating Eqs. (2.16a) and (2.16b) over energy group g from E_g to E_{g-1} gives

$$\begin{aligned}
& \frac{\partial}{\partial t} \left[\frac{\int_{E_g}^{E_{g-1}} \frac{1}{v(E)} \psi(\mathbf{x}, \hat{\Omega}, E, t) dE}{\int_{E_g}^{E_{g-1}} \psi(\mathbf{x}, \hat{\Omega}, E, t) dE} \right] \psi_g(\mathbf{x}, \hat{\Omega}, t) + \hat{\Omega} \cdot \nabla \psi_g(\mathbf{x}, \hat{\Omega}, t) \\
& - \left[\frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\mathbf{x}, E, t) \psi(\mathbf{x}, \hat{\Omega}, E, t) dE}{\int_{E_g}^{E_{g-1}} \psi(\mathbf{x}, \hat{\Omega}, E, t) dE} \right] \psi_g(\mathbf{x}, \hat{\Omega}, t) = \\
& \sum_{g'=1}^g \int_{4\pi} \left[\frac{\int_{E_g}^{E_{g-1}} \int_{E_{g'}}^{E_{g'-1}} \Sigma_s(\mathbf{x}, \hat{\Omega} \cdot \hat{\Omega}', E' \rightarrow E, t) \psi(\mathbf{x}, \hat{\Omega}', E', t) dE' dE}{\int_{E_{g'}}^{E_{g'-1}} \psi(\mathbf{x}, \hat{\Omega}', E', t) dE'} \right] \psi_{g'}(\mathbf{x}, \hat{\Omega}', t) d\Omega' \\
& + [1 - \bar{\beta}(\mathbf{x}, t)] \frac{\chi_{p,g}}{4\pi}(\mathbf{x}, t) \sum_{g'=1}^G \int_{4\pi} \left[\frac{\int_{E_{g'}}^{E_{g'-1}} \nu \Sigma_f(\mathbf{x}, E', t) \psi(\mathbf{x}, \hat{\Omega}', E', t) dE'}{\int_{E_{g'}}^{E_{g'-1}} \psi(\mathbf{x}, \hat{\Omega}', E', t) dE'} \right] \psi_{g'}(\mathbf{x}, \hat{\Omega}', t) d\Omega' \\
& + \sum_{j=1}^J \frac{\chi_{d,j,g}}{4\pi} \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t) + \frac{1}{4\pi} Q_g(\mathbf{x}, t), \tag{2.89a}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial C_j}{\partial t}(\mathbf{x}, t) &= \frac{1}{4\pi} \bar{\beta}_j(\mathbf{x}, t) \sum_{g'=1}^G \int_{4\pi} \left[\frac{\int_{E_{g'}}^{E_{g'-1}} \nu_f \Sigma_f(\mathbf{x}, E', t) \psi(\mathbf{x}, \hat{\Omega}', E', t) dE'}{\int_{E_{g'}}^{E_{g'-1}} \psi(\mathbf{x}, \hat{\Omega}', E', t) dE'} \right] \psi_{g'}(\mathbf{x}, \hat{\Omega}', t) d\Omega' \\
& - \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t). \tag{2.89b}
\end{aligned}$$

The angular flux in all of the bracketed terms in Eq. (2.89) is substituted with the approximation defined in Eq. (2.84). These terms can then be replaced by the multigroup material properties defined in Eqs. (2.86) to arrive at the multigroup transport equation

$$\begin{aligned}
& \frac{1}{v_g} \frac{\partial \psi_g(\mathbf{x}, \hat{\Omega}, t)}{\partial t} + \hat{\Omega} \cdot \nabla \psi_g(\mathbf{x}, \hat{\Omega}, t) - \Sigma_{t,g}(\mathbf{x}) \psi_g(\mathbf{x}, \hat{\Omega}, t) = \\
& \sum_{g'=1}^G \int_{4\pi} \Sigma_{s,g' \rightarrow g}(\mathbf{x}, \hat{\Omega}' \cdot \hat{\Omega}, t) \psi_{g'}(\mathbf{x}, \hat{\Omega}', t) d\Omega' \\
& + [1 - \beta] \frac{\chi_{p,g}}{4\pi}(\mathbf{x}, t) \sum_{g'=1}^G \nu \Sigma_{f,g'}(\mathbf{x}, t) \int_{4\pi} \psi_{g'}(\mathbf{x}, \hat{\Omega}', t) d\Omega' \\
& + \sum_{j=1}^J \frac{\chi_{d,j,g}}{4\pi} \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t) + \frac{1}{4\pi} Q_g(\mathbf{x}, t), \tag{2.90a}
\end{aligned}$$

$$\frac{\partial C_j}{\partial t}(\mathbf{x}, t) = \frac{1}{4\pi} \bar{\beta}_j(\mathbf{x}, t) \sum_{g'=1}^G \int_{4\pi} \nu \Sigma_{f,g'} \psi_{g'}(\mathbf{x}, \hat{\Omega}', t) d\Omega' - \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t). \tag{2.90b}$$

Equations (2.90) are defined over the domain $\mathbf{x} \in V$, $\hat{\Omega} \in 4\pi$, $1 \leq g \leq G$, $0 < t$. The boundary and initial

conditions are written as

$$\psi_g(\mathbf{x}, \hat{\boldsymbol{\Omega}}, t) = \psi_g^b(\mathbf{x}, \hat{\boldsymbol{\Omega}}, t), \quad \mathbf{x} \in V, \quad \hat{\boldsymbol{\Omega}} \cdot \hat{\mathbf{n}} < 0, \quad 1 \leq g \leq G, \quad 0 < t, \quad (2.90c)$$

$$\psi_g(\mathbf{x}, \hat{\boldsymbol{\Omega}}, 0) = \psi_g^0(\mathbf{x}, \hat{\boldsymbol{\Omega}}), \quad \mathbf{x} \in V, \quad \hat{\boldsymbol{\Omega}} \in 4\pi, \quad 1 \leq g \leq G, \quad (2.90d)$$

where

$$\psi_g^b(\mathbf{x}, \hat{\boldsymbol{\Omega}}, t) \equiv \int_{E_g}^{E_{g-1}} \psi^b(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) dE, \quad (2.91a)$$

$$\psi_g^0(\mathbf{x}, \hat{\boldsymbol{\Omega}}) \equiv \int_{E_g}^{E_{g-1}} \psi^0(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E) dE. \quad (2.91b)$$

As stated previously, the multigroup approximation is one of the most consequential approximations made in discretizing the neutron transport equation. This is primarily due to the fact that the energy dependencies of neutron interaction cross sections do not behave as histograms. Instead, they have rapid variations spanning a few magnitudes of order as shown in Fig. 2.4. For an accurate representation of the structure of these cross sections, thousands of very narrow energy groups would be necessary.

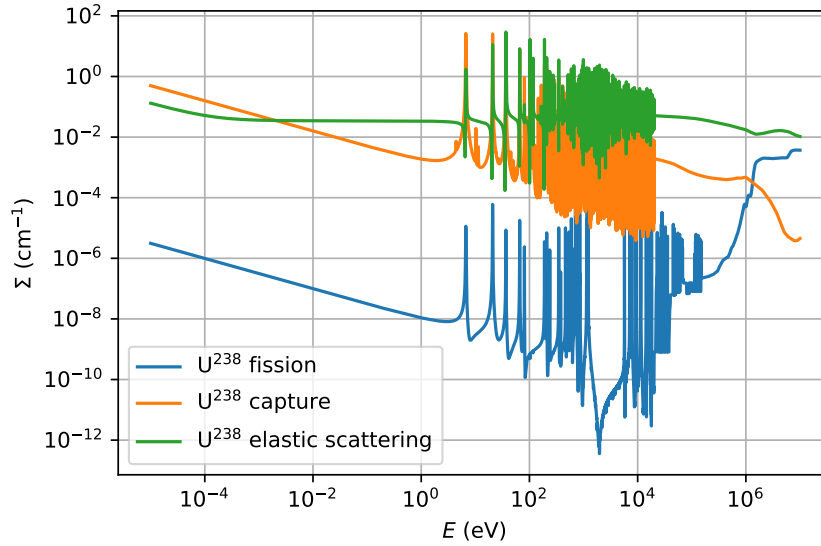


Figure 2.4: The rapidly varying energy dependence of the fission, capture and elastic scattering interactions from U^{238} .

2.6.2 Angular Approximations

When deriving angular approximations to Eqs. (2.16), it is sometimes helpful to define two special angular moments of the angular flux

$$\phi(\mathbf{x}, E, t) = \int_{4\pi} \psi(\mathbf{x}, \hat{\Omega}, E, t) \, d\Omega, \quad (2.92a)$$

$$\mathbf{J}(\mathbf{x}, E, t) = \int_{4\pi} \hat{\Omega} \psi(\mathbf{x}, \hat{\Omega}, E, t) \, d\Omega, \quad (2.92b)$$

where $\phi(\mathbf{x}, E, t)$ is the *scalar flux*, a scalar quantity, and $\mathbf{J}(\mathbf{x}, E, t)$ is the *neutron current vector*, a vector quantity. The exact relationship between ϕ and $\mathbf{J}$ is defined by integrating Eq. (2.16a) over $\hat{\Omega} \in 4\pi$:

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial \phi}{\partial t}(\mathbf{x}, E, t) + \nabla \cdot \mathbf{J}(\mathbf{x}, E, t) - \Sigma_t(\mathbf{x}, E, t) \phi(\mathbf{x}, E, t) = \\ \int_0^\infty \Sigma_{s0}(\mathbf{x}, E' \rightarrow E, t) \phi(\mathbf{x}, E', t) \, dE' \\ + \chi_p(\mathbf{x}, E, t) \int_0^\infty [1 - \beta(\mathbf{x}, E', t)] \nu \Sigma_f(\mathbf{x}, E', t) \phi(\mathbf{x}, E', t) \, dE' \\ + \sum_{j=1}^J \chi_{d,j}(\mathbf{x}, E, t) \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t) + Q(\mathbf{x}, E, t). \end{aligned} \quad (2.93)$$

Equation (2.93) is referred to as the *neutron continuity equation* and introduces $\Sigma_{s0}(\mathbf{x}, E' \rightarrow E)$ as the first Legendre moment of the differential scattering cross section

$$\Sigma_{s0}(\mathbf{x}, E' \rightarrow E) = 2\pi \int_{-1}^1 \Sigma_s(\mu_0, E' \rightarrow E) \, d\mu. \quad (2.94)$$

Recall from Sec. 2.2 that the angular dependence of neutron scattering is rotationally invariant and only depends on the cosine of the scattering angle $\mu_0 = \hat{\Omega}' \cdot \hat{\Omega}$ between the incoming and outgoing direction vectors. As a cosine, μ_0 is bounded by $-1 \leq \mu_0 \leq 1$ and is well-represented as an expansion of Legendre polynomials

$$\Sigma_s(\mathbf{x}, E' \rightarrow E, \mu_0) = \sum_{n=0}^{\infty} \frac{2n+1}{4\pi} \Sigma_{sn}(\mathbf{x}, E' \rightarrow E) P_n(\mu_0), \quad (2.95)$$

where for $0 \leq n \leq \infty$,

$$\Sigma_{sn}(\mathbf{x}, E' \rightarrow E) = 2\pi \int_{-1}^1 P_n(\mu'_0) \Sigma_s(\mathbf{x}, \mu'_0, E) \, d\mu'_0. \quad (2.96)$$

The $\hat{\Omega}$ dependence of the angular flux itself can also be expanded as a series, this time using spherical

harmonics functions³

$$\psi(\mathbf{x}, \hat{\Omega}, E) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \psi_{n,m}(\mathbf{x}, E) Y_n^m(\hat{\Omega}), \quad (2.97)$$

where the spherical harmonics functions are expressed as

$$Y_n^m(\hat{\Omega}) = \left[\frac{2n+1}{4\pi} \frac{(n-|m|)!}{(n+|m|)!} \right]^{\frac{1}{2}} P_n(\mu) e^{im\gamma}, \quad 0 \leq |m| \leq n < \infty, \quad (2.98)$$

and satisfy the identities

$$Y_n^{-m}(\hat{\Omega}) = \overline{Y_n^m}(\hat{\Omega}), \quad (2.99a)$$

$$\int_{4\pi} Y_n^m(\hat{\Omega}) \overline{Y_j^k}(\hat{\Omega}) d\Omega = \delta_{n,j} \delta_{m,k}, \quad (2.99b)$$

where $\overline{Y_n^m}$ is the complex conjugate to Y_n^m . One popular approximation to the neutron transport equation is the *spherical harmonics*, or P_N , approximation. In the P_N approximation, the angular flux is approximated with a spherical harmonic expansion up to order N

$$\psi(\mathbf{x}, \hat{\Omega}, E) \approx \sum_{n=0}^N \sum_{m=-n}^n \psi_{n,m}(\mathbf{x}, E) Y_n^m(\hat{\Omega}). \quad (2.100)$$

Although the P_N approximation is not directly used in this dissertation work, it is the foundation for other approximations that are used in this work. A discussion of those approximations follows.

2.6.2.1 P_1 and Diffusion Approximations

The P_1 approximation derivation begins by assuming that the angular dependence of ψ is a linear function of $\hat{\Omega}$ and therefore the angular flux ψ , scalar flux ϕ , and neutron current vector $\mathbf{J}$ are related by

$$\psi(\mathbf{x}, \hat{\Omega}, E, t) = \frac{1}{4\pi} \left[\phi(\mathbf{x}, E, t) + 3\hat{\Omega} \cdot \mathbf{J}(\mathbf{x}, E, t) \right]. \quad (2.101)$$

Additionally, recall from Sec. 2.2 that neutron scattering has the property of rotational invariance, meaning that the outgoing direction of flight is only dependent on the cosine of the scattering angle $\mu_0 = \hat{\Omega}' \cdot \hat{\Omega}$. To avoid a full spherical harmonics derivation, assume that neutron scattering is *linearly anisotropic*, i.e., $\Sigma_s(\mathbf{x}, E' \rightarrow E, \mu_0)$ has only a linear dependence on μ_0 and the following expression is true after taking the

³Appendix A of Brody Bassett's doctoral dissertation[68] gives the definitions and identities associated with Legendre polynomials and spherical harmonics used in this dissertation.

expansion defined by Eq. (2.95)

$$\Sigma_s(\mathbf{x}, E' \rightarrow E, \mu_0, t) = \frac{1}{4\pi} [\Sigma_{s0}(\mathbf{x}, E' \rightarrow E, t) + 3\mu_0 \Sigma_{s1}(\mathbf{x}, E' \rightarrow E, t)] . \quad (2.102)$$

Substituting Eq. (2.102) into Eq. (2.16a) gives

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial \psi}{\partial t}(\mathbf{x}, \hat{\Omega}, E, t) + \nabla \cdot \hat{\Omega} \psi(\mathbf{x}, \hat{\Omega}, E, t) - \Sigma_t(\mathbf{x}, E, t) \psi(\mathbf{x}, \hat{\Omega}, E, t) = \\ \frac{1}{4\pi} \int_0^\infty \int_{4\pi} [\Sigma_{s0}(\mathbf{x}, E' \rightarrow E, t) + 3\mu_0 \Sigma_{s1}(\mathbf{x}, E' \rightarrow E, t)] \psi(\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE' \\ + \frac{1}{4\pi} \chi_p(\mathbf{x}, E, t) \int_0^\infty \int_{4\pi} [1 - \beta(\mathbf{x}, E', t)] \nu \Sigma_f(\mathbf{x}, E', t) \psi(\mathbf{x}, \hat{\Omega}', E', t) d\Omega' dE' \\ + \sum_{j=1}^J \frac{\chi_{d,j}(E)}{4\pi} \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t) + \frac{1}{4\pi} Q(\mathbf{x}, \hat{\Omega}, E, t) . \end{aligned} \quad (2.103)$$

Applying the flux expansion from Eq. (2.101) to Eq. (2.103) and taking the zeroth and first angular moments of the resulting equations and gives the P_1 equations

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial \phi}{\partial t}(\mathbf{x}, E, t) + \nabla \cdot \mathbf{J}(\mathbf{x}, E, t) - \Sigma_t(\mathbf{x}, E, t) \phi(\mathbf{x}, E, t) = \\ \int_0^\infty \Sigma_{s0}(\mathbf{x}, E' \rightarrow E, t) \phi(\mathbf{x}, E', t) dE' \\ + \chi_p(\mathbf{x}, E, t) \int_0^\infty [1 - \beta(\mathbf{x}, E', t)] \nu \Sigma_f(\mathbf{x}, E', t) \phi(\mathbf{x}, E', t) dE' \\ + \sum_{j=1}^J \chi_{d,j}(E) \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t) + Q(\mathbf{x}, E, t) , \end{aligned} \quad (2.104a)$$

and

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial J}{\partial t}(\mathbf{x}, E, t) + \frac{1}{3} \nabla \cdot \phi(\mathbf{x}, E, t) + \Sigma_t(\mathbf{x}, E, t) \mathbf{J}(\mathbf{x}, E, t) = \\ \int_0^\infty \Sigma_{s1}(\mathbf{x}, E' \rightarrow E, t) \mathbf{J}(\mathbf{x}, E', t) dE' . \end{aligned} \quad (2.104b)$$

Initial conditions must be specified to solve Eqs. (2.104) and are derived by taking the zeroth and first angular moments of Eq. (2.16d)

$$\phi(\mathbf{x}, E, 0) = \phi^0(\mathbf{x}, E) \equiv \int_{4\pi} \psi^0(\mathbf{x}, \hat{\Omega}, E, t) d\Omega, \quad \mathbf{x} \in V, \quad 0 \leq E < \infty, \quad (2.105a)$$

$$\mathbf{J}(\mathbf{x}, E, 0) = \mathbf{J}^0(\mathbf{x}, E) \equiv \int_{4\pi} \hat{\Omega} \psi^0(\mathbf{x}, \hat{\Omega}, E, t) d\Omega, \quad \mathbf{x} \in V, \quad 0 \leq E < \infty. \quad (2.105b)$$

Finally, the boundary conditions are defined by integrating Eq. (2.16c) over all incoming directions $\hat{\Omega} \cdot \hat{n} < 0$

$$\begin{aligned} \int_{\hat{\Omega} \cdot \hat{n} < 0} \left| \hat{\Omega} \cdot \hat{n} \right| \psi(\mathbf{x}, \hat{\Omega}, E, t) \, d\Omega &= \int_{\hat{\Omega} \cdot \hat{n} < 0} \left| \hat{\Omega} \cdot \hat{n} \right| \psi^b(\mathbf{x}, \hat{\Omega}, E, t) \, d\Omega \\ &\equiv J^-(\mathbf{x}, E, t) , \end{aligned} \quad (2.106a)$$

where $J^-(\mathbf{x}, E, t)$ is defined to be the incoming partial current and is a scalar quantity. The left-hand side of Eq. (2.106a) can be further expanded with Eq. (2.101) so that integral over incoming directions evaluates to

$$\begin{aligned} \int_{\hat{\Omega} \cdot \hat{n} < 0} \left| \hat{\Omega} \cdot \hat{n} \right| \psi(\mathbf{x}, \hat{\Omega}, E, t) \, d\Omega &\approx -\frac{1}{4\pi} \int_{\hat{\Omega} \cdot \hat{n} < 0} \left| \hat{\Omega} \cdot \hat{n} \right| \left[\phi(\mathbf{x}, E, t) + 3\hat{\Omega} \cdot \mathbf{J}(\mathbf{x}, E, t) \right] \, d\Omega \\ &= \frac{1}{4} \phi(\mathbf{x}, E, t) - \frac{1}{2} \hat{\Omega} \cdot \mathbf{J}(\mathbf{x}, E, t) . \end{aligned} \quad (2.106b)$$

Equating Eqs. (2.106) gives the *Marshak boundary condition*

$$\frac{1}{4} \phi(\mathbf{x}, E, t) - \frac{1}{2} \hat{\Omega} \cdot \mathbf{J}(\mathbf{x}, E, t) = J^-(\mathbf{x}, E, t) . \quad (2.107)$$

The diffusion approximation makes two additional assumptions to the P_1 equations. First, assume that the time rate of change of the neutron current vector can be ignored

$$\frac{\partial \mathbf{J}}{\partial t} \approx 0 . \quad (2.108)$$

Second, consider the following expression for the neutron current density after rearranging Eq. (2.104b)

$$\mathbf{J}(\mathbf{x}, E, t) = -\frac{1}{3 \left[\Sigma_t(\mathbf{x}, E, t) - \frac{1}{J(\mathbf{x}, E, t)} \int_0^\infty \Sigma_{s1}(\mathbf{x}, E' \rightarrow E, t) \mathbf{J}(\mathbf{x}, E', t) \, dE' \right]} \nabla \phi(\mathbf{x}, E, t) , \quad (2.109)$$

where the diffusion coefficient is defined as the term in front of the gradient

$$D(\mathbf{x}, E, t) = \frac{1}{3 \left[\Sigma_t(\mathbf{x}, E, t) - \frac{1}{J(\mathbf{x}, E, t)} \int_0^\infty \Sigma_{s1}(\mathbf{x}, E' \rightarrow E, t) \mathbf{J}(\mathbf{x}, E', t) \, dE' \right]} . \quad (2.110)$$

To properly compute the continuous-energy diffusion coefficient, the integral over the in-scatter energy transfer reaction should be weighted by the neutron current spectrum. This calculation is particularly difficult within Monte Carlo solvers [69]. Additionally, Eq. (2.110) only makes sense if all of the components of $\mathbf{J}$ have the same energy dependence. To overcome these issues, another assumption is made that the medium is weakly absorbing. If this is true, then the in-scatter rate of neutrons from E' to E will be approximately

equal to out-scatter rate of neutrons from E to all other energies E' . The *out-scatter approximation* is written as

$$\int_0^\infty \Sigma_{s1}(\mathbf{x}, E' \rightarrow E, t) \mathbf{J}(\mathbf{x}, E', t) dE' \approx \int_0^\infty \Sigma_{s1}(\mathbf{x}, E \rightarrow E', t) \mathbf{J}(\mathbf{x}, E, t) dE'. \quad (2.111)$$

From Eq. (2.111), the current is no longer dependent on the integration variable and can be taken outside of the integral to define the approximate diffusion coefficient used in this work

$$\begin{aligned} D(\mathbf{x}, E, t) &\approx \frac{1}{3 \left[\Sigma_t(\mathbf{x}, E, t) - \frac{\mathbf{J}(\mathbf{x}, E, t) \cdot \mathbf{J}(\mathbf{x}, E, t)}{J(\mathbf{x}, E, t)} \int_0^\infty \Sigma_{s1}(\mathbf{x}, E' \rightarrow E, t) dE' \right]} \\ &= \frac{1}{3 [\Sigma_t(\mathbf{x}, E, t) - \mu_0 \Sigma_{s0}(\mathbf{x}, E, t)]} \\ &= \frac{1}{3 \Sigma_{tr}(\mathbf{x}, E, t)}, \end{aligned} \quad (2.112)$$

where Σ_{tr} is called the macroscopic transport cross section. Substituting Eq. (2.112) into Eq. (2.109) gives an approximate expression (known as *Fick's law*) for the neutron current vector

$$\mathbf{J}(\mathbf{x}, E, t) \approx -D(\mathbf{x}, E, t) \nabla \phi(\mathbf{x}, E, t). \quad (2.113)$$

Substituting Eq. (2.113) into Eq. (2.104a) gives the continuous-energy neutron diffusion equation

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial \phi}{\partial t}(\mathbf{x}, E, t) - \nabla \cdot D(\mathbf{x}, E, t) \nabla \phi(\mathbf{x}, E, t) - \Sigma_t(\mathbf{x}, E, t) \phi(\mathbf{x}, E, t) = \\ \int_0^\infty \Sigma_{s0}(\mathbf{x}, E' \rightarrow E, t) \phi(\mathbf{x}, E', t) dE' \\ + \chi_p(\mathbf{x}, E, t) \int_0^\infty [1 - \beta(\mathbf{x}, E', t)] \nu \Sigma_f(\mathbf{x}, E', t) \phi(\mathbf{x}, E', t) dE' \\ + \sum_{j=1}^J \chi_{d,j}(\mathbf{x}, E, t) \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t) + Q(\mathbf{x}, E, t). \end{aligned} \quad (2.114a)$$

The corresponding precursor equations can be derived by taking the zeroth angular moment of Eq. (2.16b)

$$\frac{\partial C_j}{\partial t}(\mathbf{x}, t) = \int_0^\infty \beta_j(\mathbf{x}, E', t) \nu_f \Sigma_f(\mathbf{x}, E', t) \phi(\mathbf{x}, E', t) dE' - \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t). \quad (2.114b)$$

The conditions for the diffusion approximation to be sufficiently accurate include that the spatial derivatives of the scalar flux must be weak in addition to angular dependence of the transport solution being nearly linear. While these conditions are not satisfied for many radiation shielding applications, they are generally met for most nuclear reactor configurations. For this reason, the diffusion approximation is likely the most prevalent approximation to the neutron transport equation behind the multigroup approximation (which can

be applied simultaneously) for reactor physics applications.

2.6.2.2 SP₃ Approximation

The P_N approximation in 3D is complex and results in a system of $(N + 1)^2$ equations. The system can be difficult to solve, as it is coupled through mixed spatial derivatives of angular moments of Eq. (2.16a). Sparing the reader a lengthy derivation, the steps to derive the P_N approximation can be described in words as:

1. Multiply Eq. (2.16a) by $\overline{Y}_j^k$ for each j and k satisfying $0 \leq |k| \leq j \leq N$.
2. Integrate the resulting equation over the entire angular space $\hat{\Omega} \in 4\pi$.
3. Introduce the spherical harmonic expansion of order N as defined in Eq. (2.100).

In contrast, these same steps applied to 1D geometries can be formulated as a system of $(N + 1)/2$ diffusion-like equations with direct second-order spatial derivatives coupled only through angular moments. For reasons described in more detail in subsequent paragraphs, apply these steps specifically to the k -eigenvalue formulation given by Eq. (2.22a) instead of the general neutron transport equation given by Eq. (2.16a). In 1D slab geometry for $N = 3$, the resulting equations are

$$\begin{aligned}
 -\frac{d}{dx} D_0(x, E) \frac{d}{dx} \Psi_0(x, E) + [\Sigma_t(x, E) - \Sigma_{s0}(x, E)] \Psi_0(x, E) = \\
 2[\Sigma_t(x, E) - \Sigma_{s0}(x, E)] \Psi_2(x, E) \\
 + \frac{\chi(x, E)}{k_{\text{eff}}} \int_0^\infty \nu \Sigma_f(x, E') \Psi_0(x, E') dE', \quad 0 \leq x \leq X, \quad (2.115a)
 \end{aligned}$$

$$\begin{aligned}
 -\frac{d}{dx} D_2(x, E) \frac{d}{dx} \Psi_2(x, E) + [\Sigma_t(x, E) - \Sigma_{s2}(x, E)] \Psi_2(x, E) = \\
 \frac{2}{5} \left\{ [\Sigma_t(x, E) - \Sigma_{s0}(x, E)] [\Psi_0(x, E) - 2\Psi_2(x, E)] \right. \\
 \left. - \frac{\chi(x, E)}{k_{\text{eff}}} \int_0^\infty \nu \Sigma_f(x, E') \Psi_0(x, E') dE' \right\}, \quad 0 \leq x \leq X, \quad (2.115b)
 \end{aligned}$$

with the corresponding boundary conditions defined for all energies $0 \leq E < \infty$

$$\frac{1}{2}\Psi_0(0, E) + D_0(0, E) \frac{d}{dx}\Psi_0(0, E) = 2 \int_0^1 P_1(\mu) \psi^b(\mu, E) d\mu + \frac{3}{8}\Psi_2(0, E), \quad (2.115c)$$

$$\frac{1}{2}\Psi_0(X, E) + D_0(X, E) \frac{d}{dx}\Psi_0(X, E) = 2 \int_{-1}^0 P_1(|\mu|) \psi^b(\mu, E) d\mu + \frac{3}{8}\Psi_2(X, E), \quad (2.115d)$$

$$\frac{21}{40}\Psi_2(0, E) + D_2(0, E) \frac{d}{dx}\Psi_2(0, E) = \frac{6}{5} \int_0^1 P_3(\mu) \psi^b(\mu, E) d\mu + \frac{3}{40}\Psi_0(0, E), \quad (2.115e)$$

$$\frac{21}{40}\Psi_2(X, E) + D_2(X, E) \frac{d}{dx}\Psi_2(X, E) = \frac{6}{5} \int_{-1}^0 P_3(|\mu|) \psi^b(\mu, E) d\mu + \frac{3}{40}\Psi_0(X, E). \quad (2.115f)$$

The fluxes Ψ_0 and Ψ_2 are defined to make the notation more convenient

$$\Psi_0(x, E) \equiv \psi_0(x, E) + 2\psi_2(x, E), \quad (2.116a)$$

$$\Psi_2(x, E) \equiv \psi_2(x, E), \quad (2.116b)$$

and ψ_0 and ψ_2 are the zeroth and second order angular moments of the angular flux. Finally, the diffusion coefficients in Eq. (2.115) are defined as

$$D_0(x, E) = \frac{1}{3} \frac{1}{[\Sigma_t(x, E) - \Sigma_{s1}(x, E)]}, \quad (2.117)$$

$$D_2(x, E) = \frac{9}{35} \frac{1}{[\Sigma_t(x, E) - \Sigma_{s3}(x, E)]}. \quad (2.118)$$

Gelbard first derived the Simplified P_N (SP_N) equations in the early 1960s [70]–[72] by replacing the spatial derivatives $-\frac{d}{dx}D(x, E)\frac{d}{dx}$ with the general multi-dimensional Laplacian operator $\nabla \cdot D(\mathbf{x}, E)\nabla$. The 3D SP_3 eigenvalue equations are then

$$\begin{aligned} -\nabla \cdot D_0(\mathbf{x}, E)\nabla\Psi_0(\mathbf{x}, E) + [\Sigma_t(\mathbf{x}, E) - \Sigma_{s0}(\mathbf{x}, E)]\Psi_0(\mathbf{x}, E) = \\ 2[\Sigma_t(\mathbf{x}, E) - \Sigma_{s0}(\mathbf{x}, E)]\Psi_2(\mathbf{x}, E) \\ + \frac{\chi(\mathbf{x}, E)}{k_{\text{eff}}} \int_0^\infty \nu\Sigma_f(\mathbf{x}, E')\Psi_0(\mathbf{x}, E') dE', \quad \mathbf{x} \in V, \end{aligned} \quad (2.119a)$$

$$\begin{aligned} -\nabla \cdot D_2(\mathbf{x}, E)\nabla\Psi_2(\mathbf{x}, E) + [\Sigma_t(\mathbf{x}, E) - \Sigma_{s2}(\mathbf{x}, E)]\Psi_2(\mathbf{x}, E) = \\ \frac{2}{5} \left\{ [\Sigma_t(\mathbf{x}, E) - \Sigma_{s0}(\mathbf{x}, E)] [\Psi_0(\mathbf{x}, E) - 2\Psi_2(\mathbf{x}, E)] \right. \\ \left. - \frac{\chi(\mathbf{x}, E)}{k_{\text{eff}}} \int_0^\infty \nu\Sigma_f(\mathbf{x}, E')\Psi_0(\mathbf{x}, E') dE' \right\}, \quad \mathbf{x} \in V, \end{aligned} \quad (2.119b)$$

with corresponding boundary conditions

$$\begin{aligned} \frac{1}{2} \Psi_0(\mathbf{x}, E) + D_0(\mathbf{x}, E) \hat{\mathbf{n}} \cdot \nabla \Psi_0(\mathbf{x}, E) = \\ 2 \int_{\hat{\Omega} \cdot \hat{\mathbf{n}} < 0} P_1 \left(\left| \hat{\Omega} \cdot \hat{\mathbf{n}} \right| \right) \psi^b(\mathbf{x}, E) \, d\Omega + \frac{3}{8} \Psi_2(\mathbf{x}, E), \quad \mathbf{x} \in \partial V, \quad 0 \leq E < \infty, \end{aligned} \quad (2.119c)$$

$$\begin{aligned} \frac{21}{40} \Psi_2(\mathbf{x}, E) + D_2(\mathbf{x}, E) \hat{\mathbf{n}} \cdot \nabla \Psi_2(\mathbf{x}, E) = \\ \frac{6}{5} \int_{\hat{\Omega} \cdot \hat{\mathbf{n}} < 0} P_3 \left(\left| \hat{\Omega} \cdot \hat{\mathbf{n}} \right| \right) \psi^b(\mathbf{x}, E) \, d\Omega + \frac{3}{40} \Psi_0(\mathbf{x}, E), \quad \mathbf{x} \in \partial V, \quad 0 \leq E < \infty. \end{aligned} \quad (2.119d)$$

For nearly 30 years, the SP_N approximation continued to be used for practical reactor analyses with impressively accurate results despite the weak justification proposed in Gelbard's derivation. Though Gelbard was aware of this, it wasn't until the 1990s that asymptotic [73] and variational [74] analyses were used to provide more rigorous justifications for the accuracy of the SP_N approximation.

In the context of this work, the SP_3 approximation is only used to solve the adjoint eigenvalue equations used within the subsolvers of the TML method described in Section 3.2. Ideally, an SP_3 solution of the adjoint eigenvalue equations would not be necessary to implement the TML method, as it is possible to compute adjoint-weighted tallies directly from a Monte Carlo solver or reformulate the forward CMFD to solve adjoint problems. Alas, at the time of writing, adjoint-weighted tallies are not a supported feature within the MC code used throughout this dissertation work and the only adjoint eigenvalue solution available within the suite of deterministic solvers used for this dissertation was computed using an SP_3 solver. The SP_3 solution was used instead to take advantage of the existing capabilities enabled in Shift to reduce additional implementation time.

The adjoint eigenvalue form of the SP_3 equations are given as

$$\begin{aligned} \nabla \cdot D_0(\mathbf{x}, E) \nabla \Psi_0^\dagger(\mathbf{x}, E) + \left[\Sigma_t(\mathbf{x}, E) - \Sigma_{s0}^\dagger(\mathbf{x}, E) \right] \Psi_0^\dagger(\mathbf{x}, E) = \\ 2 \left[\Sigma_t(\mathbf{x}, E) - \Sigma_{s0}^\dagger(\mathbf{x}, E) \right] \Psi_2^\dagger(\mathbf{x}, E) \\ + \frac{1}{k_{\text{eff}}} \nu \Sigma_f(\mathbf{x}, E) \int_0^\infty \chi(\mathbf{x}, E') \Psi_0^\dagger(\mathbf{x}, E') \, dE', \quad \mathbf{x} \in V, \quad 0 \leq E < \infty, \end{aligned} \quad (2.120a)$$

$$\begin{aligned} \nabla \cdot D_2(\mathbf{x}, E) \nabla \Psi_2^\dagger(\mathbf{x}, E) + \left[\Sigma_t(\mathbf{x}, E) - \Sigma_{s2}^\dagger(\mathbf{x}, E) \right] \Psi_2^\dagger(\mathbf{x}, E) = \\ \frac{2}{5} \left\{ \left[\Sigma_t(\mathbf{x}, E) - \Sigma_{s0}^\dagger(\mathbf{x}, E) \right] \left[\Psi_0^\dagger(\mathbf{x}, E) - 2\Psi_2^\dagger(\mathbf{x}, E) \right] \right. \\ \left. - \frac{1}{k_{\text{eff}}} \nu \Sigma_f(\mathbf{x}, E) \int_0^\infty \chi(\mathbf{x}, E') \Psi_0^\dagger(\mathbf{x}, E') \, dE' \right\}, \quad \mathbf{x} \in V, \quad 0 \leq E < \infty. \end{aligned} \quad (2.120b)$$

Marshak-like boundary conditions are used that ensure no incoming current, which is appropriate for eigenvalue

problems. These boundary conditions are consistent with those derived by Brantley and Larsen [74] and take the form

$$\begin{aligned} \frac{1}{2}\Psi_0^\dagger(\mathbf{x}, E) - D_0(\mathbf{x}, E) \hat{\mathbf{n}} \cdot \nabla \Psi_0^\dagger(\mathbf{x}, E) = \\ 2 \int_{\hat{\Omega} \cdot \hat{\mathbf{n}} > 0} P_1(\hat{\Omega} \cdot \hat{\mathbf{n}}) \psi^{\dagger b}(\mathbf{x}, \hat{\Omega}, E) d\Omega + \frac{3}{8}\Psi_2^\dagger(\mathbf{x}, E), \quad \mathbf{x} \in \partial V, \quad 0 \leq E < \infty, \end{aligned} \quad (2.120c)$$

$$\begin{aligned} \frac{21}{40}\Psi_2^\dagger(\mathbf{x}, E) - D_2(\mathbf{x}, E) \hat{\mathbf{n}} \cdot \nabla \Psi_2^\dagger(\mathbf{x}, E) = \\ \frac{6}{5} \int_{\hat{\Omega} \cdot \hat{\mathbf{n}} > 0} P_3(\hat{\Omega} \cdot \hat{\mathbf{n}}) \psi^{\dagger b}(\mathbf{x}, \hat{\Omega}, E) d\Omega + \frac{3}{40}\Psi_0^\dagger(\mathbf{x}, E), \quad \mathbf{x} \in \partial V, \quad 0 \leq E < \infty. \end{aligned} \quad (2.120d)$$

The $\Sigma_{sn}^\dagger$ terms in Eqs. (2.120) are the angular moments of order n for the adjoint scattering cross section where neutrons scatter from $E \rightarrow E'$

$$\Sigma_{sn}^\dagger(\mathbf{x}, E) = \int_0^\infty \Sigma_{sn}(\mathbf{x}, E \rightarrow E') dE'. \quad (2.121)$$

2.7 Transient Fixed Source Problem

Let Eq. (2.16a) be written in operator notation as

$$\frac{1}{v(E)} \frac{\partial \psi}{\partial t}(\mathbf{x}, \hat{\Omega}, E, t) + \mathbf{M}\psi(\mathbf{x}, \hat{\Omega}, E, t) = \frac{1}{k_{\text{eff}}^0} \mathbf{F}_p \psi(\mathbf{x}, \hat{\Omega}, E, t) + \sum_{j=1}^J S_{d,j}(\mathbf{x}, \hat{\Omega}, E, t), \quad (2.122a)$$

$$\frac{\partial C_j}{\partial t}(\mathbf{x}, t) = \frac{1}{\chi_{d,j}(\mathbf{x}, E, t)} \frac{1}{k_{\text{eff}}^0} \mathbf{F}_{d,j} \psi(\mathbf{x}, \hat{\Omega}, E, t) - \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t), \quad (2.122b)$$

where the operators $\mathbf{F}_p$, $\mathbf{F}_{d,j}$ and the delayed neutron source term $S_{d,j}$ are defined in Eqs. (2.44), (2.45), and (2.48), respectively. Note the appearance of the initial k -eigenvalue $\frac{1}{k_{\text{eff}}^0}$ normalizing the terms that contain the fission source integral. This scaling of the fission source is in anticipation of taking the inner product of the angular flux weighted with the k -eigenvalue adjoint flux to generate the point kinetics parameters as was discussed in Sec. 2.4. There are several choices to discretize the temporal domain of Eqs. (2.122). For this work, an implicit Euler scheme is selected to avoid negative particle weights when the equation is solved with the Monte Carlo method. The implicit Euler discretization is derived by integrating Eq. (2.122a) over the

range $t^{n-1} \leq t \leq t^n$ and approximating all time-dependent quantities using a right-hand Riemann sum

$$\begin{aligned} \frac{1}{v(E) \Delta t^n} \left[\psi^n(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E) - \psi^{n-1}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E) \right] + \mathbf{M} \psi^n(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E) \\ = \frac{1}{k_{\text{eff}}^0} \mathbf{F}_p \psi^n(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E) + \sum_{j=1}^J S_{\text{d},j}^n(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E). \end{aligned} \quad (2.123)$$

To integrate Eq. (2.122b), approximate $\frac{1}{\chi_{\text{d},j}(\mathbf{x}, E, t)} \frac{1}{k_{\text{eff}}^0} \mathbf{F}_{\text{d},j} \psi(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t)$ as varying linearly in time⁴ to arrive at

$$\begin{aligned} C_j^n(\mathbf{x}) = C_j^{n-1}(\mathbf{x}) f_{1,j} + \frac{1}{\chi_{\text{d},j}^{n-1}(\mathbf{x}, E)} \frac{1}{k_{\text{eff}}^0} \mathbf{F}_{\text{d},j} \psi^{n-1}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) f_{2,j} \\ + \frac{1}{\chi_{\text{d},j}^n(\mathbf{x}, E)} \frac{1}{k_{\text{eff}}^0} \mathbf{F}_{\text{d},j} \psi^n(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E) f_{3,j}, \end{aligned} \quad (2.124)$$

where the time-interpolated weighting factors are defined as

$$f_{1,j} = e^{-\lambda_j \Delta t^n}, \quad (2.125a)$$

$$f_{2,j} = \frac{1 - e^{-\lambda_j \Delta t^n} - \lambda_j \Delta t^n e^{-\lambda_j \Delta t^n}}{\lambda_j \Delta t^n}, \quad (2.125b)$$

$$f_{3,j} = \frac{e^{-\lambda_j \Delta t^n} (1 - e^{\lambda_j \Delta t^n} + \lambda_j \Delta t^n e^{\lambda_j \Delta t^n})}{\lambda_j \Delta t^n}, \quad (2.125c)$$

and $\Delta t^n = t^n - t^{n-1}$. A derivation of these time interpolation factors and a more detailed description of the integration of the precursor equations can be found in Appendix A. Recall that the accumulated delayed neutron source term $S_{\text{d},j}$ is dependent on C_j through Eq. (2.47)

$$S_{\text{d},j}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) = \frac{1}{4\pi} \chi_{\text{d},j}(\mathbf{x}, E, t) \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t). \quad (2.126)$$

⁴When integrating the precursor equations for a deterministic transport solver, $\frac{1}{\chi_{\text{d},j}(\mathbf{x}, E, t)} \frac{1}{k_{\text{eff}}^0} \mathbf{F}_{\text{d},j} \psi(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t)$ is typically approximated as being quadratic in time. This however could allow negative particle weights in the MC calculation, which is technically possible but generally avoided as negative particle weights have no physical interpretation.

Substituting Eq. (2.124) into Eq. (2.123), the Transient Fixed Source Problem (TFSP) is defined as

$$\begin{aligned}
\left(M + \frac{1}{v(E) \Delta t^n}\right) \psi^n(\mathbf{x}, \hat{\Omega}, E) &= \frac{\psi^{n-1}(\mathbf{x}, \hat{\Omega}, E)}{v(E) \Delta t^n} + \mathbf{F}_p \psi^n(\mathbf{x}, \hat{\Omega}, E) \\
&+ \sum_{j=1}^6 \mathbf{S}_{d,j}^{n-1}(\mathbf{x}, \hat{\Omega}, E) f_{1,j} \\
&+ \sum_{j=1}^6 \mathbf{F}_{d,j} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) f_{2,j} + \sum_{j=1}^6 \mathbf{F}_{d,j} \psi^n(\mathbf{x}, \hat{\Omega}, E) f_{3,j}. \quad (2.127)
\end{aligned}$$

Written in this way, the TFSP is indeed able to be solved as a fixed-source problem.

2.7.1 Diffusion Formulation

The process for deriving the TFSP for the neutron diffusion approximation is largely the same as for the full transport derivation. Recall the continuous-energy neutron diffusion approximation from Eqs. (2.114). Assume the delayed neutron fractions $\beta_j(\mathbf{x}, E, t)$ have been averaged over all energies as in Eq. (2.19) and replaced by $\bar{\beta}_j(\mathbf{x}, t)$ to arrive at

$$\begin{aligned}
\frac{1}{v(E)} \frac{\partial \phi(\mathbf{x}, E, t)}{\partial t} &= \nabla \cdot D(\mathbf{x}, E, t) \nabla \phi(\mathbf{x}, E, t) - \Sigma_t(\mathbf{x}, E, t) \phi(\mathbf{x}, E, t) \\
&+ \int_0^\infty \Sigma_s(\mathbf{x}, E' \rightarrow E, t) \phi(\mathbf{x}, E', t) dE' \\
&+ \chi_p(\mathbf{x}, E, t) (1 - \bar{\beta}(\mathbf{x}, t)) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi(\mathbf{x}, E, t) + \chi_d(\mathbf{x}, E, t) S_d(\mathbf{x}, E, t), \quad (2.128a)
\end{aligned}$$

$$\frac{\partial C_j}{\partial t}(\mathbf{x}, t) = \bar{\beta}_j(\mathbf{x}, t) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi(\mathbf{x}, E, t) - \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t), \quad (2.128b)$$

where the fission source terms are again scaled by the initial steady-state eigenvalue and the delayed source and fission operator terms are redefined to be

$$S_d(\mathbf{x}, E, t) \equiv \sum_j^J \lambda_j(\mathbf{x}, t) C_j(\mathbf{x}, t), \quad (2.129a)$$

$$\mathbf{F} \phi(\mathbf{x}, E, t) \equiv \int_0^\infty \nu \Sigma_f(\mathbf{x}, E', t) \phi(\mathbf{x}, E', t) dE'. \quad (2.129b)$$

Again, an implicit Euler time discretization is applied to Eq. (2.128a). This time though, the delayed neutron source S_d is approximated as behaving quadratically in time as opposed to linearly with the transport equation. This is because the chosen TML scheme (see Sec. 3.2) uses the MC method (see Sec. 2.10) to solve the transport-level TFSP and quadratic time interpolation factors could result in non-ideal negative particle weights. The diffusion-level TFSP will instead be solved by the deterministic CMFD method, which does not

need to avoid negative interpolation factors. For a more detailed explanation on how to apply the quadratic approximation and implicit Euler time discretization to Eq. (2.128b), see Appendix A. After applying the discretization, Eq. (2.128a) becomes

$$\begin{aligned}
\frac{\phi^n(\mathbf{x}, E) - \phi^{n-1}(\mathbf{x}, E)}{v(E) \Delta t^n} = & \nabla \cdot D^n(\mathbf{x}, E) \nabla \phi^n(\mathbf{x}, E) - \Sigma_t^n(\mathbf{x}, E) \phi^n(\mathbf{x}, E) \\
& + \int_0^\infty \Sigma_{s0}^n(\mathbf{x}, E' \rightarrow E) \phi^n(\mathbf{x}, E') dE' \\
& + \chi_p^n(\mathbf{x}, E) \left(1 - \bar{\beta}^n(\mathbf{x})\right) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^n(\mathbf{x}, E) \\
& + \chi_d^n(\mathbf{x}, E) S_d^n(\mathbf{x}, E) ,
\end{aligned} \tag{2.130}$$

where the time-interpolated delayed neutron terms are defined as

$$S_d^n(\mathbf{x}, E) = \omega^n(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^n(\mathbf{x}, E) + \tilde{S}_d^{n-1}(\mathbf{x}, E) , \tag{2.131}$$

$$\omega^n(\mathbf{x}) = \sum_{j=1}^6 \bar{\beta}_j^n(\mathbf{x}) \Omega_j^n(\lambda_j^n(\mathbf{x})) , \tag{2.132}$$

$$\begin{aligned}
\tilde{S}_d^{n-1}(\mathbf{x}, E) = & \sum_{j=1}^6 \lambda_j^n(\mathbf{x}) \Omega_j^0(\lambda_j^n(\mathbf{x})) C_j^{n-1}(\mathbf{x}) \\
& + \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^{n-1}(\mathbf{x}, E) \sum_{j=1}^6 \bar{\beta}_j^{n-1}(\mathbf{x}) \Omega_j^{n-1}(\lambda_j^n(\mathbf{x})) \\
& + \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^{n-2}(\mathbf{x}, E) \sum_{j=1}^6 \bar{\beta}_j^{n-2}(\mathbf{x}) \Omega_j^{n-2}(\lambda_j^n(\mathbf{x})) ,
\end{aligned} \tag{2.133}$$

and the Ω functions are defined in Eqs. (A.14). Substituting Eq. (2.131) into Eq. (2.130) gives

$$\begin{aligned}
\frac{\phi^n(\mathbf{x}, E) - \phi^{n-1}(\mathbf{x}, E)}{v(E) \Delta t^n} = & \nabla \cdot D^n(\mathbf{x}, E) \nabla \phi^n(\mathbf{x}, E) - \Sigma_t^n(\mathbf{x}, E) \phi^n(\mathbf{x}, E) \\
& + \int_0^\infty \Sigma_{s0}^n(\mathbf{x}, E' \rightarrow E) \phi^n(\mathbf{x}, E') dE' \\
& + \chi_p^n(\mathbf{x}, E) \left(1 - \bar{\beta}^n(\mathbf{x})\right) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^n(\mathbf{x}, E) \\
& + \chi_d^n(\mathbf{x}, E) \omega^n(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^n(\mathbf{x}, E) + \chi_d^n(\mathbf{x}, E) \tilde{S}_d^{n-1}(\mathbf{x}, E) .
\end{aligned} \tag{2.134}$$

Then using the cumulative fission spectrum defined in Eq. (2.20)

$$\chi^n(\mathbf{x}, E) = \chi_p^n(\mathbf{x}, E) \left(1 - \bar{\beta}^n(\mathbf{x})\right) + \chi_d^n(\mathbf{x}, E) \bar{\beta}^n(\mathbf{x}) , \tag{2.135}$$

Equation (2.130) becomes

$$\begin{aligned} \frac{\phi^n(\mathbf{x}, E) - \phi^{n-1}(\mathbf{x}, E)}{v(E) \Delta t^n} = & \nabla \cdot D^n(\mathbf{x}, E) \nabla \phi^n(\mathbf{x}, E) - \Sigma_t^n(\mathbf{x}, E) \phi^n(\mathbf{x}, E) \\ & + \int_0^\infty \Sigma_{s0}^n(\mathbf{x}, E' \rightarrow E) \phi^n(\mathbf{x}, E') dE' + \chi^n(\mathbf{x}, E) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^n(\mathbf{x}, E) \\ & + \chi_d^n(\mathbf{x}, E) \left(\omega^n(\mathbf{x}) - \bar{\beta}^n(\mathbf{x}) \right) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^n(\mathbf{x}, E) + \chi_d^n(\mathbf{x}, E) \tilde{S}_d^{n-1}(\mathbf{x}) . \end{aligned} \quad (2.136)$$

The final balance equation is written as

$$\begin{aligned} -\nabla \cdot D^n(\mathbf{x}, E) \nabla \phi^n(\mathbf{x}, E) + \Sigma_t^n(\mathbf{x}, E) \phi^n(\mathbf{x}, E) = & \int_0^\infty \Sigma_{s0}^n(\mathbf{x}, E' \rightarrow E) \phi^n(\mathbf{x}, E') dE' \\ & + \chi^n(\mathbf{x}, E) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^n(\mathbf{x}, E) + S_{\text{tr}}^n(\mathbf{x}, E) , \end{aligned} \quad (2.137)$$

where the transient source is defined as

$$S_{\text{tr}}^n(\mathbf{x}, E) = A^n(\mathbf{x}, E) \phi^n(\mathbf{x}, E) + B^n(\mathbf{x}, E) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \phi^n(\mathbf{x}, E) + C^{n-1}(\mathbf{x}, E) . \quad (2.138)$$

Here A and B are flux and fission source dependent coefficients, and C is a constant coefficient depending on quantities defined at the previous point in time t^{n-1}

$$A^n(\mathbf{x}, E) = -\frac{1}{v(E) \Delta t^n} , \quad (2.139a)$$

$$B^n(\mathbf{x}, E) = \chi_d^n(\mathbf{x}, E) \left(\omega^n(\mathbf{x}) - \bar{\beta}^n(\mathbf{x}) \right) , \quad (2.139b)$$

$$C^{n-1}(\mathbf{x}, E) = \chi_d^n(\mathbf{x}, E) \tilde{S}_d^{n-1}(\mathbf{x}, E) + \frac{\phi^{n-1}(\mathbf{x}, E)}{v(E) \Delta t^n} . \quad (2.139c)$$

2.8 Coarse Mesh Finite Difference Equations

Large or complex reactor geometries commonly suffer from issues of slow convergence for iterative solution schemes. Oftentimes, these issues are caused by the presence of materials with high scattering ratios (the ratio of the scattering cross section to the total cross section) like water or even nuclear fuel. To address these issues, several acceleration techniques have been developed where a reduced-order system of equations is usually constructed between iterations to correct the solution for the next iteration. For fixed source calculations, early acceleration methods included using Chebychev-weighted averages of the flux at previous source iterations to estimate the final converged flux [75], using directionally-dependent rebalance factors to construct the neutron continuity equation and predict the flux at future iterations [76], and matrix preconditioners. Today,

Diffusion Synthetic Acceleration (DSA) [77]–[79] is arguably the most prominent acceleration technique for deterministic fixed-source calculations. DSA uses the difference between two subsequent source iterations to construct equations resembling neutron diffusion. The solution of this linear diffusion-like system serves as an additive correction to the transport flux. DSA remains a popular technique to accelerate deterministic source iteration schemes, and is extendable to eigenvalue calculations as a nonlinear scheme.

In addition to the iterative convergence issues described for fixed-source calculations, large reactor systems can additionally suffer from issues of high dominance ratios (the ratio of the first two largest eigenvalues). Popular acceleration techniques for eigenvalue calculations include the Coarse Mesh Rebalance (CMR) [80] and CMFD [81], [82] techniques. While both of these techniques are well-established in both deterministic and MC production codes, CMFD is more widely-adopted due to its simplicity, relative stability for coarse spatial (see Fig. 2.5) and energy grids, and speed-up. In this dissertation work, the CMFD method is **not** used as an eigenvalue acceleration technique, but rather as a level of the hierarchical TML scheme that is discussed in Ch. 3. In this capacity, the CMFD equations are formulated as a system of transport-corrected diffusion-level TFSP equations.

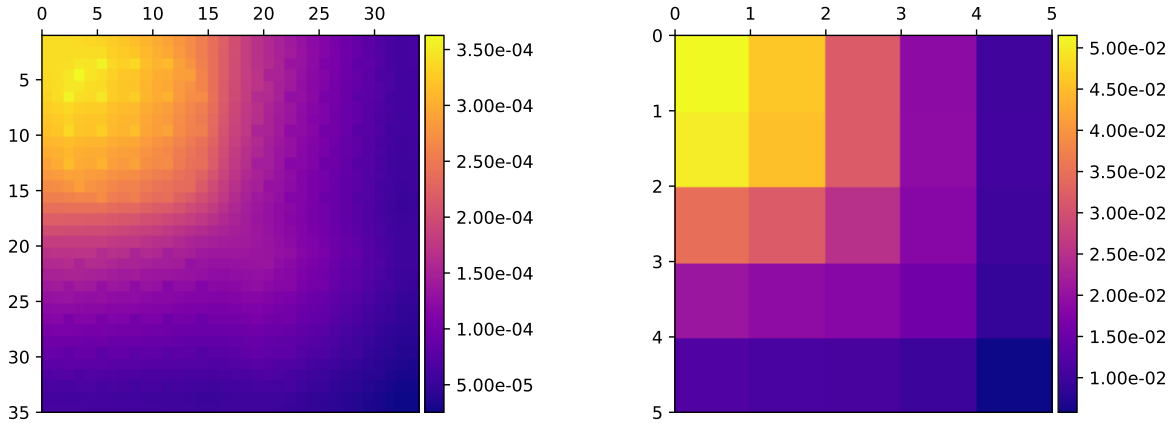


Figure 2.5: Scalar flux of a reactor quarter-core discretized on both fine (left) and coarse (right) spatial meshes, where the coarse mesh might represent the fidelity of a CMFD discretization. Scalar flux units are $\text{n cm}^{-2} \text{s}^{-1}$.

To derive the fixed-source CMFD equations, first recall the neutron balance equation from Eq. (2.93). Before considering temporal dependence, assume steady-state and let the internal source term $Q(\mathbf{x}, E) = 0$. Additionally assume that the cumulative fission spectrum from Eq. (2.20) is used

$$\begin{aligned} \nabla \cdot \mathbf{J}(\mathbf{x}, E) - \Sigma_t(\mathbf{x}, E) \phi(\mathbf{x}, E) &= \int_0^\infty \Sigma_{s0}(\mathbf{x}, E' \rightarrow E) \phi(\mathbf{x}, E') dE' \\ &+ \chi(\mathbf{x}, E) \int_0^\infty \nu \Sigma_f(\mathbf{x}, E') \phi(\mathbf{x}, E') dE'. \end{aligned} \quad (2.140)$$

Discretizing the energy domain of Eq. (2.140) is straightforward by applying the multigroup approximation. The spatial domain can be discretized by integrating Eq. (2.140) over a CMFD spatial element, i.e., $x_{i-\frac{1}{2}} \leq x \leq x_{i+\frac{1}{2}}$, $y_{j-\frac{1}{2}} \leq y \leq y_{j+\frac{1}{2}}$, $z_{i-\frac{1}{2}} \leq z \leq z_{i+\frac{1}{2}}$. An example CMFD spatial discretization in 2D is shown in Fig. 2.6. The resulting fully-discretized neutron balance equation is given by⁵

$$\sum_{u \in (x,y,z)} \Delta_m^v \Delta_n^w \left(\bar{J}_{l+\frac{1}{2},m,n,g}^u - \bar{J}_{l-\frac{1}{2},m,n,g}^u \right) + \Delta_l^u \Delta_m^v \Delta_n^w \bar{\bar{\Sigma}}_{t,l,m,n,g} \bar{\bar{\phi}}_{l,m,n,g} \\ = \Delta_l^u \Delta_m^v \Delta_n^w \left(\sum_{g'=1}^G \bar{\bar{\Sigma}}_{s0,l,m,n,g' \rightarrow g} \bar{\bar{\phi}}_{l,m,n,g'} + \bar{\bar{\chi}}_{l,m,n,g} \sum_{g'=1}^G \bar{\bar{\nu}} \bar{\bar{\Sigma}}_{f,l,m,n,g' \rightarrow g} \bar{\bar{\phi}}_{l,m,n,g'} \right). \quad (2.141)$$

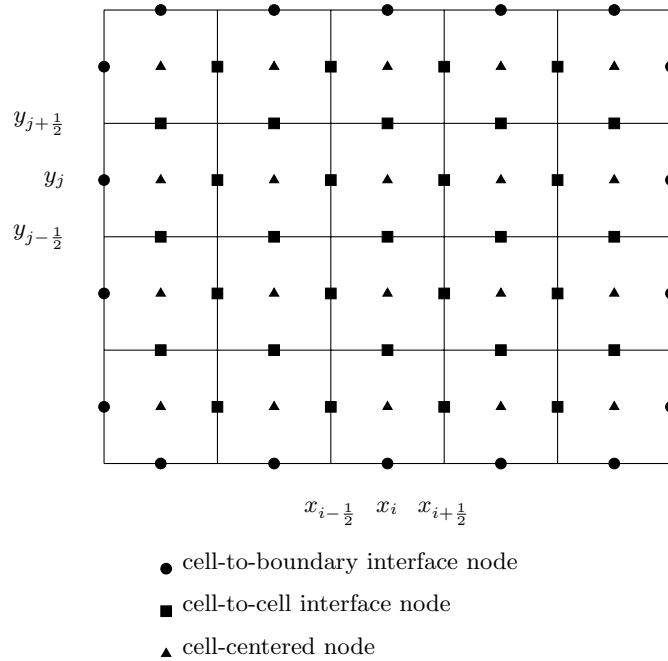


Figure 2.6: Example of a 2D CMFD spatial discretization.

Equation (2.141) serves as a second-order finite volume representation of Eq. (2.140) on a coarse mesh. As in all finite volume methods, an integral form of a conservation law must be applied to relate the quantity of interest in each spatial cell (the neutron scalar flux, ϕ) to the transfer of such quantity to adjacent cells (the neutron current, $\mathbf{J}$). These couplings exist between both cell-to-cell interfaces and cell-to-boundary interfaces and are defined as

⁵In Eq. (2.141), notation is borrowed from the OpenMC documentation [83]. When describing specific CMFD cells, the indices (i, j, k) are used and correspond to the directions (x, y, z) . For most cases though, the same operation is performed in all three directions. For a more compact notation, an arbitrary direction set, (u, v, w) , corresponding to the cell indices (l, m, n) is used. The $\bar{\bar{\square}}$ notation indicates that scalar quantity that is volume-averaged over a coarse CMFD cell, while the $\bar{\square}$ notation indicates a scalar quantity that is surface-averaged over the interfaces between the CMFD cells. The Δ terms described the respective length of a CMFD cell. For example, $\Delta_l^u = u_{l+\frac{1}{2}} - u_{l-\frac{1}{2}}$

- cell-to-cell coupling

$$\bar{J}_{l\pm\frac{1}{2},m,n,g}^u = - \underbrace{\frac{2\bar{\bar{D}}_{l\pm 1,m,n,g}\bar{\bar{D}}_{l,m,n,g}}{\bar{\bar{D}}_{l\pm 1,m,n,g}\Delta_l^u + \bar{\bar{D}}_{l,m,n,g}\Delta_{l\pm 1}^u}}_{\tilde{D}_{l,m,n,g}^{u,\text{cell}}} \left(\pm \bar{\bar{\phi}}_{l\pm 1,m,n,g} \mp \bar{\bar{\phi}}_{l,m,n,g} \right) , \quad (2.142a)$$

- cell-to-boundary coupling

$$\bar{J}_{l\pm\frac{1}{2},m,n,g}^u = \pm \underbrace{\frac{2\bar{\bar{D}}_{l,m,n,g} \left(1 - \alpha_{l\pm\frac{1}{2},m,n,g}^u \right)}{4\bar{\bar{D}}_{l,m,n,g} \left(1 + \alpha_{l\pm\frac{1}{2},m,n,g}^u \right) + \left(1 - \alpha_{l\pm\frac{1}{2},m,n,g}^u \right) \Delta_l^u}}_{\tilde{D}_{l,m,n,g}^{u,\text{bound}}} \bar{\bar{\phi}}_{l,m,n,g} , \quad (2.142b)$$

where

$$\alpha_{l\pm\frac{1}{2},m,n,g}^u = \frac{\bar{J}_{l\pm\frac{1}{2},m,n,g}^{u-}}{\bar{J}_{l\pm\frac{1}{2},m,n,g}^{u+}} \quad (2.143)$$

is the general albedo boundary condition. Common albedo conditions include: vacuum ($\alpha = 0$), reflective ($\alpha = 1$) and zero-flux ($\alpha = -1$). The $\tilde{D}_{l,m,n,g}^u$ parameter serves as the linear coupling term between the current and flux. Substituting Eqs. (2.142) as they are into Eq. (2.141) yields multigroup diffusion equations that are not consistent with higher order transport solutions. The addition of an equivalence factor, $\hat{D}_{l,m,n,g}^u$, is necessary to account for the truncation error associated with using Fick's Law. The modified Eqs. (2.142) are then

$$\bar{J}_{l\pm\frac{1}{2},m,n,g}^u = -\tilde{D}_{l,m,n,g}^{u,\text{cell}} \left(\pm \bar{\bar{\phi}}_{l\pm 1,m,n,g} \mp \bar{\bar{\phi}}_{l,m,n,g} \right) + \hat{D}_{l,m,n,g}^{u,\text{cell}} \left(\bar{\bar{\phi}}_{l\pm 1,m,n,g} + \bar{\bar{\phi}}_{l,m,n,g} \right) \quad (2.144a)$$

and

$$\bar{J}_{l\pm\frac{1}{2},m,n,g}^u = \pm \tilde{D}_{l,m,n,g}^{u,\text{bound}} \bar{\bar{\phi}}_{l,m,n,g} + \hat{D}_{l,m,n,g}^{u,\text{bound}} \bar{\bar{\phi}}_{l,m,n,g} . \quad (2.144b)$$

Equations (2.144) can be used to solve for $\hat{D}_{l,m,n,g}^u$ at the previous transport solution iteration, as the remaining parameters can be obtained or derived from the transport solution. This process is ‘nonlinear’ because it is dependent on the flux which is updated on the next outer transport iteration. Finally, substituting Eqs. (2.144) into Eq. (2.141) and dividing by the CMFD cell volume creates a linear system that is consistent

with transport physics

$$\begin{aligned}
& \sum_{u \in x, y, z} \frac{1}{\Delta_l^u} \left[\left(-\tilde{D}_{l-\frac{1}{2}, m, n, g}^u - \hat{D}_{l-\frac{1}{2}, m, n, g}^u \right) \bar{\bar{\phi}}_{l-1, m, n, g} \right. \\
& \quad + \left(\tilde{D}_{l-\frac{1}{2}, m, n, g}^u + \tilde{D}_{l+\frac{1}{2}, m, n, g}^u - \hat{D}_{l-\frac{1}{2}, m, n, g}^u + \hat{D}_{l+\frac{1}{2}, m, n, g}^u \right) \bar{\bar{\phi}}_{l, m, n, g} \\
& \quad \left. + \left(-\tilde{D}_{l+\frac{1}{2}, m, n, g}^u + \hat{D}_{l+\frac{1}{2}, m, n, g}^u \right) \bar{\bar{\phi}}_{l+1, m, n, g} \right] - \bar{\bar{\Sigma}}_{t, l, m, n, g} \bar{\bar{\phi}}_{l, m, n, g} \\
& = \sum_{g'=1}^G \bar{\bar{\Sigma}}_{s0, l, m, n, g' \rightarrow g} \bar{\bar{\phi}}_{l, m, n, g'} + \bar{\bar{\chi}}_{l, m, n, g} \sum_{g'=1}^G \bar{\bar{\nu}}_{\Sigma_f, l, m, n, g' \rightarrow g} \bar{\bar{\phi}}_{l, m, n, g'} .
\end{aligned} \tag{2.145}$$

Equation (2.145) can alternatively be represented in energy and spatially-discretized matrix operator form as

$$\mathbb{M}\phi = \mathbb{S}\phi + \frac{1}{k_{\text{eff}}^0} \mathbb{F}\phi , \tag{2.146}$$

where

- $\mathbb{M} \equiv -\nabla \mathbb{D} \nabla + \bar{\bar{\Sigma}}_t$ is the discretized removal operator,
- $\mathbb{S} \equiv \bar{\bar{\Sigma}}_{s0}$ is the discretized scattering operator,
- $\mathbb{F} \equiv \nu \bar{\bar{\Sigma}}_f$ is discretized the fission operator.

Notice that Eq. (2.137) is the same as Eq. (2.141), except that Eq. (2.137) is defined at time $t = t^n$, has the fission source weighted by the initial eigenvalue $\frac{1}{k_{\text{eff}}^0}$, and has the additional transient source term $S_{\text{tr}}^n(\mathbf{x}, E)$. Making these modifications, the discretized CMFD formulation of the TFSP is written as

$$\begin{aligned}
& \sum_{u \in x, y, z} \frac{1}{\Delta_l^u} \left[\left(-\tilde{D}_{l-\frac{1}{2}, m, n, g}^{u, n} - \hat{D}_{l-\frac{1}{2}, m, n, g}^{u, n} \right) \bar{\bar{\phi}}_{l-1, m, n, g}^n \right. \\
& \quad + \left(\tilde{D}_{l-\frac{1}{2}, m, n, g}^{u, n} + \tilde{D}_{l+\frac{1}{2}, m, n, g}^{u, n} - \hat{D}_{l-\frac{1}{2}, m, n, g}^{u, n} + \hat{D}_{l+\frac{1}{2}, m, n, g}^{u, n} \right) \bar{\bar{\phi}}_{l, m, n, g}^n \\
& \quad \left. + \left(-\tilde{D}_{l+\frac{1}{2}, m, n, g}^{u, n} + \hat{D}_{l+\frac{1}{2}, m, n, g}^{u, n} \right) \bar{\bar{\phi}}_{l+1, m, n, g}^n \right] - \bar{\bar{\Sigma}}_{t, l, m, n, g}^n \bar{\bar{\phi}}_{l, m, n, g}^n \\
& = \sum_{g'=1}^G \bar{\bar{\Sigma}}_{s0, l, m, n, g' \rightarrow g}^n \bar{\bar{\phi}}_{l, m, n, g'}^n + \frac{\bar{\bar{\chi}}_{l, m, n, g}^n}{k_{\text{eff}}^0} \sum_{g'=1}^G \bar{\bar{\nu}}_{\Sigma_f, l, m, n, g' \rightarrow g}^n \bar{\bar{\phi}}_{l, m, n, g'}^n + \bar{\bar{S}}_{\text{tr}, l, m, n, g}^n ,
\end{aligned} \tag{2.147}$$

where

$$\bar{\bar{S}}_{\text{tr}, l, m, n, g}^n = \bar{\bar{A}}_{l, m, n, g}^n \bar{\bar{\phi}}_{l, m, n, g}^n + \bar{\bar{B}}_{l, m, n, g}^n \frac{1}{k_{\text{eff}}^0} \mathbf{F} \bar{\bar{\phi}}_{l, m, n, g}^n + C_{l, m, n, g}^{n-1} , \tag{2.148}$$

and $\bar{\bar{A}}$, $\bar{\bar{B}}$, and $\bar{\bar{C}}$ are analogous to those defined in Eqs. (2.139). Also, note that in Eqs. (2.147) and (2.148), the superscript n corresponds to the time step index, while the subscript n corresponds to the CMFD cell

spatial index. Zhu adds Eq. (2.148) to Eq. (2.146) to define the Multigroup Matrix (MGM) method

$$\begin{aligned}\mathbb{M}\phi &= \mathbb{S}\phi + \frac{1}{k_{\text{eff}}^0}\mathbb{F}\phi + \mathbb{S}_{\text{tr}} \\ &= \mathbb{S}\phi + \frac{1}{k_{\text{eff}}^0}\mathbb{F}\phi + \mathbb{A}\phi + \mathbb{B}\frac{1}{k_{\text{eff}}^0}\mathbb{F}\phi + \mathbb{C},\end{aligned}\tag{2.149}$$

where Eq. (2.149) can be rearranged and solved as a linear system using any standard matrix inversion technique

$$\left(\mathbb{M} - \mathbb{S} - \frac{1}{k_{\text{eff}}^0}\mathbb{F} - \mathbb{A} - \frac{1}{k_{\text{eff}}^0}\mathbb{B}\mathbb{F}\right)\phi = \mathbb{C}.\tag{2.150}$$

2.9 Deterministic Methods

Deterministic methods solve the neutron transport equation for $\psi(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t)$ by discretizing all of phase space. The resulting system of equations can then be solved using any number of numerical methods, though most of them are iterative schemes. The multigroup approximation (see Sec. 2.6.1) is the standard technique for discretizing the energy domain. The P_N approximation (see Sec. 2.6.2) is one technique for discretizing the angular domain, where the angular flux ψ is approximated as a finite series of spherical harmonics moments.

Perhaps the most popular angular discretization technique is the Discrete Ordinates (S_N) method, which approximates the continuous angular domain with discrete unit direction vectors, or “ordinates”, each having a corresponding quadrature weight to assist with integration over the angular domain. Figure 2.7 (left) shows an example of one simple *level-symmetric* quadrature, where the same discretization levels are used for all three direction cosines. Note that there are eight discrete ordinates in the octant shown, making this an “ S_8 ” discretization. One interesting feature of the S_N method is that the order in which the spatial nodes are visited when “sweeping” through the discretized phase space is dependent on the general direction of the discrete ordinate $\hat{\boldsymbol{\Omega}}_m$, where m is the ordinate index. The right side of Fig. 2.7 depicts a 2D S_N spatial discretization and the corresponding visit order for an ordinate with $\mu_m > 0$ and $\eta_m > 0$.

Another solution technique gaining attention that also uses the discrete ordinates approximation is the Method of Characteristics (MOC), which integrates out the angular dependence of the transport equations as opposed to sweeping through the space-angle grid in the direction of particle travel. The MOC converts the transport equation to a linear ODE by integrating along discrete characteristic rays and making the variable substitution $\psi(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) \rightarrow \psi(\mathbf{x}_0 + \ell\hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{\Omega}}, E, t)$, where $\mathbf{x}_0$ is some position of interest and ℓ is the distance traveled along $\hat{\boldsymbol{\Omega}}$. The MOC is able to handle complex geometries as well as Monte Carlo and easily parallelizable. In practice though, its large memory requirements limit it to 2D spatial solutions unless it is coupled with other approximations, such as 2D/1D and multigroup. The TML results presented

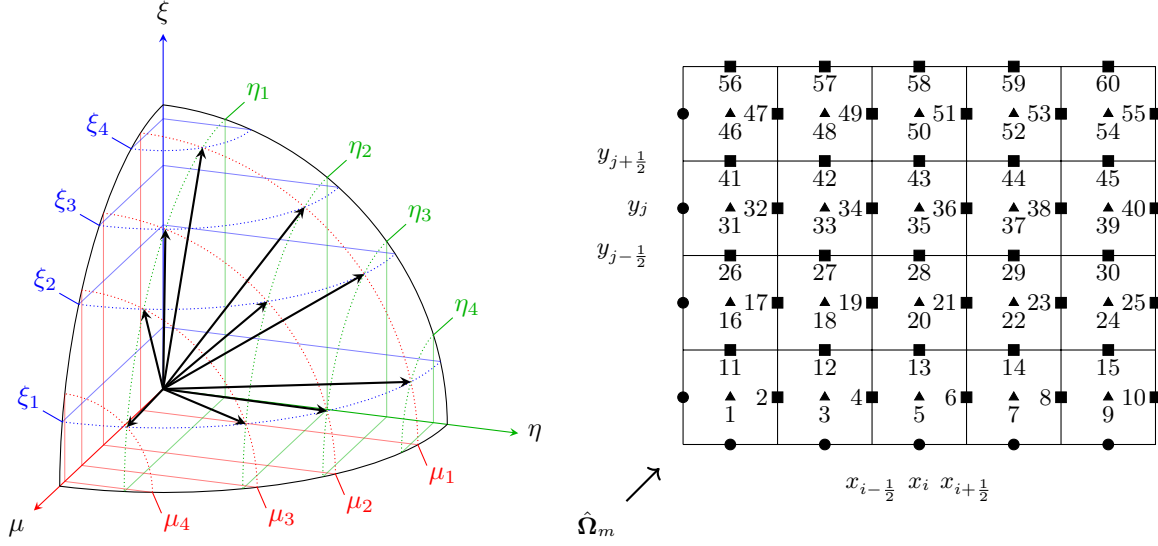


Figure 2.7: Level-symmetric S_8 ordinates (left) shown for a single octant of the unit sphere and the order of sweeping for 2D cell-centered spatial mesh (right) for $\mu_m > 0$, $\eta_m > 0$ [43]. The nodes on the spatial discretization diagram are differentiated as ■ $\rightarrow$ computed flux at cell interface, ● $\rightarrow$ known boundary flux, ▲ $\rightarrow$ computed flux at cell center.

for the C5G7-TD reactor problem in Ch. 3 are compared against the Michigan Parallel Characteristics Transport (MPACT) code, an MOC code maintained by University of Michigan and Oak Ridge National Laboratory (ORNL).

This dissertation is specifically focused on transient MC calculations. As such, the discussion here of deterministic methods is brief. However, there is no one solution technique for neutron transport problems that is superior to all others. Each technique has its own strengths and weaknesses. Increased attention and funding is being focused on developing numerical schemes that strike a balance between the accuracy of MC calculations and the speed of deterministic calculations. The work presented in this dissertation uses a MC solver at the highest level, but deterministic solvers at other levels in its solution hierarchy. For example, the adjoint SP_3 , CMFD, and EPKE solvers used within the TML algorithm are all deterministic and have either integrated out domains of the full phase space or discretized them.

2.10 Monte Carlo Method

The Monte Carlo method emulates the stochastic nature of particle transport by following the “history” of an individually simulated particle from birth by a particle source to death by leakage or absorption. A particle’s history is simulated by a random walk process, where events (such as streaming and collisions) are randomly sampled to follow transitions in phase-space. Each of these phase-space transitions are sampled from the

probability density functions defined by the neutron transport equation and are Markovian processes, where the immediate state is only dependent on the current state. The main advantage of the Monte Carlo method is that it does not require the discretization of phase space. This means that the spatial, energy, angular, and temporal domains of the neutron transport equation do not have to be approximated. Instead, transmissions and interactions are sampled from Probability Density Functions (PDFs), where the probability parameters come from either experimentally-measured nuclear data or first-principle models. In addition to continuous phase space, geometry can also be represented exactly (to the degree that is specified by a user). The main disadvantage to the Monte Carlo method is that, as a stochastic process, it is subject to statistical noise. In order for uncertainties to be resolved, many particle histories must be simulated to satisfy the central limit theorem, which states that the properly-normalized sum of independent and identically-distributed random trials (particle histories) will tend to a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2 . The standard deviation of the mean σ_μ converges to zero at a rate proportional to the inverse square of the number of histories N_h . For example, to achieve a $10\times$ reduction in uncertainty σ_μ , a $100\times$ increase in N_h is required. This means that simulating large and complex systems with the MC method with acceptably small uncertainties sometimes incurs extreme computational costs.

The general MC process for simulating neutron interactions is as follows: at the start of a history, a particle is initialized or “sourced” in phase space with a predefined position, direction, and energy. The particle can additionally be assigned an initial statistical weight w that can be interpreted as the number of particles in the physical world that a simulated particle represents. In this way, the distance to the nearest particle history event then has to be determined. In the simplest MC transport schemes, there are two possible events for a particle to travel to:

1. *collision*: The particle experiences a collision along its trajectory governed by the PDF

$$\begin{aligned}
 P(\ell) &= \Sigma_t e^{-\Sigma_t \ell} \\
 &= \text{the probability that a particle will collide} \\
 &\quad \text{with a nucleus between distances } \ell \text{ and } \ell + d\ell.
 \end{aligned} \tag{2.151}$$

In Eq. (2.151), Σ_t is the total macroscopic cross section of the homogenous material superimposed over

all nuclides. The corresponding Cumulative Distribution Function (CDF) is then

$$\begin{aligned}
 P(\ell' \leq \ell) &= \int_0^\ell \Sigma_t e^{-\Sigma_t \ell'} d\ell' \\
 &= 1 - e^{-\Sigma_t \ell} \\
 &= \text{the probability that a particle will experience} \\
 &\quad \text{a collision within distance } \ell.
 \end{aligned} \tag{2.152}$$

The distance to collision can then be computed by inverting the CDF and sampling from a uniform distribution with a pseudo-random number generator $\xi \sim U(0, 1)$

$$\begin{aligned}
 d_{\text{collision}} &= -\frac{1}{\Sigma_t} \ln(1 - \xi) \\
 &= -\frac{1}{\Sigma_t} \ln(\xi).
 \end{aligned} \tag{2.153}$$

2. *surface crossing*: The particle passes through a homogenous region without experiencing a collision and passes through one of the region's bounding surfaces. The distance to the nearest surface can be computed using the characteristic equation of each surface given the particle's given position (see Fig. 2.8) and accepting the minimum returned value.

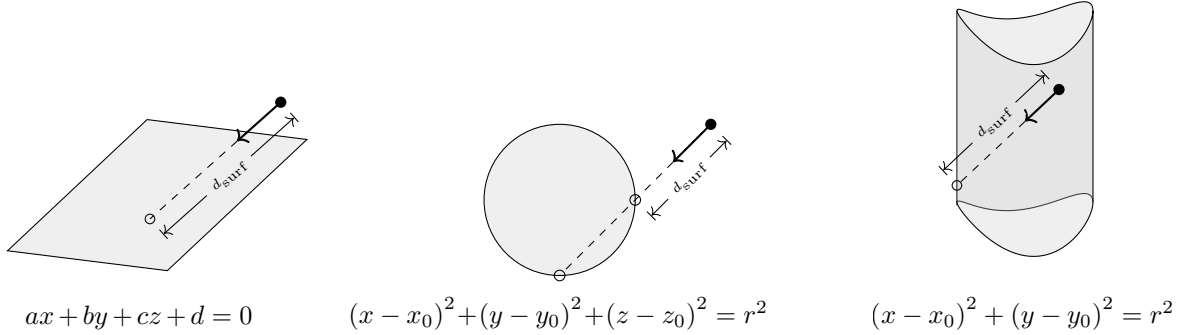


Figure 2.8: Distances to plane (left), sphere (center), and z-aligned cylinder (right). The characteristic equation of each surface is also shown below the surface.

If the distance to the nearest surface is less than the sampled distance to collision, then the particle advances along its current trajectory to just beyond the nearest surface with a superficial nudge to avoid numerical precision issues. If the nearest surface was the boundary of the system, the particle history is terminated. Otherwise, the calculations for the distances to the next surface crossing and collision events are repeated. If instead the distance to collision is less than the distance to the nearest surface, then the colliding

nuclide and specific interaction must be sampled. The colliding nuclide can be sampled by another uniform random variable $\xi \sim U(0, 1)$. The PDF used to govern the sampling is constructed as a histogram, where there is one bin for every nuclide in a homogenized material and the height of each bin corresponds to ratio of the nuclide's total cross section to the material's total cross section. The specific interaction can be sampled in nearly the same way, except now each histogram bin corresponds to one of the constituent reactions possible for the given nuclide and the bin height is the ratio Σ_x/Σ_t for $x \in (\text{scatter, capture, fission, etc.})$.

Of all the possible interactions to be sampled, capture is the easiest to handle; the particle history is simply terminated. In practice though, capturing reactions are typically artificially suppressed for computational efficiency through implicit scattering. Sometimes reactions can release secondary particles, such as (n,2n), (n,3n), etc. For this discussion, these types of secondary particles are not considered. Scattering reactions are more complicated to handle, as neutron scattering can be further subdivided into distinct types, including elastic scattering, inelastic scattering, resonance scattering, and more. To handle these distinct cases, several *scattering laws* have been defined to apply the appropriate physics when sampling the new energy and direction of the emerging particle. A detailed description of their implementation can be found in [84]. Fission reactions are handled specially. For each fission event, an average of $\bar{\nu}$ fission neutron are released. Their angular distribution is sampled isotropically, and their energy distribution is sampled from the appropriate fission spectrum $(\chi, \chi_p, \chi_{d,j})$. Each of the resulting fission neutrons, or “progeny”, need to be saved so that their random walks can be simulated after all of the initial source particles have been tracked to termination. If a particle emerges from a collision event without being terminated, the distance to its next event (either another collision or surface crossing) can be sampled in the previously described manner.

2.10.1 Tallies

To report quantities of interest, a region of interest in phase space can be defined such that every time a particle contributes to this region, a response is recorded or “tallied”. In the physical world, one analogous example might be a radiation detector registering a signal every time a particle interacts within its active detection volume. Unlike a deterministic calculation, the objective of an MC calculation is not to solve for the underlying angular flux distribution $\psi(\mathbf{x}, \hat{\Omega}, E, t)$. Rather, the objective is to sample the underlying PDFs of the physical random processes that define neutron transport and extract results from these processes to estimate functionals of the angular flux. These PDFs are functions of the source and physical properties of the system, and are generally unknown a priori. Contributions to the tallies are tracked throughout the random walks to construct empirical score PDFs that will converge towards the true score PDF in the limit of an infinite number of random walks.

There are several different types of quantities of interest and corresponding tallies that can be defined for an MC calculation. A majority of the practical tallies for reactor physics are based on estimating either integrated fluxes, reaction rates, or surface currents. For estimating these quantities, a few scoring mechanisms have been established:

1. *analog tallies*: The scoring mechanism for analog tallies is very intuitive. Every time a tallying condition is met within the specified phase space region of interest, the particle's statistical weight is scored. For example, if the quantity of interest is the total collision rate within volume V , every time a particle collides within V , the particle weight w will be scored. Alternatively, if the quantity of interest is the net current through surface ∂V , then the particle weight will be scored every time a simulated particle crosses the surface. Analog tallies, while intuitive, are typically avoided in practice, as they can have difficult-to-resolve uncertainties for low-probability events. For some applications though, such as computing surface currents or group to group scattering reaction rates, they are often unavoidable.
2. *collision tallies*: The scoring mechanism for collision tallies is a convenient extension of the scoring mechanism for analog tallies and enables flux estimates. First, consider that the collision density within V is defined by $\langle \Sigma_t \psi \rangle$.⁶ The phase space-integrated flux $\langle \psi \rangle$ can then be estimated by scoring $\frac{w}{\Sigma_t}$ at every collision. For reaction rates, the score can be multiplied by the macroscopic cross section for reaction x at every collision, i.e., $\frac{w \Sigma_x}{\Sigma_t}$. Because this tallying mechanism occurs at every collision within V , it is often preferred over analog reaction rate tallies if the reaction is a low probability event.
3. *path-length tallies*: The scoring mechanism for path-length tallies stems from the interpretation of the neutron flux as a path-length rate density. The amount of path-length traversed ℓ by a simulated particle within V is fairly straightforward to keep track of in an MC calculation. The flux contribution is simply scored as $w\ell$. Reaction rate contributions can be scored by multiplying by the macroscopic cross section for reaction x , i.e., $w\Sigma_x\ell$. This scoring mechanism allows every particle traversing through V to contribute to the tally regardless of whether or not it collides within V .

MC tallies are subject to the statistical noise associated with the stochastic particle transport process. To quantify the uncertainty in MC tally estimates, first let N_h be the number of particle histories (or ‘walks’ or ‘trials’) tracked from birth to termination for a given simulation. Additionally, let s_h be the total score that history h contributes to the tally. The sample mean of the tally μ_{MC} is computed as

$$\mu_{MC} = \frac{1}{N_h} \sum_{h=1}^{N_h} s_h, \quad (2.154)$$

⁶Recall that the angle brackets denote integration over the spatial, angular and energy domains of phase space.

Within each random walk, a particle might contribute to a tally more than once. For example, a particle might cross a surface of interest, collide within a region of interest, or contribute track length to a region of interest more than once within the same history. The score for every individual random walk is then the sum of contributions c_i over all i individual contributions

$$s_h = \sum_{i=1}^{N_c} c_i . \quad (2.155)$$

From statistical analysis, the corresponding sample variance is computed as

$$\sigma_{\text{MC}}^2 = \frac{1}{N_h - 1} \sum_{w=1}^{N_h} s_w^2 - \mu_{\text{MC}}^2 . \quad (2.156)$$

From the Weak Law of Large Numbers, the sample mean and sample variance will converge towards the population mean and population variance as then number of histories is increased. From the Central Limit Theorem, the properly-normalized sum of independent and identically-distributed random trials will tend towards a normal distribution. It can be shown that the sample variance of the sample mean $\sigma_{\mu, \text{MC}}$ is given by

$$\sigma_{\mu, \text{MC}}^2 = \frac{\sigma_{\text{MC}}^2}{N} . \quad (2.157)$$

Thus, the standard deviation (uncertainty) of the MC tally estimate decreases at a rate of $1/\sqrt{N}$.

2.10.1.1 Generating Multigroup Cross Sections

As mentioned previously, one of the primary advantages of the MC method is that it does not rely on discretization of the particle phase space. This is particularly significant for the energy domain, where the use of multigroup cross sections is one of the most consequential sources of discretization error in deterministic solvers. For hybrid methods though, MC solvers can be leveraged to generate multigroup cross sections to be used as input for deterministic solvers. Recall from Section 2.6.1, that the multigroup cross section for some reaction x is defined as the flux-weighted average of the energy-dependent cross section over the range $E_g \leq E < E_{g-1}$

$$\Sigma_{x,g}(\mathbf{x}, t) = \frac{\int_{E_g}^{E_{g-1}} \Sigma_x(\mathbf{x}, E, t) \Psi(\mathbf{x}, E) \, dE}{\int_{E_g}^{E_{g-1}} \Psi(\mathbf{x}, E) \, dE} . \quad (2.158)$$

By integrating over some region of interest, V , multigroup cross sections can be computed as the ratio of integrated quantities, i.e., MC tallies

$$\Sigma_{x,g}(t) = \frac{\int_V \int_{E_g}^{E_{g-1}} \Sigma_x(\mathbf{x}, E, t) \Psi(\mathbf{x}, E) dE dV}{\int_V \int_{E_g}^{E_{g-1}} \Psi(\mathbf{x}, E) dE dV}. \quad (2.159)$$

The MC TML implementation described in Chapter 3 uses MC tallies to generate multigroup cross sections that are used as input for a deterministic CMFD solver.

2.10.2 Criticality Calculations

Criticality, or eigenvalue, calculations present unique challenges (compared to fixed-source calculations) in processing neutron fission chains. For example, if the simulated system is in a supercritical state, then the neutron population could potentially exponentially grow with individual fission chains that never terminate. Conversely, subcritical configurations would have the neutron population eventually die out. This issue is addressed by the *power iteration*, or *fission source iteration* method

$$\mathbf{M}\psi^{l+1}(\mathbf{x}, \hat{\Omega}, E) = \frac{1}{k_{\text{eff}}^l} \mathbf{F}\psi^l(\mathbf{x}, \hat{\Omega}, E), \quad (2.160)$$

where l denotes the fission source iteration (or ‘generation’) index. As an iterative scheme, the user first specifies an initial guess for the fission source distribution and multiplication factor k_{eff} . The fission source particles are tracked to termination at the current iteration and all of their resulting progeny are “banked” to be used as the fission source for the next iteration. The eigenvalue k_{eff}^{l+1} is estimated during iteration l via collision or track length tallies (or a combination of both). Neutron production is always scaled by the multiplication factor to prevent growth or decay of the fission source. Since all generations must contribute equally to the final converged k_{eff} estimate, the total fission source weight W is typically held constant across generations. This means that after every iteration, the starting source weight of each fission source particle must be renormalized such that the total fission source weight is preserved.

It should be noted that, because the fission source distribution is not known a priori, any tally contribution collected before the fission source has converged to the fundamental mode would be biased towards the user-provided initial guess. While the convergence can be assessed by monitoring the Shannon entropy of the fission distribution, in practice, the user typically specifies a fixed number of iteration cycles, colloquially called *inactive cycles*, where tally information is not collected. After these cycles have concluded, *active cycles* begin, and eigenvalue estimates are collected.

2.10.3 Non-analog Monte Carlo

As mentioned previously, stochastic processes are limited by the slow convergence of their uncertainties. For many applications in the physical world, transporting sufficient numbers of particles to resolve these uncertainties is not an issue. Unfortunately, transporting MC particles with computers is much more time consuming than transporting physical particles and the number of MC particles that can reasonably be simulated is often several orders of magnitude smaller than what occurs in physical world radiation environments. Under the constraint of one MC particle corresponding to one physical particle, this could lead to unacceptably high variances if there are insufficient interactions in a region of interest. To overcome this constraint, the concept of particle statistical weight w is introduced, which can be thought of as the number of physical world particles that an MC particle represents. In the physical world, each transported particle would have its own random walk sequence. In the MC simulation, all of the w particles represented by the MC particle have the same random walk. Despite this not being an exact simulation, statistical moments are preserved by multiplying tally contributions by the particle weight.

Even with the concept statistical weight, MC simulations sampling PDFs defined by empirical nuclear data and first principle models can still be inefficient if they spend significant time transport particles in regions of phase space that do not contribute to user-defined tallies. These types of simulations are called *analog* MC simulations. While analog MC simulations are perfectly acceptable when MC particles are contributing to regions of interest, they become inefficient when computational time is spent transporting particles that do not contribute to these regions. Several non-analog techniques have been established to address these inefficiencies. Some of these techniques include biasing particles towards regions of phase space that are of interest to the user. Other techniques are used as a mechanism of population control to artificially increase or decrease the number of particles either globally or in specified regions of phase space. Most of these techniques are not discussed in this dissertation, but are well documented [43], [84]. For this dissertation, particle *splitting* and *rouletting* are the only non-analog techniques prominently used. Particle splitting and rouletting are weight control techniques that prevent particle weight from becoming too large (which can lead to increased tally variance) or too low (which can lead to inefficient history tracking).

The rouletting procedure begins when a particle's weight falls below a specified threshold. When this occurs, a random number $\xi \sim U(0, 1)$ is sampled and compared against an input number Θ , which is typically between two and ten. If $\xi > \frac{1}{\Theta}$, then the particle is terminated. Otherwise, the particle weight is multiplied by a factor of Θ so as to not introduce bias. In general, rouletting increases the variance of the MC result but decreases the time spent transport a particle that does not contribute to the tally result. In a converse

process, the splitting procedure begins when a particle’s weight exceeds a specified threshold. Typically, a predetermined number Θ copies of the particle are artificially introduced to the simulation. Each particle copy then proceeds with a unique random walk and weight reduced by a factor of $\frac{1}{\Theta}$ to preserve the initial particle’s weight. Splitting generally decreases the variance of the result but increases the overall simulation time by increasing the number of random walks that must be simulated.

2.11 Exnihilo Code Suite

A majority of the transient calculations presented in this work are performed with Exnihilo, which is a suite of code libraries for transport application development on high-performance computation platforms. Exnihilo is written to be object-oriented in C++ and leverages existing software libraries such as Trilinos [85] (linear algebra solvers), SCALE [86] (reactor physics, criticality safety, radiation shielding, and spent fuel characterization), and HDF5 [87] (high-performance data reading, writing, and processing). Within Exnihilo, there are three main packages used for this work:

1. *Denovo* [88]: A library of deterministic solvers capable of solving the S_N , SP_N , or MOC equations on a distributed Cartesian mesh. It supports both fixed-source and eigenvalue (with optional CMFD acceleration) calculations as well as forward and adjoint modes. Notably, the only adjoint eigenvalue calculation mode currently available in Denovo uses the SP_3 solver. It features parallel decomposition across both the spatial and energy group domains, enabling scaling to $\mathcal{O}(1e5)$ cores. Finally, it also features some GPU-accelerated implementations, though none of these features were enabled for the hybrid calculations performed in this work.
2. *Shift* [22]: A flexible, high-performance Monte Carlo radiation transport framework that supports both multigroup and continuous-energy physics. It is designed to be “geometry-agnostic”, and supports several different geometry specifications, including those of SCALE [86], MCNP [84], and DAGMC (Computer-Aided Design (CAD)-based). It is also capable of both fixed-source and eigenvalue calculations and is integrated with Denovo for hybrid methods (for example, the CADIS [53] and FW-CADIS [54] variance reduction schemes). It also features a high-performance object-oriented tallying infrastructure, making the implementation of new tallies fairly straight-forward.
3. *Insilico* [89]: A transport-depletion interface that enables in-memory coupling between Shift/Denovo and the ORIGEN depletion package from SCALE. It supports several coupling methods, including fully-explicit, multi-step, and predictor-corrector schemes. It also supports advanced power-normalization methods for accurate constant-power depletion.

2.12 Summary

In this chapter, a heuristic derivation of the time-dependent neutron transport equation was presented. The adjoint neutron transport equation and its relevance as an importance weighting function was also discussed. Several special forms and approximations of the neutron transport equation were derived that lay the foundation for the multi-level implementation that will be discussed in Chapter 3. Of these, the EPKEs and CMFD equations play key roles as levels of the hierarchical TML to enable transient MC calculations in Shift. Finally, deterministic and Monte Carlo methods for solving the neutron transport equation were discussed and a brief summary of the production-level solvers used throughout this work was given. The next chapter discusses how these methods are woven together so that their individual strengths are leveraged to enable high-fidelity time-dependent reactor analyses.

Chapter 3

Transient Multi-Level Method for Monte Carlo

In this chapter, the state of the art multi-level methods for solving the time-dependent neutron transport equations are discussed. Elements from two of these methods in particular are leveraged to enable transient calculations in the production-level Monte Carlo (MC) transport solver, Shift [22]. The first of these two methods is the Transient Multi-Level (TML) method, first implemented by Zhu et al. [13] and later extended by Shen et al. [14] in the Michigan Parallel Characteristics Transport (MPACT) deterministic solver. The TML uses nested Predictor-Corrector Quasi-static Method (PCQM) factorizations to offload much of the computational burden from the transport-level solver to lower-order Coarse Mesh Finite Difference (CMFD) and Exact Point Kinetics Equations (EPKE) solvers to propagate the solution forward in time. The second is a MC implementation of the PCQM implemented by Jo et al. in the McBOX research code [18]. The MC implementation modifies a static fission source iteration solver by introducing four additional particle source banks to propagate the solution forward in time and a CMFD solver for fission source acceleration (not for time propagation between MC solves).

The transient capabilities enabled in Shift reduce the number of total particle source banks required by Jo's method from five to two (one passed between fission source iterations, and one passed between time steps). They also represent the first MC application of Zhu's TML method. The implementation of these combined capabilities in Shift is very invasive and features significant modifications to the Shift code base. Verification of the TML implementation in Shift against the TML implementation in MPACT is presented in this chapter for prescribed transients from the C5G7-TD3 multigroup benchmark. Finally, a prompt feedback-dominated transient for a Dual-Purpose Canister (DPC) model is presented to demonstrate progress towards the transient criticality capabilities cited as motivation for this dissertation in Chapter 1.

3.1 Quasi-Static Methods

Early on in the field of reactor kinetics, computational hardware limited the analysis of non-equilibrium behavior of time-dependent systems to approximate point kinetics, where the time-independent distribution of the neutron flux was used to approximate point kinetics parameters. This *adiabatic approximation* is formulated by a separation of variables into a time-dependent amplitude function and a time-independent shape function

$$\psi(\mathbf{x}, \hat{\Omega}, E, t) = p(t) \Psi(\mathbf{x}, \hat{\Omega}, E) . \quad (3.1)$$

While the adiabatic approximation was reasonably-justified for tightly-coupled systems (where the flux shape does not change throughout the transient), many members of the reactor physics community questioned its validity for loosely-coupled systems.

To remedy the issues of loosely-coupled systems and computational limitations, Henry [3] introduced time-dependence to the shape function and made a key assumption: the shape function varies much more slowly in time and to a lesser degree than the amplitude function. The time derivative of the neutron flux is expanded using the chain rule as

$$\begin{aligned} \frac{\partial \psi}{\partial t}(\mathbf{x}, \hat{\Omega}, E, t) &= \frac{\partial}{\partial t} \left(p(t) \Psi(\mathbf{x}, \hat{\Omega}, E, t) \right) \\ &= \frac{dp(t)}{dt} \Psi(\mathbf{x}, \hat{\Omega}, E, t) + p(t) \frac{\partial \Psi}{\partial t}(\mathbf{x}, \hat{\Omega}, E, t) . \end{aligned} \quad (3.2)$$

Henry's assumption that the shape and amplitude functions change at different rates allows them to be updated on different time scales, as seen in Fig. 3.1. Today, methods built upon this formulation are called *quasi-static* methods [90]. The traditional quasi-static method makes an additional assumption that the time rate of change of the shape function is effectively zero

$$\frac{\partial \Psi}{\partial t}(\mathbf{x}, \hat{\Omega}, E, t) \approx 0 . \quad (3.3)$$

This reduces the shape function solution to be essentially static, where only cross sections and nuclear data depend on time. Changes in the shape function are therefore represented as a prompt jump between reactivity insertions. For slow transients, this is a reasonable approximation. For fast transients, improved methods need to be considered.

Ott and Meneley [6] introduced the Improved Quasi-Static Method (IQM) in 1969 and approximated the

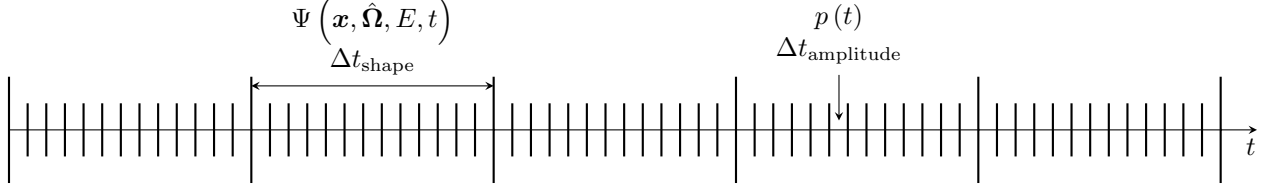


Figure 3.1: Quasi-static time discretization scheme.

time derivative of the shape function to the quasi-static formulation as an implicit Euler discretization

$$\frac{\partial \Psi}{\partial t}(\mathbf{x}, \hat{\Omega}, E, t) \simeq \frac{1}{t^n - t^{n-1}} \left[\Psi(\mathbf{x}, \hat{\Omega}, E, t^n) - \Psi(\mathbf{x}, \hat{\Omega}, E, t^{n-1}) \right]. \quad (3.4)$$

Whereas the quasi-static method ensured uniqueness with the condition that the time derivative of the flux is zero, the IQM ensures uniqueness by imposing a normalization condition

$$c \equiv \left\langle W(\mathbf{x}, \hat{\Omega}, E), \frac{1}{v(E)} \psi^0 \right\rangle, \quad (3.5)$$

where c is the normalization constant. After each shape function evaluation, the normalization condition needs to be checked and iterated on until convergence is determined. To determine convergence, the error in the shape function is computed as

$$\epsilon_{\Psi} = \frac{\left| \left\langle W(\mathbf{x}, \hat{\Omega}, E), \frac{1}{v(E)} \Psi \right\rangle - c \right|}{c}, \quad (3.6)$$

where $W(\mathbf{x}, \hat{\Omega}, E)$ is a weighting function often chosen to be the solution to the initial state adjoint problem, $W(\mathbf{x}, \hat{\Omega}, E) = \psi^{\dagger,0}(\mathbf{x}, \hat{\Omega}, E)$. The general iterative scheme for solving the IQM are described in Algorithm 3.1. To begin, the forward and adjoint steady-state transport fluxes are evaluated at $t = 0$. These two solutions are used to define the normalization constant c with Eq. (3.5). Next, the amplitude function is evaluated on the fine time scale and used to evaluate the shape function with Eq. (2.127) after substituting ψ with Eq. (3.1). Once the shape and amplitude functions have been evaluated on the coarse time step, the error ϵ_{Ψ} is assessed with Eq. (3.6). If the error is not less than some predetermined tolerance, the shape function is evaluated again until the error converges.

3.1.1 Predictor Corrector Quasi-Static Method

An alternative approach to the handling the time derivative of the shape function uses a predictor corrector scheme and was first proposed by Gehin in 1992 [7] as the PCQM. While the PCQM did not receive significant

Algorithm 3.1 Improved quasi-static method iteration scheme.

```

1: Solve for steady-state forward flux  $\psi^0$ ; Solve for steady-state adjoint flux  $\psi^{\dagger,0}$ ;
2:  $c \leftarrow$  Eq. (3.5)
3: for all  $1 \leq n \leq N_{\text{shape}}$  do
4:    $\epsilon \leftarrow \inf$ 
5:   while  $\epsilon > \text{tolerance}$  do
6:     Evaluate point kinetics parameters up until  $\Delta t_{\text{shape}}^n$ 
7:      $p^n \leftarrow$  Eq. (2.55)
8:      $\Psi^n \leftarrow$  Eq. (2.127) after substituting  $\psi$  with Eq. (3.1)
9:      $\psi^n = p^n \Psi^n$ 
10:     $\epsilon_{\Psi} \leftarrow$  Eq. (3.6)
11:   end while
12: end for

```

attention upon its initial publication, it has gained popularity since Dulla revisited the method in 2008 [9]. The PCQM also relies on the assumption that the shape function varies slowly in time but instead of updating the shape function on the macro time scale, it begins by updating the entire neutron flux ψ (see Fig. 3.2).

This initial flux solution is denoted the “predictor” flux ψ^P , and will have errors associated with integrating over a large time step. The algorithm proceeds by updating the amplitude function on the micro time scale to “correct” the flux estimate

$$\psi^{c,n} = \frac{c}{\left\langle W(\mathbf{x}, \hat{\Omega}, E), \frac{1}{v(E)} \psi^{P,n} \right\rangle} p^n \psi^{P,n}. \quad (3.7)$$

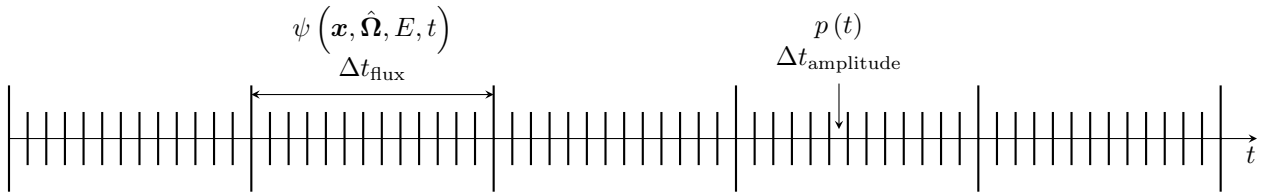


Figure 3.2: PCQM time discretization scheme.

Algorithm 3.2 Predictor corrector quasi-static method iteration scheme.

```

1: Solve forward steady-state forward flux  $\psi^0$ ; Solve for steady-state adjoint flux  $\psi^{\dagger,0}$ ;
2:  $c \leftarrow$  Eq. (3.5)
3: for all  $1 \leq n \leq N_{\text{shape}}$  do
4:    $\psi^{P,n} \leftarrow$  Eq. (2.127)
5:   Interpolate point kinetics parameters between  $\psi^{n-1}$  and  $\psi^n$  using Eqs. (2.54)
6:    $p^n \leftarrow$  Eq. (2.55)
7:    $\psi^{c,n} \leftarrow$  Eq. (3.7)
8: end for

```

The main benefit of the PCQM algorithm is that it eliminates the inner iteration required by the IQM, seen in Fig. 3.3. The inner shape convergence loop can be expensive if the shape function changes significantly

throughout the transient. However, there is a tradeoff, as the PCQM predictor evaluation of the transport solution on the coarse time scale is typically more expensive than that of the shape evaluation required by the IQM. For the IQM, if the shape function converges quickly, the algorithm is more efficient than that of the PCQM. The PCQM is therefore better-suited for transients where the shape function is difficult to resolve.

As a consequence of the PCQM being linear, there is no way to improve the flux solution after the predictor flux has been corrected. This makes implementing the PCQM with feedback mechanisms more difficult than with the IQM, and sometimes calls for the introduction of an intermediate time discretization between the predictor and amplitude discretizations to update feedback calculations for multiphysics coupling [12].

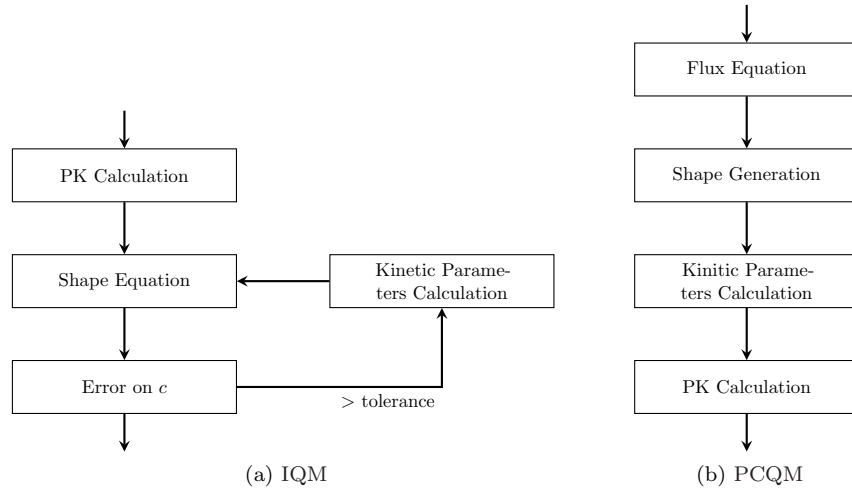


Figure 3.3: Comparison of the IQM and PCQM algorithms.

3.2 Transient Multi-Level Method

The TML method [13] extends the predictor-corrector quasi-static method by introducing an intermediate nested time discretization level, shown in Fig. 3.4. As with the PCQM, the angular flux is computed with a high-fidelity solver and factorized into the product of amplitude and shape functions. The amplitude function $p_{\text{transport}}$ is now the solution of the CMFD equations discussed in Sec. 2.8 and is used to correct the transport-level predictor flux $\psi_{\text{transport}}^{\text{P}}$. The CMFD solution is further factorized into the product of another shape and amplitude function $\phi_{\text{CMFD}}(\mathbf{x}, E, t) = \Psi(\mathbf{x}, E, t) p_{\text{CMFD}}$, where this time the amplitude, p_{CMFD} , is the solution of the EPKEs discussed in Sec. 2.4 and used to update the CMFD-level predictor flux $\phi_{\text{CMFD}}^{\text{P}}$.

The main benefit of adding the extra CMFD level is that it allows the spatial and energy dependence of the shape function to be updated in between MC time steps that would not otherwise be captured by the

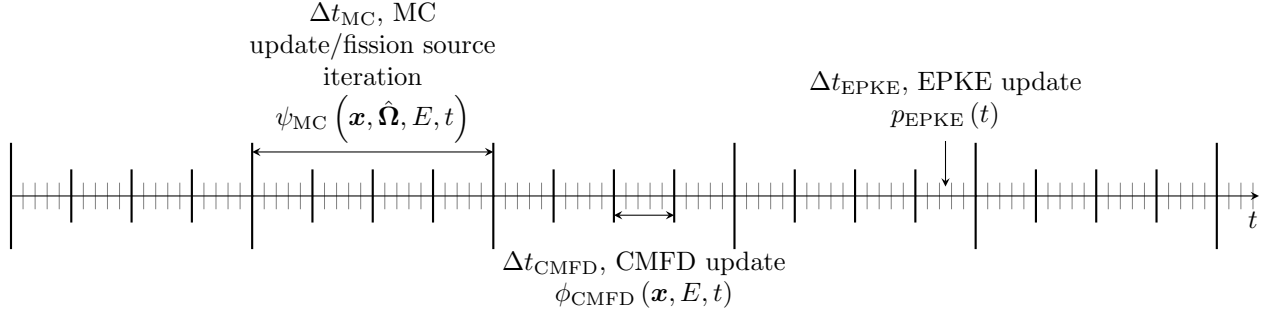


Figure 3.4: TML time stepping scheme [13].

globally-integrated point kinetics equations. The CMFD level was also shown to allow for an increase in the transport-level macro time step, improving the overall speed of the transient calculation.

3.2.1 Transient Monte Carlo Solver

Hackemack et al. showed that MC implementations of the Transient Fixed Source Problem (TFSP) given by Eq. (2.127) can result in prohibitively long individual particle histories for large positive reactivity insertions [17]. This is because fixed-source algorithms typically cycle through the entire fission chain until an individual history terminates. One way to fix the issue of long non-terminating fission chains is to implement particle rouletting for low-weight particles.

Alternatively, Jo et al. proposed that Eq. (2.127) be solved as a modified fission source iteration problem [18]. The fission source iteration algorithm only banks the subsequent generation of fission particles at each iteration, effectively guaranteeing the termination of each history. Additionally, a fission source iteration could help avoid numerical instability by allowing a greater chance for the neutron flux distribution to become balanced before advancing to the next time step. For these reasons, the same tactic is also used in this work. To accommodate these changes, a fission source iteration index, l , is applied to all operators applied at time t^n in Eq. (2.127) to arrive at

$$\begin{aligned}
 \left(M + \frac{1}{v(E) \Delta t^n} \right) \psi^{l+1,n}(\mathbf{x}, \hat{\Omega}, E) &= \frac{\psi^{n-1}(\mathbf{x}, \hat{\Omega}, E)}{v(E) \Delta t^n} + \mathbf{F}_p \psi^{l,n}(\mathbf{x}, \hat{\Omega}, E) \\
 &+ \sum_{j=1}^6 \mathbf{S}_{d,j}^{n-1}(\mathbf{x}, \hat{\Omega}, E) f_{1,j} \\
 &+ \sum_{j=1}^6 \mathbf{F}_{d,j} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) f_{2,j} + \sum_{j=1}^6 \mathbf{F}_{d,j} \psi^{l,n}(\mathbf{x}, \hat{\Omega}, E) f_{3,j}, \quad (3.8)
 \end{aligned}$$

where

$$f_{1,j} = e^{-\lambda_j \Delta t^n} , \quad (3.9a)$$

$$f_{2,j} = \frac{1 - e^{-\lambda_j \Delta t^n} - \lambda_j \Delta t^n e^{-\lambda_j \Delta t^n}}{\lambda_j \Delta t^n} , \quad (3.9b)$$

$$f_{3,j} = \frac{e^{-\lambda_j \Delta t^n} (1 - e^{\lambda_j \Delta t^n} + \lambda_j \Delta t^n e^{\lambda_j \Delta t^n})}{\lambda_j \Delta t^n} . \quad (3.9c)$$

The first term on the right-hand side of Eq. (3.8) accounts for “census” neutrons that are still in flight at the end of the simulated MC time step. Their particle state information must be passed forward to the next MC time step simulation. The second term corresponds to prompt neutrons, which are already present in static fission source iteration calculations. Their particle state information is only passed between fission source iterations. The third term corresponds to an accumulation of delayed neutron precursors when integrating over the time step Δt^n . Finally, the fourth and fifth terms correspond to delayed neutrons born of delayed neutron precursors that decay during Δt^n . These delayed neutrons are numerically weighted as coming from the beginning t^{n-1} and end t^n of the time step, respectively.

A total of $L = L_{\text{inactive}} + L_{\text{active}}$ fission source iterations are computed for each time step, where L_{inactive} and L_{active} are the number of inactive and active cycles, respectively, and set by the user. Traditional criticality solvers solve the steady-state equation

$$\mathbf{M}\psi^{l+1,n}(\mathbf{x}, \hat{\Omega}, E) = \mathbf{F}\psi^{l,n}(\mathbf{x}, \hat{\Omega}, E) , \quad (3.10)$$

where all operators are defined for a single point in time, t^n , and the right-hand side contains only a single source term communicated between fission source iterations that corresponds to both prompt and delayed fission neutrons. On the other hand, the transient criticality formulation from Eq. (3.8) has five distinct

source terms, some of which are defined at t^n and some of which are defined at t^{n-1}

$$\begin{aligned}
\left(M + \frac{1}{v(E) \Delta t^n}\right) \psi^{l+1,n}(\mathbf{x}, \hat{\Omega}, E) = & \underbrace{\frac{\psi^{n-1}(\mathbf{x}, \hat{\Omega}, E)}{v(E) \Delta t^n}}_{\textcircled{1}} + \underbrace{\mathbf{F}_p \psi^{l,n}(\mathbf{x}, \hat{\Omega}, E)}_{\textcircled{2}} \\
& + \underbrace{\sum_{j=1}^J \mathbf{S}_{d,j} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) f_{1,j}}_{\textcircled{3}} \\
& + \underbrace{\sum_{j=1}^J \mathbf{F}_{d,j} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) f_{2,j}}_{\textcircled{4}} + \underbrace{\sum_{j=1}^J \mathbf{F}_{d,j} \psi^{l,n}(\mathbf{x}, \hat{\Omega}, E) f_{3,j}}_{\textcircled{5}}. \quad (3.11)
\end{aligned}$$

3.2.1.1 Shift Implementation

For this work, an implementation of Eq. (3.11) was accomplished in Shift, an MC solver developed and maintained at Oak Ridge National Laboratory (ORNL). The implementation exists on a static research branch that diverged from main development branch in March of 2020.

The left-hand side of Eq. (3.11) is implemented by augmenting the total cross section at the beginning of a random walk by the factor $\frac{1}{v \Delta t^n}$. This augmentation accounts for the sampling of an additional “time census” pseudo-absorption event when performing the distance-to-collision calculation. When interpreted this way as an additional absorption interaction, the distance to the nearest collision is sampled with random number ξ as

$$d_{\text{collision}} = -\frac{1}{\Sigma_t + \frac{1}{v \Delta t^n}} \ln(\xi). \quad (3.12)$$

The implementation of the source terms on the right-hand side of Eq. (3.11) uses two separate particle source banks: one passed between fission source iterations (terms $\textcircled{2}$ and $\textcircled{5}$), and one passed between time indices (terms $\textcircled{1}$, $\textcircled{3}$, and $\textcircled{4}$). To decrease computational time, all five source terms are sampled implicitly at all collisions by pushing a copy of the incident particle to its respective source bank with an appropriate weighting factor. The individual source terms and their corresponding weight modifications are:

- $\textcircled{1}$ Neutrons from time t^{n-1} that were still in flight by the time t^n . These “time census” neutrons are implicitly captured at every collision during active cycles. This is done by passing a copy of the incident particle’s state information (position, direction, energy, and weight) at every collision to the transient time-stepping bank. In order to ensure an unbiased result, the banked particle state requires a weight modification that corresponds to the probability that a time-absorption event is sampled at the collision

site

$$w_{\text{banked}} = w_{\text{incident}} \left(\frac{\frac{1}{v(E)\Delta t^n}}{\Sigma_t^n(\mathbf{x}, E) + \frac{1}{v(E)\Delta t^n}} \right), \quad (3.13a)$$

where w_{incident} is the incident weight of the particle prior to experiencing a collision, and w_{banked} is the weight that is passed to the time stepping bank data structure. The incident particle will then proceed through the MC algorithm and collision mechanics will be processed with a weight adjustment factor corresponding to the compliment of Eq. (3.13a) as

$$w_{\text{collision}} = w_{\text{incident}} \left(\frac{\Sigma_t^n(\mathbf{x}, E)}{\Sigma_t^n(\mathbf{x}, E) + \frac{1}{v(E)\Delta t^n}} \right). \quad (3.13b)$$

For very small time steps the weight of the particles exiting a time-absorption event could be very small if $\frac{1}{v\Delta t} \ll \Sigma_t$. Therefore, the particle rouletting procedure is performed upon exiting a collision.

- ② Neutrons resulting from prompt fission. This bank is already handled in steady-state k -eigenvalue calculations and therefore requires no modification to the solver. If a prompt neutron is sampled, its state information is banked at the fission site to be carried over the next fission source iteration and used to source a new particle history. Because these neutrons are emitted on very small time scales, their state information is not passed to subsequent time steps.
- ③ This is a special term that accounts for the accumulation of delayed neutron precursors when integrating over previous time steps. These accumulated delayed neutron precursors will emit delayed neutrons at time scales that might exceed Δt^n . In the Shift implementation, copies of all of the particle state information in the time step iteration bank at time index $n - 1$ are superficially injected into the time step iteration bank at n with weight modification

$$w_{t^n} = w_{t^{n-1}} f_{1,j}. \quad (3.13c)$$

- ④ Neutrons resulting from delayed neutron precursors that were accumulated over the time step, Δt^n , and numerically weighted as being sourced from the previous time index, $n - 1$, with a weight modification of

$$w_{\text{banked}} = w_{\text{incident}} f_{2,j}. \quad (3.13d)$$

- ⑤ Neutrons resulting from delayed neutron precursors that were accumulated over the time step, Δt^n , and

numerically weighted as being sourced from the current time index, n , with a weight modification of

$$w_{\text{banked}} = w_{\text{incident}} f_{3,j} . \quad (3.13\text{e})$$

The particle state information is then copied and pushed to the time-stepping source bank.

A simplified branching process that neglects energy-dependence is shown in Fig. 3.5 to depict the implicit banking of source terms at each collision site.

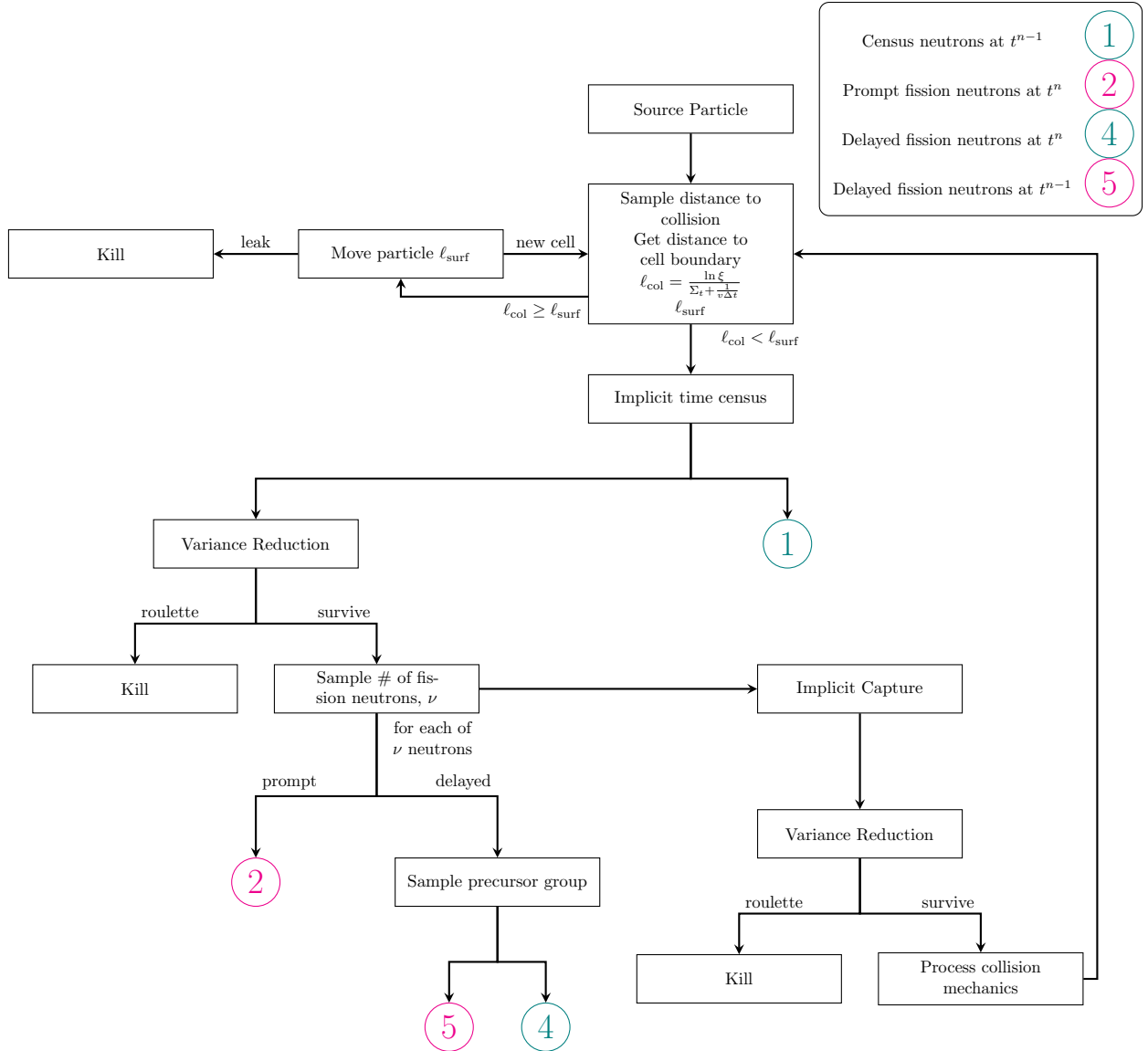


Figure 3.5: Simplified neutron history branching in transient fission source iteration scheme.

Figure 3.6 depicts the transfer of source terms between fission source iterations and time steps. Fission

source terms (2) and (5) are passed between fission source iterations. During the active fission cycles, time step source terms (1) and (4) are tallied and read in as source terms at the start of the next time step. Finally, the accumulated delayed neutron term (3) is read from the previous time step iteration bank at $n - 1$.

The repeated process of transferring delayed neutron source terms from $n - 1$ causes the time step source bank to accumulate terms exponentially. To prevent this exponential growth, a population control technique known as “combing” [91] is performed in between time steps that keeps the size of the bank fixed. Say, for example, that the size of the transient will be fixed to contain the state information of 1×10^5 source particles. Before combing, the weight of all of the particles in the transient source bank will be added together. Next, the order of all of the particles in the source bank will then be randomly shuffled and only the first only the state information for the first 1×10^5 will be retained (the remaining particle state information will be discarded). The weight of all of the remaining 1×10^5 source particles will be added together so that the weight of the remaining particles can be updated as

$$w_{j,\text{combed}} = w_j \frac{\sum_j^{1 \times 10^5} w_j}{\sum_i w_i}, \quad (3.14)$$

where i corresponds to the index of the source particle before shuffling and j corresponds to the index of the same source particle after shuffling.

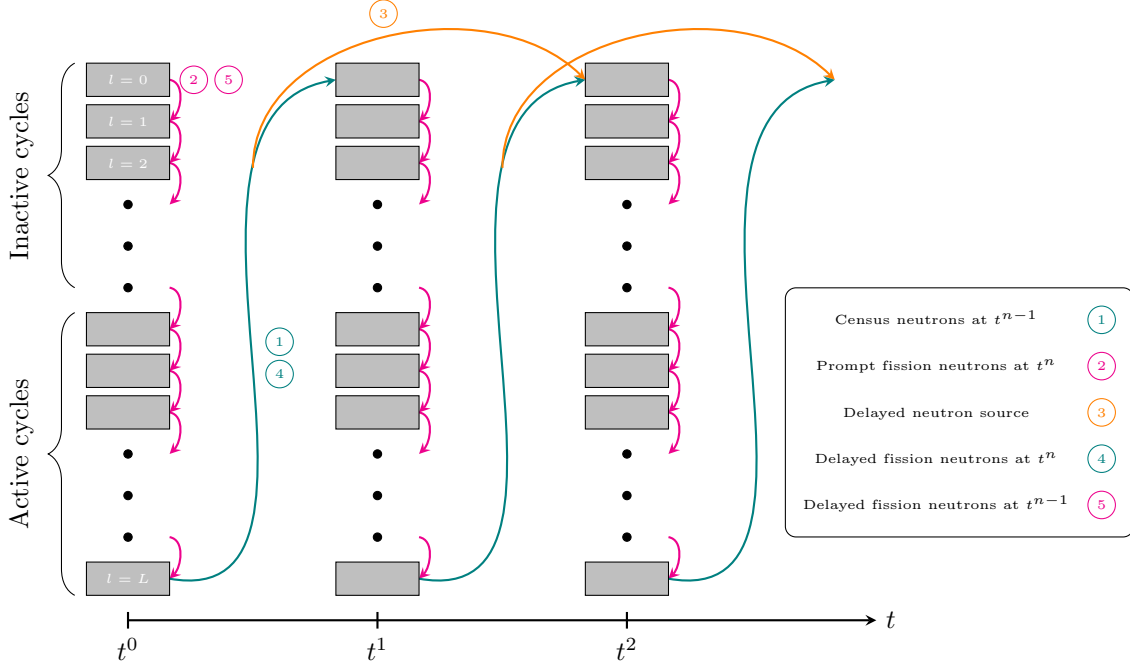


Figure 3.6: Transfer of source terms between fission source iterations and time steps.

The implementation in Shift [22] transfers particles between time steps by tallying particle state information (position, energy, precursor group, and weight) during transport and then writing the state information to an HDF5 file at the end of every MC time step. The state information is then read back into memory at the beginning of the next MC time step as a source term. This is certainly not the most efficient way to transfer the state information in between time steps, as in-memory transfer of particle state information is more efficient than file input/output. The decision to write state information to an HDF5 file was made early on in the transient implementation to make use of the existing surface census tally and source infrastructure that was already implemented in Shift.

3.2.2 Monte Carlo Transport and CMFD Coupling

The PCQM formulation between the MC and CMFD solvers factorizes the angular flux as

$$\psi_{\text{MC}}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) = p_{\text{MC}}(\mathbf{x}, E, t) \Psi_{\text{MC}}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) , \quad (3.15)$$

where $p_{\text{MC}}(\mathbf{x}, E, t)$ is the transport-level amplitude function and $\Psi_{\text{MC}}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t)$ is the transport-level shape function. The shape function is computed

$$\Psi_{\text{MC}}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) = \frac{\psi_{\text{MC}}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t)}{p_{\text{MC}}(\mathbf{x}, E, t)} \approx \frac{\psi_{\text{MC}}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t)}{\phi_{\text{CMFD}}(\mathbf{x}, E, t)} , \quad (3.16)$$

where $\phi_{\text{CMFD}}(\mathbf{x}, E, t)$ is the CMFD scalar flux and is the solution to Eq. (2.147). The time-dependent CMFD solution can be obtained with any standard linear algebra solver after setting up the system of equations as a matrix described by Zhu's Multigroup Matrix (MGM) method (defined in Section 2.8). The CMFD solver in Shift uses the Generalized Minimal Residual (GMRES) method to solve the matrix formulation of Eq. (2.147) where the solution vector consists of the cell-centered CMFD scalar fluxes for each energy group. Uniqueness with the factorization given by Eq. (3.15) is ensured with a normalization constraint for each CMFD cell

$$\frac{1}{V_{\text{CMFD}}} \left\langle \psi_{\text{MC}}(\mathbf{x}, \hat{\boldsymbol{\Omega}}, E, t) \right\rangle = 1 . \quad (3.17)$$

The MC and CMFD coupling algorithm is explicitly defined in Algorithm 3.3. To begin, the forward and adjoint steady state fluxes are evaluated at $t = 0$. Then, the MC solution is advanced on the coarse time step Δt_{MC} by solving Eq. (3.11). On the Δt_{CMFD} time scale, CMFD parameters are interpolated between the two MC solutions so that the CMFD scalar flux amplitude function (and corresponding EPKE substeps) can be propagated forward in time. Finally, the CMFD shape function is used to correct the MC tallies at t^n and

precursor concentrations are computed.

Algorithm 3.3 PCQM coupling between MC and CMFD levels in the TML.

- 1: Solve Eq. (3.10) for forward steady-state forward flux ψ_{MC}^0
 - 2: Solve for steady-state adjoint flux $\psi^{\dagger,0}$;
 - 3: **for all** $1 \leq n \leq N_{\text{MC}}$ **do**
 - 4: $\psi_{\text{MC}}^{\text{p},n} \leftarrow \text{Eq. (3.11)}$
 - 5: Linearly interpolate CMFD volume- and surface-integrated MC tallies between ψ^{n-1} and ψ^n
 - 6: Propagate through Algorithm 3.4 to compute $\phi_{\text{CMFD}}^{\text{p},n}$ and $\phi_{\text{CMFD}}^{\text{c},n}$
 - 7: $\psi_{\text{MC}}^{\text{c},n} \leftarrow \psi_{\text{MC}}^{\text{p},n} \frac{\phi_{\text{CMFD}}^{\text{c},n}}{\phi_{\text{CMFD}}^{\text{p},n}}$
 - 8: Update precursor concentrations using corrected MC flux distribution.
 - 9: **end for**
-

3.2.2.1 Macroscopic Cross Sections

Many of the averaged coarse mesh quantities seen in Eq. (2.141) can be obtained directly from the MC transport solution tallies. To do so, first rewrite Eq. (2.141) as

$$\begin{aligned} \sum_{u \in (x,y,z)} \left(\left\langle \bar{J}_{l+\frac{1}{2},m,n}^{u,g} \Delta_m^v \Delta_n^w \right\rangle - \left\langle \bar{J}_{l-\frac{1}{2},m,n}^{u,g} \Delta_m^v \Delta_n^w \right\rangle \right) + \left\langle \bar{\Sigma}_{t,l,m,n}^g \bar{\phi}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle \\ = \sum_{g'=1}^G \left\langle \bar{\Sigma}_{s,l,m,n}^{g' \rightarrow g} \bar{\phi}_{l,m,n}^{g'} \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle + \bar{\chi}_{l,m,n,g} \sum_{g'=1}^G \left\langle \bar{\nu}_f \bar{\Sigma}_{f,l,m,n}^{g' \rightarrow g} \bar{\phi}_{l,m,n}^{g'} \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle. \end{aligned} \quad (3.18)$$

The bracketed parameters in Eq. (3.18) are scalar quantities and defined as:

- $\left\langle \bar{J}_{l \pm \frac{1}{2},m,n}^{u,g} \Delta_m^v \Delta_n^w \right\rangle$ – surface area-integrated net current over surface $(l \pm \frac{1}{2}, m, n)$ with surface normal in direction u with energy group g . The surface area-averaged net current is computed from this quantity by dividing by the transverse area, $\Delta_m^v \Delta_n^w$.
- $\left\langle \bar{\Sigma}_{t,l,m,n}^g \bar{\phi}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle$ – volume-integrated total reaction rate over energy group g .
- $\left\langle \bar{\Sigma}_{s,l,m,n}^{g' \rightarrow g} \bar{\phi}_{l,m,n}^{g'} \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle$ – volume-integrated scattering production rate of neutrons that begin with energy group g' and exit reaction in group g .
- $\left\langle \bar{\nu}_f \bar{\Sigma}_{f,l,m,n}^{g' \rightarrow g} \bar{\phi}_{l,m,n}^{g'} \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle$ – volume-integrated fission production rate of neutrons from fissions in group g' that exit in group g .

In order to solve for the multigroup fluxes needed by the diffusion model, macroscopic cross sections can be derived by conserving reaction rates from the MC tallies. The total, scattering, and fission macroscopic

cross sections are defined as the ratios

$$\overline{\overline{\Sigma}}_{t,l,m,n}^g \equiv \frac{\langle \overline{\overline{\Sigma}}_{t,l,m,n}^g \overline{\phi}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \rangle}{\langle \overline{\overline{\phi}}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \rangle}, \quad (3.19a)$$

$$\overline{\overline{\Sigma}}_{s,l,m,n}^{g' \rightarrow g} \equiv \frac{\langle \overline{\overline{\Sigma}}_{s,l,m,n}^{g' \rightarrow g} \overline{\phi}_{l,m,n}^{g'} \Delta_l^u \Delta_m^v \Delta_n^w \rangle}{\langle \overline{\overline{\phi}}_{l,m,n}^{g'} \Delta_l^u \Delta_m^v \Delta_n^w \rangle}, \quad (3.19b)$$

$$\overline{\overline{\Sigma}}_{f,l,m,n}^{g' \rightarrow g} \equiv \frac{\langle \overline{\overline{\Sigma}}_{f,l,m,n}^{g' \rightarrow g} \overline{\phi}_{l,m,n}^{g'} \Delta_l^u \Delta_m^v \Delta_n^w \rangle}{\langle \overline{\overline{\phi}}_{l,m,n}^{g'} \Delta_l^u \Delta_m^v \Delta_n^w \rangle}. \quad (3.19c)$$

The diffusion model also requires diffusion coefficients to preserve leakage rates. The definition of the diffusion coefficients can determine the convergence properties of the CMFD equations.

$$\overline{\overline{D}}_{l,m,n}^g = \frac{\langle \overline{\overline{\phi}}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \rangle}{3 \langle \overline{\overline{\Sigma}}_{tr,l,m,n}^g \overline{\overline{\phi}}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \rangle}, \quad (3.20)$$

where

$$\langle \overline{\overline{\Sigma}}_{tr,l,m,n}^g \overline{\overline{\phi}}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \rangle = \langle \overline{\overline{\Sigma}}_{t,l,m,n}^g \overline{\overline{\phi}}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \rangle - \langle \overline{\overline{\Sigma}}_{s1,l,m,n}^g \overline{\overline{\phi}}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \rangle. \quad (3.21)$$

The diffusion coefficient given by Eq. (3.20) requires the computation of a “transport” reaction rate integrated over the CMFD mesh element, i.e., $[\Sigma_{tr}^g \phi^g]$. The transport reaction rate is computed in Eq. (3.21) as the difference between the total reaction rate and the P_1 scattering reaction rate. The P_1 scattering reaction rate is computed using the out-scatter approximation from Eq. (2.111) is used to define the diffusion coefficient as

$$\langle \overline{\overline{\Sigma}}_{s1,l,m,n}^g \overline{\overline{\phi}}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \rangle = \overline{\mu}_0 \sum_{g'=1}^G \langle \overline{\overline{\Sigma}}_{s0,l,m,n}^{g' \rightarrow g} \overline{\overline{\phi}}_{l,m,n}^{g'} \Delta_l^u \Delta_m^v \Delta_n^w \rangle. \quad (3.22)$$

Whereas most of the previous reaction rate definitions used as input for the CMFD solver can be computed via path-length or collision tallies, the group-to-group scattering cross sections defined in Eq. (3.22) must be computed as analog MC tallies at each collision. The group-to-group scattering matrix entry for $g' \rightarrow g$ receives a scoring contribution when a neutron transitions from group g' to group g .

Additionally, the mean scattering cosine, μ_0 must be computed via an analog tally by computing the cosine of the angle between the incident ($\hat{\Omega}'$) and outgoing ($\hat{\Omega}$) directions. Though the use of these analog tallies provides a relatively straightforward process for computing diffusion coefficients, their statistical uncertainties can be difficult to resolve for small spatial CMFD elements. Other, more accurate definitions can be computed

in MC calculations via migration area tallies and other techniques. A detailed description of these more accurate methods can be found in [69]. Finally, the CMFD method is formulated such that limitations in the accuracy of $\overline{\overline{D}}$ will be compensated by the nonlinear coupling term, $\hat{D}$, that was introduced in Eq. (2.144).

3.2.3 CMFD and EPKE Coupling

The PCQM formulation between the CMFD and EPKE solvers factorizes the scalar flux as

$$\phi_{\text{CMFD}}(\mathbf{x}, E, t) = p_{\text{CMFD}}(t) \Psi_{\text{CMFD}}(\mathbf{x}, E, t), \quad (3.23)$$

where $p_{\text{CMFD}}(t) = p_{\text{EPKE}}(t)$. The normalization constraint is the same as that in Eq. (3.5)

$$c = \left\langle \psi^{\dagger,0}(\mathbf{x}, \hat{\Omega}, E), \frac{1}{v(E)} \Psi_{\text{CMFD}}^0(\mathbf{x}, E) \right\rangle = \left\langle \psi^{\dagger,0}(\mathbf{x}, \hat{\Omega}, E), \frac{1}{v(E)} \Psi_{\text{CMFD}}(\mathbf{x}, E, t) \right\rangle. \quad (3.24)$$

The amplitude and shape functions are then computed as

$$\phi_{\text{CMFD}}^{\text{c},n}(\mathbf{x}, E) = \frac{c}{\left\langle \psi^{\dagger,0}(\mathbf{x}, \hat{\Omega}, E), \frac{1}{v(E)} \phi_{\text{CMFD}}^{\text{p},n}(\mathbf{x}, E) \right\rangle} p_{\text{EPKE},n} \phi_{\text{CMFD}}^{\text{p},n}(\mathbf{x}, E). \quad (3.25)$$

The PCQM coupling between the CMFD/EPKE levels is described by Algorithm 3.4. To begin, the multigroup cross sections needed for the CMFD solver are post-processed from the direct MC tallies and interpolated on the CMFD time scale. Next, the normalization constant c is computed with Eq. (3.24) to ensure that the shape/amplitude factorization is unique. Then, the CMFD predictor solution is propagated forward in time. The EPKE parameters are then interpolated on the Δt_{EPKE} time scale and the EPKE amplitude function is propagated forward in time to the end of the CMFD time step. Finally, the CMFD predictor flux is corrected using Eq. (3.25) and the precursor concentrations are computed.

Algorithm 3.4 PCQM coupling between CMFD and EPKE levels in the TML.

- 1: Linearly interpolate multigroup cross sections needed for CMFD solver on Δt_{CMFD} time step
 - 2: $c \leftarrow$ Eq. (3.24)
 - 3: **for all** $1 \leq n \leq \Delta t_{\text{MC}}/\Delta t_{\text{CMFD}}$ **do**
 - 4: $\phi_{\text{CMFD}}^{\text{p},n} \leftarrow$ Eq. (2.147)
 - 5: Linearly interpolate globally-integrated EPKE parameters at Δt_{EPKE} resolution
 - 6: $p_{\text{EPKE}}^n \leftarrow$ Eq. (2.55)
 - 7: $\phi_{\text{CMFD}}^{\text{c},n} \leftarrow$ Eq. (3.25)
 - 8: Update precursor concentrations using corrected CMFD flux distribution.
 - 9: **end for**
-

As a high-level overview, Fig. 3.7 shows the flow of information between the various TML levels. At the beginning and end of each MC time step, multigroup cross sections are passed down to the CMFD solver to

be interpolated and propagated forward through the CMFD time steps. At end of the CMFD propagations, a space- and energy-dependent amplitude function $p_{\text{CMFD}}(\mathbf{x}, E)$ is be passed back to the MC solver to update the weights of the particles in the time-stepping source bank as a function of their current position and energy

$$w^{c,n-1} = w^{p,n-1} \frac{\phi_{\text{CMFD}}^{c,n-1}(\mathbf{x}, E)}{\langle \phi_{\text{CMFD}}^{c,n-1}(\mathbf{x}, E) \rangle}. \quad (3.26)$$

Between the CMFD and EPKE levels, point kinetics parameters are passed from the CMFD calculation. At the end of each EPKE propagation, a globally-integrated amplitude function is passed back up to the CMFD solver to update the scalar flux prediction.

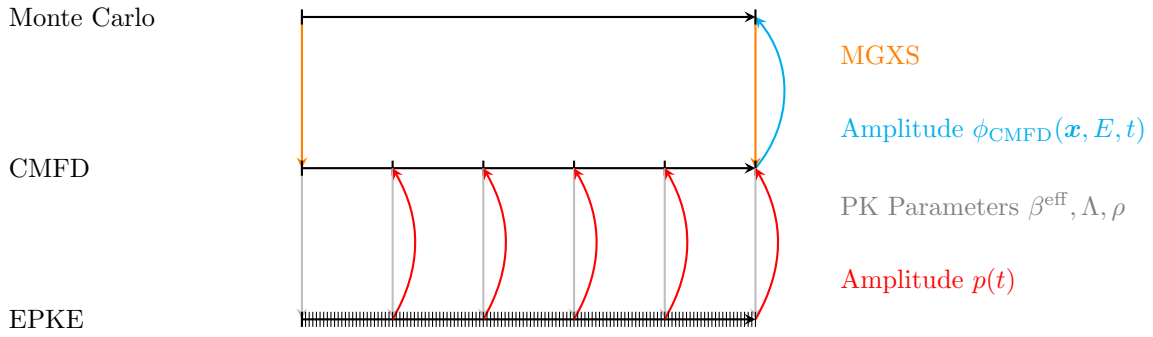


Figure 3.7: Flow of information for coupling between the MC, CMFD, and EPKE levels.

3.2.4 Implementation Overview

The purpose of this section is to provide insight into what needs to be done to implement the TML algorithm into a production solver. The final multigroup implementation in Shift required the modification or creation of well over 100 individual files in nearly every directory of the Shift source code. Below is a list of the major structural changes made to the Shift source code to enable the transient calculations:

- *time-stepping*: Shift’s existing depletion structure, housed in the “Insilico” package and briefly described in Section 2.11, is leveraged to set up a series of MC k -eigenvalue solves. The Shift input file processing is modified to require a list of user-supplied time step sizes. These time step sizes are used to construct data structures for individual fission source iteration solves. In between MC solves, tally updates for time t^n are written to an HDF5 file and post-processed to generate multigroup cross sections for the CMFD solver and adjoint-weighted point kinetics parameters for the EPKE solver. After this post-processing is completed, in-memory tally data is flushed. In the Shift input file, the user also specifies two integers corresponding to the number of CMFD time steps per every MC time step, and

the number of EPKE time steps per every CMFD time steps. In the current implementation these values are fixed throughout the transient and cannot be varied between time steps.

- *transport routine*: Modifications to the transport routine are minimal, but require augmenting the total cross section by a factor of $\frac{1}{v\Delta t_{MC}}$ before a distance to next event is sampled to account for the additional “time absorption” reaction. Implicitly census-capturing neutrons at every collision is then accomplished by passing a copy of a particle’s state information at every collision to the tallying data structure with a weight modified by Eq. (3.13a) before collision mechanics are processed. The surviving particle’s weight is then modified by the complementary factor from Eq. (3.13b) before collision mechanics are processed.
- *sampling of fission sites*: For each of the ν neutrons born at a fission site, three copies of the progenitor state information are made: one corresponding to a prompt fission neutron, one corresponding to a delayed neutron going into the current fission bank at t^n , and one corresponding to a delayed neutron tallied and carried over in the time source bank. Each copy of the particle state receives an appropriate weight adjustment (with either Eq. (3.13d) or Eq. (3.13e)), sampled energy group, and delayed neutron precursor group index (0 for prompt neutrons).
- *transient particle sources*: Shift’s data structure for surface census sources is leveraged to handle transient particle sources. At the beginning of each MC time step, the particle state information (position, direction, and energy) of neutrons that were tallied at census events from the previous time index t^{n-1} is read from an HDF5 file to populate the transient source bank at t^n . This data structure additionally reads in delayed neutron data from t^{n-1} (with appropriate weight modifier $f_{3,j}$) and performs a population control comb to keep the size of the bank from exponentially growing/decaying. The comb is implemented by summing all of the particle weights in the transient source bank and then randomly shuffling the order of the particles in the bank. Then, only the particle information of the first N_{ts} particles is retained, where N_{ts} is a user-specified maximum number of particle allowed in the source bank. Finally, the weight of the surviving particles is adjusted by Eq. (3.14) to preserve the mean particle weight.
- *modifying number densities*: The Shift input file processing is modified to read in a user-supplied list of materials that are modified throughout the transient. For each material in the list, another list of fractional density values is supplied that should be the same length as the number of time steps. Number densities from t^0 are stored in memory throughout the transient so that material densities can be updated before each MC solve.
- *Denovo adjoint solver*: The Shift source code has existing infrastructure for communicating in-memory

flux storage with the Denovo deterministic solver for adjoint-based variance reduction techniques. This infrastructure is leveraged to enable SP_3 adjoint calculations to compute point kinetics parameters. After completing the active cycles for each MC step, the deterministic adjoint solution is passed to the MC tallying data structure and used to weight numerical integrals to generate the point kinetics parameters defined by Eq. (2.54).

- *Denovo CMFD solver*: Denovo also had a CMFD solver implemented for fission source acceleration. This infrastructure is migrated to the Shift source code and refitted to process MC mesh tally information as input.
- *Census and point kinetics tallies*: Time census tallies are implemented by modifying existing infrastructure for surface census tallies. At every collision, an incident particle’s state information (position, energy, direction and modified particle weight) are tallied and written to an HDF5 file in between MC solves and read back into memory at the start of the next MC time step. The surface census tallies already implemented in Shift before this work, record the same particle state information every time a particle crosses a surface. Point kinetics tallies are implemented by modifying existing infrastructure for nodal tallies. For delayed neutron fractions, analog fission matrices are scored every time a progenitor neutron of energy g' births a fission neutron in energy group g in a given MC tally mesh cell. For neutron generation times, a track-length tally is used to score contributions weighted by $\frac{1}{v_g}$ in each MC tally mesh cell. After active cycles have completed, the tally data is post-processed as a weighted integral over all of the spatial cells and energy groups.
- *Cross section data structures*: In its original unmodified implementation, Shift only supports delayed neutron spectra and multiplicities that are averaged over the prompt and delayed constituents. In the transient implementation, new data structures are written to house multigroup delayed neutron data ($\lambda_j, \beta_j, \chi_{d,j}$) for each delayed neutron precursor group j . User-supplied group-averaged velocities are also added to the multigroup neutron data infrastructure.

3.3 C5G7 TD-3 Benchmark Results

The TML implementation in Shift was first tested using the C5G7 deterministic transient benchmark. The C5G7 reactor benchmark was initially developed in 2001 [92] to test deterministic transport codes that do not use spatial homogenization above the fuel pin level. The model is representative of a small Pressurized Water Reactor (PWR) core with quarter-core radial symmetry. There are sixteen fuel assemblies: eight Uranium Dioxide (UO₂) assemblies, and eight Mixed Oxide (MOX) assemblies, surrounded by a water reflector. Each

assembly consists of a 17×17 array with 264 fuel pins, 24 guide tubes for control rods to be placed, and one instrument tube in the center of the grid for a fission chamber. A diagram of the south-east quadrant homogenized at the pincell level can be seen in Fig. 3.8.

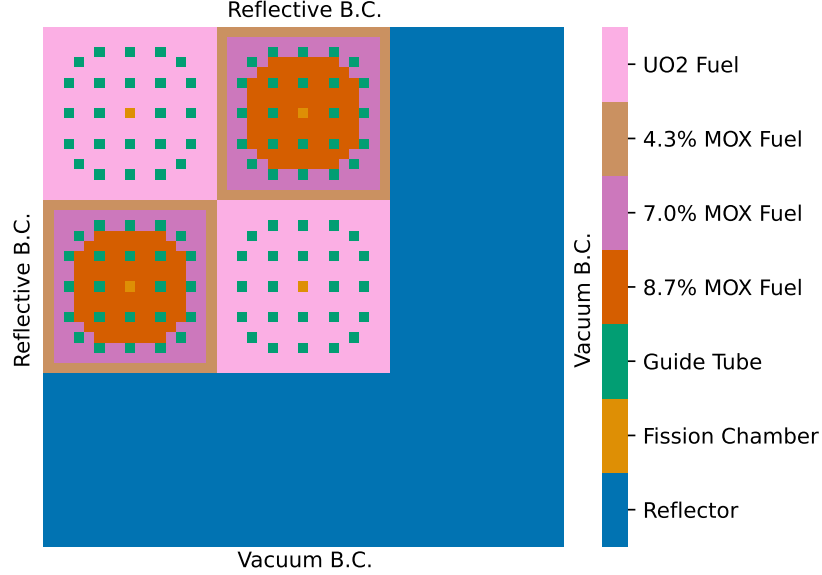


Figure 3.8: South-east quadrant of the C5G7 reactor core configuration homogenized at the pincell level.

Note that there are three levels of enrichment within the MOX assemblies: 4.3%, 7.0%, and 8.7%. Each fuel cell has a pitch of 1.26 cm and has been homogenized into two regions: a fuel zone homogenized over the ceramic fuel, gap, and cladding with radius of 0.54 cm, and a surrounding moderator zone. Transport-corrected macroscopic cross sections and scattering matrices are given in the benchmark specifications [92] for the seven energy groups structure seen in Table 3.1. Continuous-energy MC calculations are also possible, as the detailed geometry specifications for the unhomogenized fuel pins and isotopic compositions of materials are also available [93].

Table 3.1: Energy ranges for the multigroup cross sections specified by the C5G7 benchmark.

Group	1	2	3	4	5	6	7
Upper bound (eV)	1×10^7	1.36×10^6	9.2×10^3	55.6	4.1	0.63	0.13
Lower bound (eV)	1.36×10^6	9.2×10^3	55.6	4.1	0.63	0.13	0.0

A transient version of the C5G7 benchmark, called C5G7-TD, was proposed in 2015 [94], [95]. In addition to the multigroup cross sections specified for the steady-state calculation, kinetics parameters are also specified [94], i.e., delayed neutron precursor group decay fractions, decay constants, and fission spectra. While the information necessary to support continuous-energy calculations is available, at the time of implementation

and writing, the Shift MC code does not support the processing of delayed neutron fission spectra. Therefore, all calculations shown in this dissertation are performed with multigroup MC physics.

The C5G7-TD benchmark was planned to be studied in three phases:

1. *Kinetics Phase* – without feedback,
2. *Dynamics Phase* – with feedback,
3. *High-Fidelity Phase* – uncertainty propagation in high-fidelity multi-physics calculations.

At the time of writing, the C5G7-TD benchmark is still in the Kinetics Phase. In this phase, several prescribed transient exercises are specified, such as the insertion/withdrawal of control rods, and modifying the density of the moderator within the core of the reactor to varying magnitudes and rates. This chapter presents the results of simulating the third prescribed exercise, “TD3”. Exercise TD3 simulates four transient events by linearly varying the core moderator density for a 2D version of the reactor geometry with the time-dependent profile seen in Fig. 3.9, where ω is the minimum moderator density used throughout the transient at $t = 1$ s.

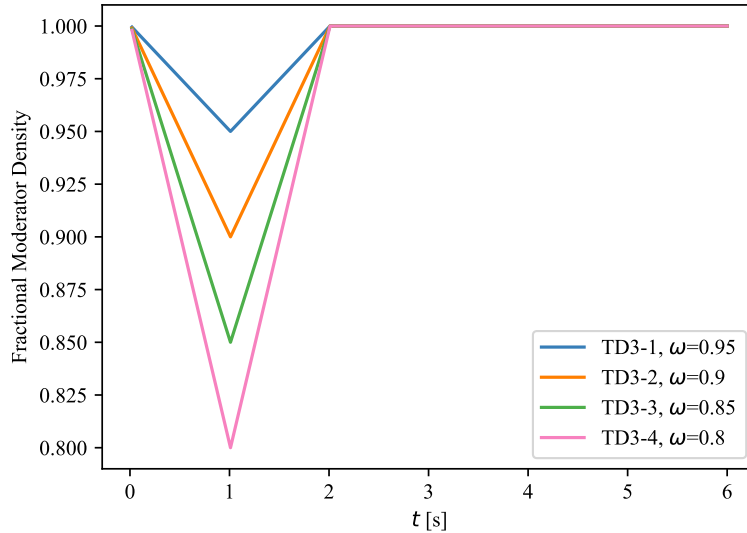


Figure 3.9: Time-dependent changes in the average moderator density specified for the C5G7-TD3 exercises.

The TD3 simulation results from Shift have been compared with the same transient simulation with the MPACT code, a 2D/1D deterministic solver maintained by the University of Michigan and ORNL that uses the Method of Characteristics (MOC) in the radial direction and an P_3 nodal method in the axial direction [96]. A detailed description of the C5G7-TD benchmark results with MPACT is provided in [14], [97]. The

TD3 problem has also been simulated by Shaner [20] using a frequency transform method [98] implemented in OpenMC, but has not been directly compared with Shift in this work.

To compute point kinetics parameters for the TML, an adjoint eigenvalue solution is necessary. Adjoint-weighted tallies are achievable with MC solvers [45]. Unfortunately, implementing them in Shift is beyond the scope of this dissertation work, so a deterministic solution is used instead. In Zhu’s original TML implementation the adjoint solution comes directly from the CMFD solver by transposing the group storage of the fission and scattering matrices. For this work, Denovo [88], the deterministic counterpart to Shift in ORNL’s code suite, is used to generate the adjoint solution. The adjoint solution requires a Cartesian mesh definition to be specified in the Shift input file as well as a multigroup cross section library. Since the implementation in Shift currently only supports multigroup physics, the same multigroup cross section library is shared between the Shift and Denovo calculations. Most of the adjoint capabilities in Denovo are specified for fixed-source calculations. However, an unadvertised SP_3 adjoint k -eigenvalue solver exists within the code and is used to make use of the existing Denovo infrastructure and avoid implementation of an additional adjoint eigenvalue solver. The adjoint SP_3 solver internally reverses the sign of the leakage operator and transposes the scattering and fission matrices so that manual modification of the multigroup cross section databases is not required.

3.3.1 PCQM Results

Before implementing the full TML in Shift, a PCQM implementation (without the CMFD intermediate level) was first implemented to test the transient fission source iteration and provide a benchmark to assess performance improvements upon implementing the CMFD level. The PCQM results are shown in Fig. 3.10 for a macro time step of $\Delta t_{MC} = 0.1$ s, which is the same transport-level time step used for a deterministic PCQM simulation of the same transient simulated by DeHart et al. [99]. Each MC step was simulated with 1×10^5 neutron histories per cycle for 150 inactive cycles and 250 active cycles. The micro time step was selected to be $\Delta t_{EPKE} = 1$ ms.

The results show that stochastic noise in the dynamic reactivity after $t > 2$ s is amplified by the exponential transform of the solution that the EPKE solver assumes. This same phenomenon was also observed by Shaner for a frequency transform method [20] that also assumes exponential behavior of the amplitude function. Shaner showed that the stochastic noise for reactivities near zero is resolved as the number of particle histories is increased.

Figure 3.11 presents the same simulations with stochastic noise in the dynamic reactivity artificially removed for $t > 2$ s by forcing the dynamic reactivity in the simulation to be zero after $t = 2$ s instead of

being derived from MC tallies (i.e. $\rho(t > 2\text{ s}) \equiv 0$). To accomplish this, the dynamic reactivity computed by the CMFD solver is hardcoded to be zero for $t \geq 2\text{ s}$ for the C5G7-TD3 transient. Removing noise in the dynamic reactivity also results in a smoothing of the fractional core fission rate profile. From these results it is also apparent that initial bias immediately after $t = 2\text{ s}$ is propagated throughout the transient, resulting in positive feedback loop as time progresses. In future work, adaptive schemes for devoting more computational resources to resolving statistical uncertainties for dynamic reactivities near zero should be considered.

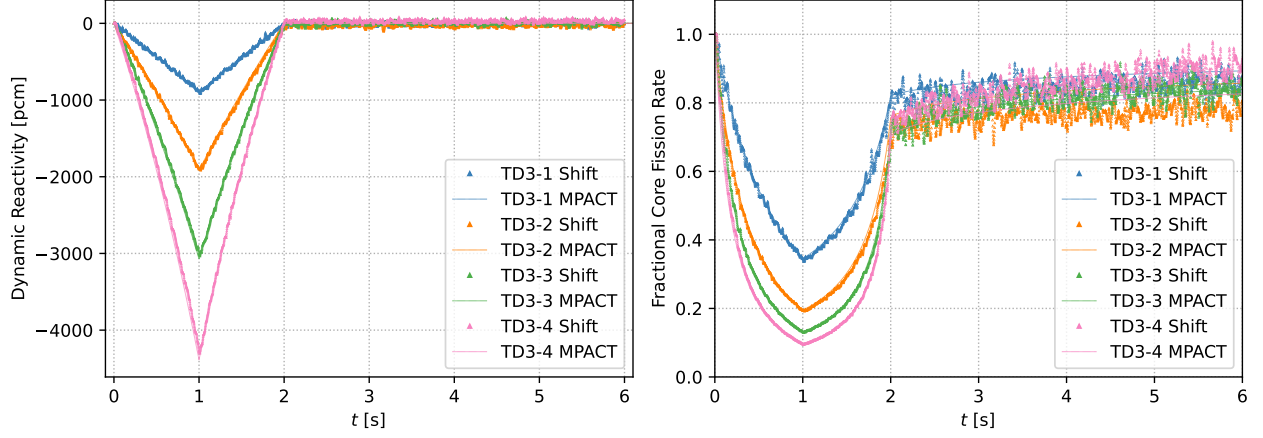


Figure 3.10: Dynamic reactivity and relative fission rate of the PCQM implementation.

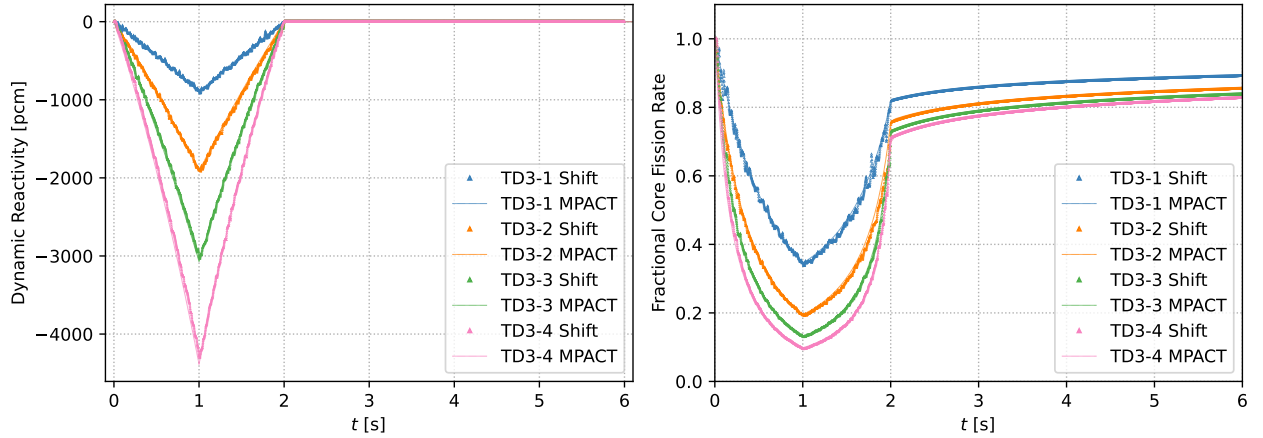


Figure 3.11: Dynamic reactivity and relative fission rate of the PCQM implementation with stochastic noise removed for $t > 2\text{ s}$.

3.4 TML Results

Figure 3.12 shows the results of the TD3 simulations using the TML implementation, now with the CMFD level. Each TD3 case simulated was run to $t = 6\text{ s}$ with 1×10^5 particle histories per cycle, 150 inactive

cycles, and 250 active cycles. The CMFD mesh was defined using a pincell resolution (51×51 mesh elements) over the active region of the reactor. Partial current tallies were used to define the CMFD boundary sources, but the CMFD calculation did not directly include the reflector regions. All of the macroscopic cross reaction rate, surface current and point kinetics tallies were accumulated on this mesh during the MC active cycles to be used as input for the CMFD solver. The time steps at the various TML levels were selected to be $\Delta t_{MC} = 0.25$ s, $\Delta t_{CMFD} = 0.05$ s, and $\Delta t_{EPKE} = 0.001$ s.

The TML method noticeably reduces the stochastic noise in both the dynamic reactivity and corresponding fractional core fission rate profiles. This is because (1) there are fewer MC steps being computed throughout the transient and (2) MC tallies (and their accompanying stochastic noise) are not directly driving the EPKE solution. Instead, point kinetics parameters are derived from the CMFD solutions that use linearly-interpolated MC-generated cross section to provide spatial and energy resolution updates to the shape function throughout the transient. The importance of these intermediate updates is illustrated by the slight curvature in the dynamic reactivity (left) for both the descending ($0 \leq t < 1$ s) and ascending ($1 \leq t < 2$ s) portions of the curve. Therefore, linearly interpolating EPKE parameters between MC solutions at $t = 0$, $t = 1$, and $t = 2$, is not sufficient to capture this feature. Error bars are not presented in Fig. 3.12 as the propagation of error in the MC tallies through the CMFD and EPKE subsolvers is not straightforward. Instead, batch statistics are used to assess the resolution of stochastic noise later in Fig. 3.14.

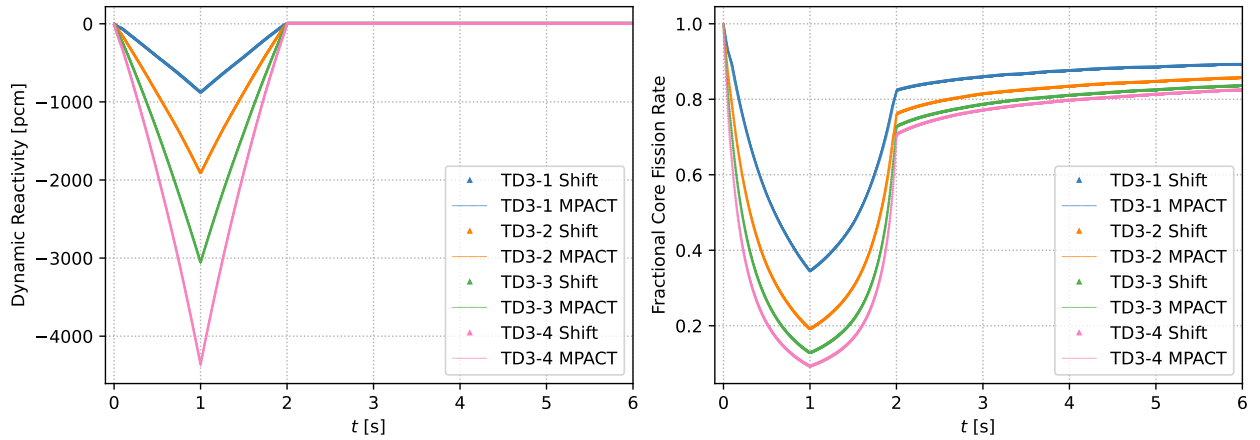


Figure 3.12: Dynamic reactivity and relative fission rate of the TML implementation with stochastic noise removed for $t > 2$ s.

Several norm comparisons are given in Table 3.2 to assess TML performance improvements over the PCQM implementation. The MPACT solutions are used as the reference solution. The L_1 and L_∞ norm evaluations for the TML decrease by roughly an order of magnitude for every TD3 case when compared to the PCQM. The L_2 norm improves by a factor of 2-4 for all TD3 cases.

Table 3.2: Norm comparisons (given in %) to MPACT solution for both PCQM and TML implementations.

Norm		TD3-1	TD3-2	TD3-3	TD3-4
$L_\infty = \max_n \left[\frac{p_{n,\text{Shift}} - p_{n,\text{MPACT}}}{p_{n,\text{MPACT}}} \right]$	PCQM	8.42	13.48	10.12	12.38
	TML	2.40	3.65	2.83	2.23
$L_2 = \frac{1}{N} \left[\sum_n \left(\frac{p_{n,\text{Shift}} - p_{n,\text{MPACT}}}{p_{n,\text{MPACT}}} \right)^2 \right]^{\frac{1}{2}}$	PCQM	0.28	0.47	0.28	0.46
	TML	0.09	0.12	0.13	0.13
$L_1 = \frac{1}{N} \sum_n \left \frac{p_{n,\text{Shift}} - p_{n,\text{MPACT}}}{p_{n,\text{MPACT}}} \right $	PCQM	2.17	3.70	2.18	3.42
	TML	0.62	0.91	1.07	1.06

Ideally, the use of a larger MC time step would bring down the total simulation time as the MC solve is much costlier than both the CMFD and EPKE solves. To show this, a rough timing comparison between the various implementations is shown in Table (3.3). All of the Shift calculations were executed on an ORNL compute cluster featuring dual-processor AMD Opteron 6378 CPUs with 16 cores per CPU, for a total of 32 cores per node, and 128 GB RAM per node. All Shift simulations of the C5G7 reactor geometry were run on eight nodes (256 cores). The MPACT simulations were carried out for the same transient specifications up until $t = 10$ s. Additionally, the MPACT benchmark only reported wall time using nine processors. The wall times reported by MPACT were then multiplied by a factor of nine to give an estimate for the number of core hours used. While these numbers are approximate and not directly comparable, it can be seen that the Shift PCQM and TML implementations are roughly one and two orders of magnitude slower, respectively, than the MPACT implementation.

Table 3.3: CPU time comparison (hours) between the TD3 solution methods.

TD3 Case	Shift-PCQM (6 s)	Shift-TML (6 s)	MPACT (10 s)
1	2510.1	395.0	30.6
2	3067.1	395.1	29.7
3	2533.1	398.0	29.7
4	2039.8	351.1	29.7

As stated previously, Shaner showed that the stochastic noise in the MC frequency transform method was resolved with an increasing number of particle histories [20]. For the MC TML implementation, small perturbations in the dynamic reactivity are amplified by the EPKE solver. A batch analysis was performed to assess whether this is also the case with the MC TML implementation. Three values for the number of histories to simulate per fission source iteration cycle: $N_{\text{hpc}} = 1 \times 10^4$, 1×10^5 , and 1×10^6 . The C5G7-TD3-4 transient was modeled based on the magnitude of its change in reactivity. Other than varying N_{hpc} , all Shift inputs used to generate each individual simulation of the batch analysis were identical to those used to generate the results presented in Fig. 3.12. All values of N_{hpc} were simulated with ten batches, where each batch began with a unique random seed. The average over the two batches for each point in time are shown

as the solid lines in Fig. 3.14. The shaded regions show the standard deviation at each point in time. It is clearly shown that the noise is resolved with increasing particle histories. The quality of the solution is fairly consistent while the reactivity is changing, i.e. $t < 2$ s. The solution is much more sensitive to stochastic noise for dynamic reactivities near zero. In future implementations, more computational resources (i.e., larger values of N_{hpc}) should be devoted to temporal regions where the imposed reactivity should be zero.

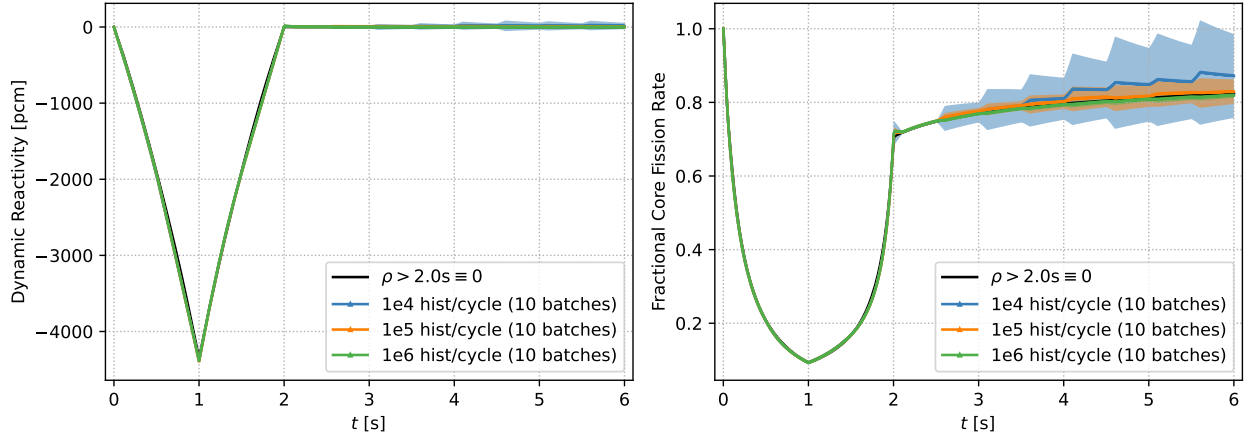


Figure 3.13: Uncertainties from stochastic noise associated with dynamic reactivity and relative fission rate for TD3-4.

A refinement of the MC and CMFD time scales was also analyzed to see how the ratio $\Delta t_{\text{MC}}/\Delta t_{\text{CMFD}}$ affects the accuracy of the solution. The L_2 norm comparison for each of the simulations for various values of Δt_{MC} is shown in Fig. 3.14 (left), where the reference solution used time steps of $\Delta t_{\text{MC}} = 0.1$ s, $\Delta t_{\text{CMFD}} = 2$ ms, and $\Delta t_{\text{EPKE}} = 80$ μ s. Accuracy gains appear to level off for values of $\Delta t_{\text{MC}} > 10$. The right plot in Fig. 3.14 shows the percentage of the total simulation time spent computing the MC time propagations. The percentage of time spent on the MC solves appear to decrease linearly with an increasing number of CMFD steps per MC step. Since the most significant gains in solution accuracy occur for $\Delta t_{\text{MC}}/\Delta t_{\text{CMFD}} < 10$, the MC solver takes upwards of 95% of the total simulation time in the TML implementation. Therefore, acceleration efforts should ideally focus on reducing the number of MC steps required to solve a transient while retaining the accuracy of the solution.

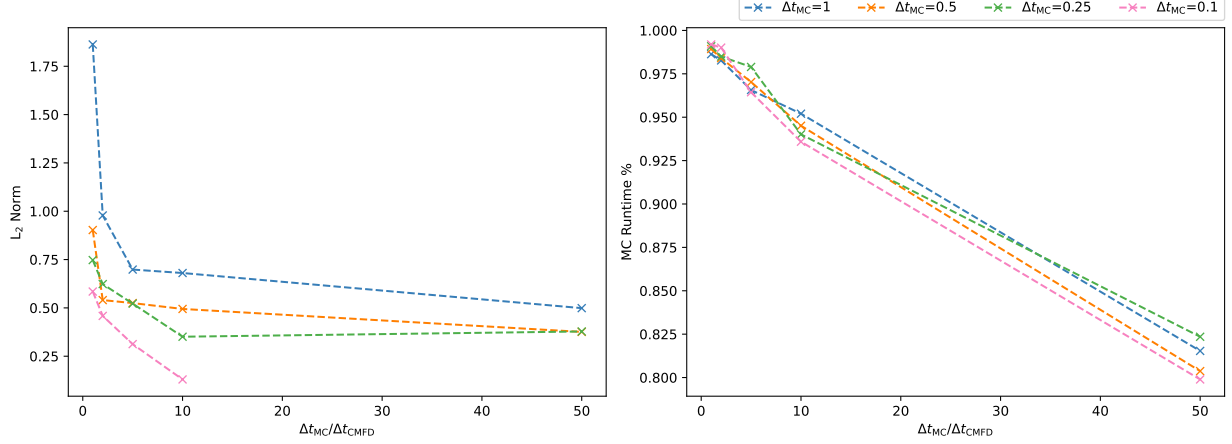


Figure 3.14: Time step convergence analysis for TD3-4 exercise.

3.5 Prompt Feedback Model

There are well-established methods for determining Doppler-broadened cross sections for continuous-energy MC codes like Shift by loading temperature-dependent cross sections into memory so that they can be interpolated either before runtime or on-the-fly [100]. Since the scope of this work is limited to multiphysics calculations, a simplified fuel temperature dependence for multi-group cross sections has been implemented in Shift instead that has the form

$$\Sigma_{t,\text{Doppler}}^n = \Sigma_t^n \left(1 + c\sqrt{T^n - T^0} \right), \quad (3.27)$$

where T^0 is the initial temperature, T^n is the temperature at $t = t^n$, Σ_t^n is the total macroscopic cross section prescribed at $t = t^n$, and c is a free parameter chosen by the user to adjust the magnitude of the Doppler broadening contribution. A value of $c \equiv 1 \times 10^4$ was used for all simulations presented in this dissertation that use Doppler feedback.

A Cartesian grid is imposed over the problem geometry to track temperature changes throughout the transient. During transport, when cross sections are queried for fissile materials, the local temperature is checked and the total cross section, modified by Eq. (3.27), is returned. Since a significant amount of the resonance absorption for ^{238}U occurs at the 6.67 eV resonance, the Doppler broadening is only applied to the corresponding energy group (group five from Table 3.1 for the C5G7-TD benchmark). To estimate the local changes in temperature, a fission tally is first used to estimate local energy deposition in the fuel. If each fission corresponds to ~ 200 MeV of recoverable energy, then an estimate for the reactor power $P(t)$ can be

made when scaled by the initial reactor power $P(t = 0)$. The changes in temperature are then computed as

$$\Delta T(t) = C_{QT} \int_0^t [P(t') - P(0)] dt', \quad (3.28)$$

$$C_{QT} \left[\frac{\text{K}}{\text{MW s}} \right] = \frac{1}{c_p} \left[\frac{\text{K g}}{\text{W s}} \right] \frac{1}{\rho_d} \left[\frac{\text{mm}^3}{\text{g}} \right] \frac{1}{V_f [\text{mm}^3]} \left[\frac{1 \times 10^6 \text{ W}}{\text{MW}} \right], \quad (3.29)$$

where C_{QT} is an energy to temperature conversion constant, c_p is the specific heat of the material, ρ_d is the density, V_f is the volume of the Cartesian mesh element. To test the Doppler broadening implementation, a step reactivity insertion was tested with the C5G7 reactor geometry. At the time of writing, the Shift implementation only supports fractional changes to material densities to drive transients and not time-dependent geometry changes that would be necessary to support a typical control rod ejection scenario. Therefore, the transient was driven by a 4% step increase in the moderator density at $t = 0.5$ s from its initial fractional value of unity. The results are shown in Fig. 3.15 and show a power pulse and subsequent decay down to an asymptotic power level higher than the initial baseline power level.

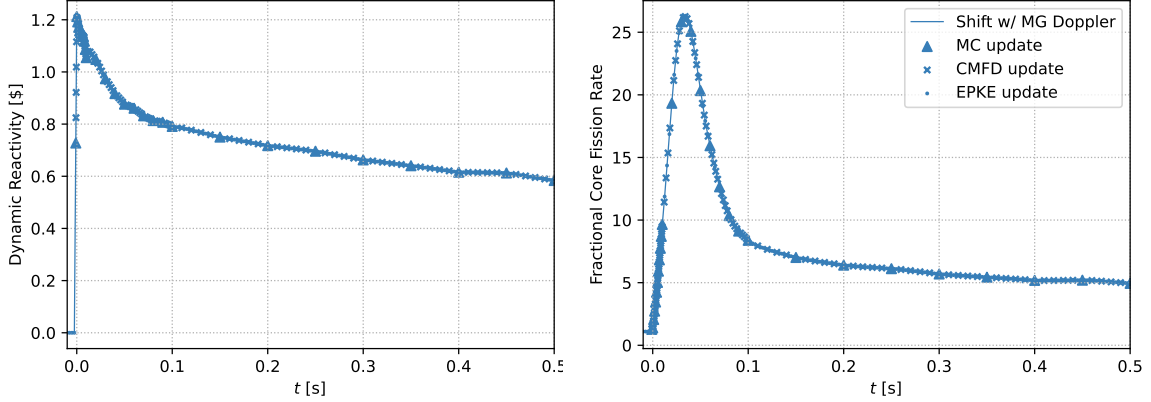


Figure 3.15: C5G7 step reactivity insertion with simplified Doppler feedback.

A variable MC time step was used to simulate the transient that placed more computational resources (i.e., more MC solves) towards resolving the initial power pulse shape. The MC time steps were specified as

$$\Delta t_{MC} = \begin{cases} 1 \text{ ms} & t < 10 \text{ ms} \\ 10 \text{ ms} & 10 \text{ ms} \leq t < 100 \text{ ms} \\ 50 \text{ ms} & 100 \text{ ms} \leq t < 500 \text{ ms} \\ 100 \text{ ms} & 500 \text{ ms} \leq t < 1.5 \text{ s} \end{cases}. \quad (3.30)$$

For each macro MC time step, the time step ratios were $\Delta t_{MC}/\Delta t_{CMFD} = 5$ and $\Delta t_{CMFD}/\Delta t_{EPKE} = 10$. A

total of 1×10^6 particle histories per fission source iteration cycle were simulated for 150 inactive cycles and 250 active cycles. The total simulation used 310.5 CPU hrs across eight compute nodes with 32 processors per node, where 94.5% of the total simulation was spent on the MC calculations. The density and specific heat were taken from Table 10-I from Ott [49] as $1 \times 10^{-3} \text{ g mm}^{-3}$ and $0.32 \text{ W s g}^{-1} \text{ K}^{-1}$, respectively. The initial power level was assumed to be that of a typical small modular reactor of 100 MW. The average weight of particles banked as a transient time source between MC solves (before population combing) is shown in Fig. 3.16. It is shown that the average weight of particles decreases for increasingly large MC time step Δt_{MC} . This is the expected behavior, as particles are more likely to hit time census events for shorter MC time steps.

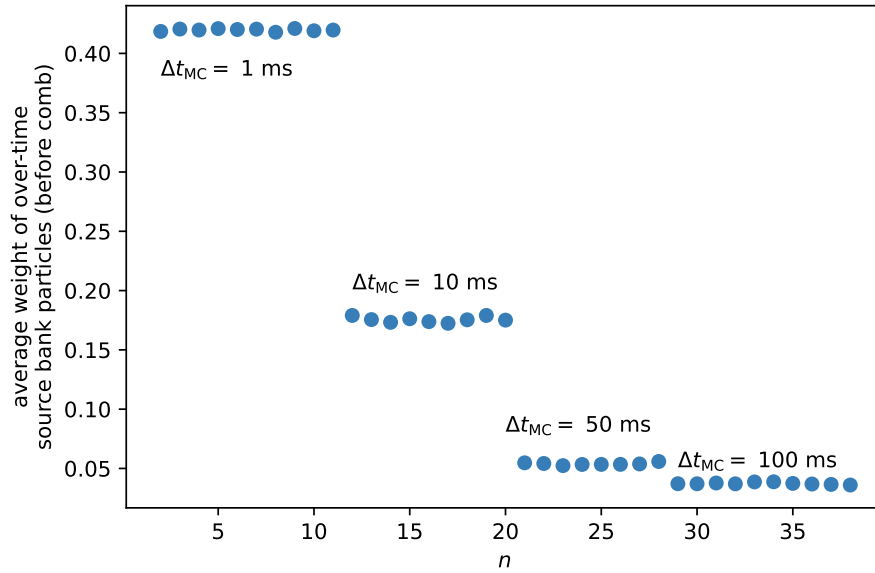


Figure 3.16: Average weight of time source particles (before population combing).

3.6 Dual-Purpose Canister Model

While the MC TML results for the C5G7-TD3 benchmark are certainly promising, it is difficult to assess the applicability of the algorithm for practical problems with this geometry because of its relatively small size, symmetry, and low dominance ratio. In order to show that the TML implementation can be used to model a spent fuel transient criticality (the funding source for this work), a large DPC for used nuclear fuel was modeled. The geometry was modeled after a Holtec International MPC-32 canister [101], where 32 spent fuel assemblies are loaded into the canister. Each assembly was modeled as a Westinghouse 17×17 fuel pin array, where the length and width are 21.4 cm and the height is 365.73 cm. The isotopic concentrations were determined *a priori* via a KENO [102] depletion calculation for a 10 GWD/MTU burnup. To capture

axial variations in the isotopic concentration, the assemblies were modeled in 18 equal-length axial segments (20.32 cm each). Because each of the assemblies occupied a different position in the reactor, each has a unique spatial profile of isotopic compositions, making for significant asymmetries in the spatial fission distribution. Figure 3.17 depicts both the axial (left) and radial (right) cross sectional views of the DPC geometry configuration, where the different colors signify materials with unique isotopic compositions.

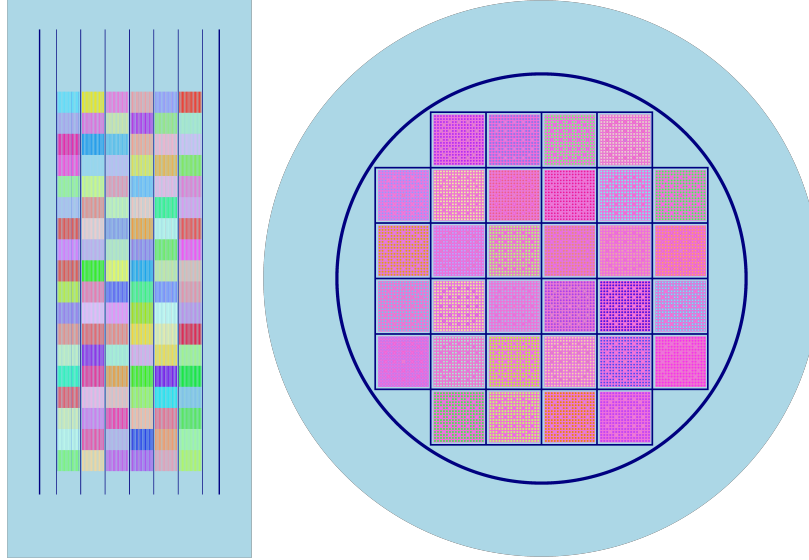


Figure 3.17: Axial (left) and radial (right) cross-sectional views of the DPC geometry modeled. This DPC configuration has 32 unique fuel assemblies, each with 18 axial segments.

Since the Shift TML implementation only supports multigroup physics at the time of writing, multigroup cross sections were computed using a continuous energy steady-state criticality calculation in Shift. The energy group structure and neutron kinetics parameters were approximated as the same as those defined in the C5G7 for UO_2 and are shown in Tables 3.4 and 3.5. To simulate a transient accident, the DPC is first flooded with water at the steady state ($t \leq 0$), with $k_{\text{eff}} = 1.24960776 \pm 1.58083823 \times 10^{-9}$. At $t = 0^+$, a 10% step increase in the water density located within the assemblies themselves is simulated.

The simulated transient is shown in Fig. 3.18. There were 350 inactive cycles and 250 active cycles run with 2×10^6 histories per cycle. The total simulation time was 1086.15 CPU hrs across eight nodes with 32 processors per node, where $\sim 98.7\%$ of the simulation was spent on the MC calculations. The time steps were the same as those given by Eq. (3.30). For reasons discussed in Section 3.7, a coarser CMFD mesh was used for this geometry as compared to the C5G7 geometry. Radially, each assembly was represented by a 3×3 grid of CMFD mesh elements, as opposed to the pincell resolution used for the C5G7 geometry. Axially, the CMFD mesh was split into 18 equal segments along the fuel assemblies. The temperature grid imposed for Doppler feedback was the same spatial resolution as the CMFD mesh. The power pulse in this transient

Table 3.4: Delayed neutron precursor decay constants and fractions for UO_2 given in the C5G7-TD benchmark [95].

Group	$\lambda_j \text{ (s}^{-1}\text{)}$	β_j
1	$2.133\,33 \times 10^{-4}$	$7.824\,84 \times 10^{-5}$
2	$1.045\,14 \times 10^{-3}$	$6.405\,34 \times 10^{-4}$
3	$6.039\,69 \times 10^{-4}$	$2.278\,84 \times 10^{-4}$
4	$1.339\,63 \times 10^{-3}$	$5.786\,24 \times 10^{-4}$
5	$2.293\,86 \times 10^{-6}$	$9.975\,39 \times 10^{-4}$
6	$7.051\,74 \times 10^{-4}$	$4.332\,65 \times 10^{-4}$
7	$6.003\,81 \times 10^{-4}$	$3.223\,55 \times 10^{-4}$
8	$2.077\,36 \times 10^{-4}$	$1.238\,82 \times 10^{-4}$
Sum	$7.009\,22 \times 10^{-3}$	$3.402\,33 \times 10^{-3}$

Table 3.5: Delayed neutron fission spectra for UO_2 given in the C5G7-TD benchmark [95].

Group	1	2	3	4-7
1	0.000 75	0.985 12	0.014 13	0
2	0.030 49	0.969 07	0.000 44	0
3	0.004 57	0.974 01	0.021 42	0
4	0.020 02	0.972 71	0.007 27	0
5	0.056 01	0.938 18	0.005 81	0
6	0.060 98	0.934 44	0.004 58	0
7	0.106 35	0.882 98	0.010 67	0
8	0.093 46	0.902 60	0.003 94	0

happens on a much quicker timescale than that of the C5G7 step reactivity insertion. This is because the Doppler broadening effect is much more pronounced and fast-acting for the larger positive reactivity insertion used in the DPC simulation.

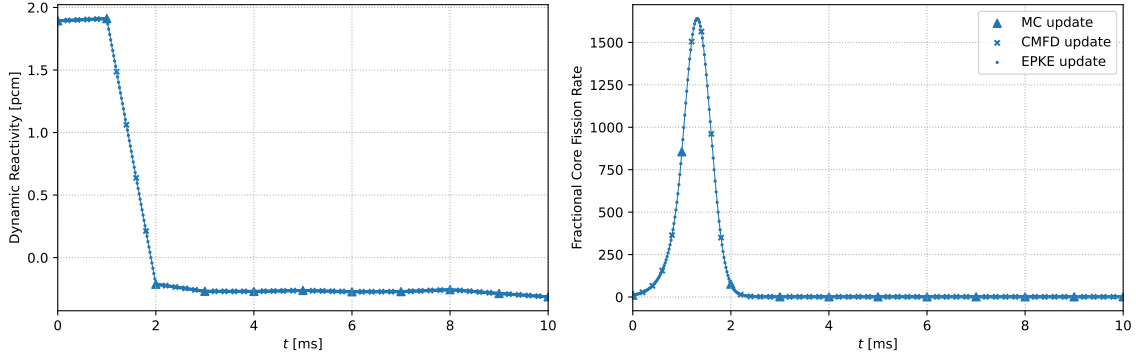


Figure 3.18: Dynamic reactivity (left) and fractional core fission rate (right) for step reactivity insertion with DPC canister.

Figure 3.19 (top) shows an x - y slice of the core fission rate integrated over the active axial length of the core at selected MC time steps. The bottom row of Fig. 3.19 shows the percent change in the normalized fission shape from the immediately preceding MC time step (t^{n-1}). Because there are no significant changes

to the problem geometry or isotopic compositions throughout the transient, these plots show that the radial fission distribution only changes by at most about $\pm 8\%$ in between MC time steps during the power excursion. The axial fission distribution is shown in Fig. 3.20, where again only minimal changes in the fission distribution $\sim \pm 4\%$ occur throughout the transient. This particular transient is then ideally suited for the quasi-static assumption that there are relatively small changes in the shape function.

While the results presented for the DPC are not physical because an approximate form of Doppler broadening was used to introduce feedback to the TML algorithm, they do suggest that the algorithm is applicable for large, asymmetric geometries. The initial motivation from Chapter 1 was to develop Monte Carlo transient capabilities with methods to solve practical problems that are more tractable than with methods currently in use in production-level transport solvers. The performance comparisons with the PCQM from the C5G7 transient and the practical geometry of the DPC transient have demonstrated that the TML meets these criteria.

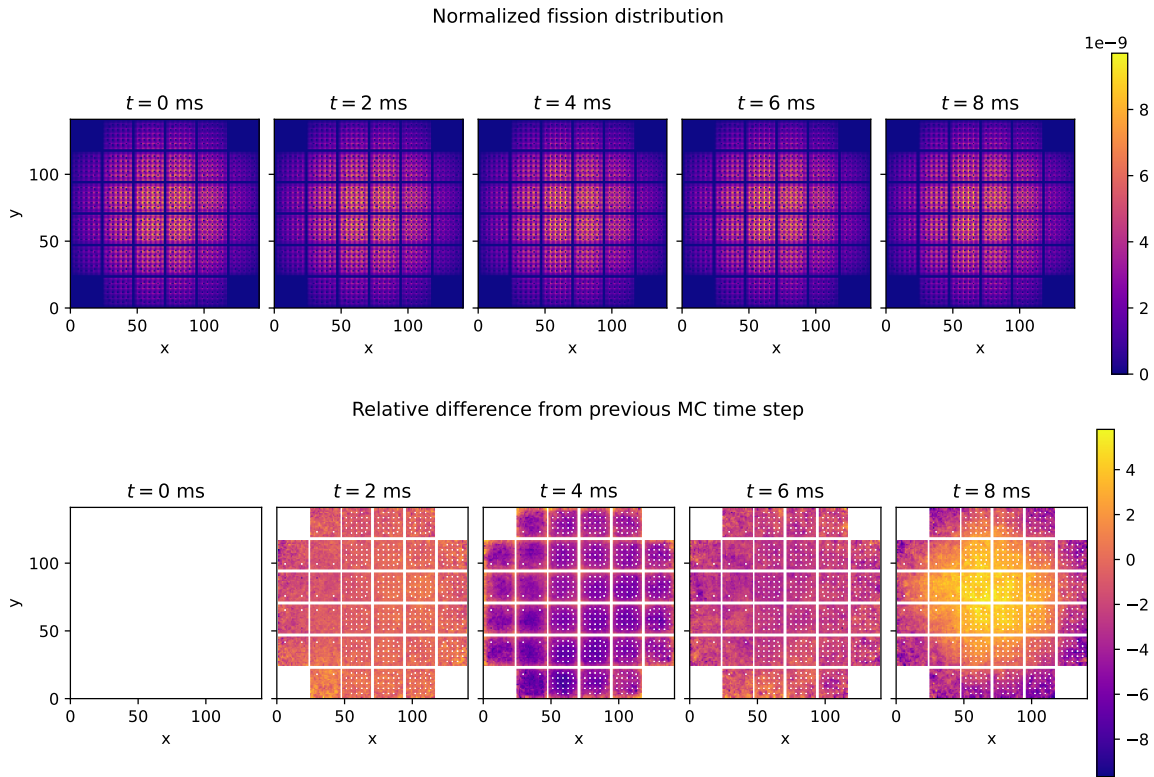


Figure 3.19: Radial fission profile (top) and percent differences in the fission profile from the previous MC time step t^{n-1} (bottom) integrated over axial length of core at select points in time throughout the DPC simulation. The normalizations are computed such that the radial fission distribution integrated over all of phase space at each individual time step is unity.

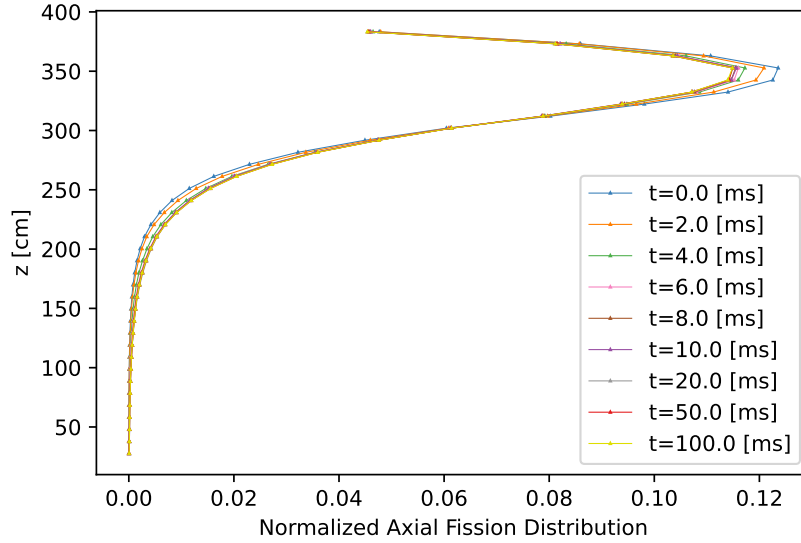


Figure 3.20: Axial fission profile (top) and percent differences from $t = 0$ s (bottom) integrated over radial length of core at select points in time throughout the DPC simulation. The normalizations are computed such that the axial fission distribution integrated over all of phase space at each individual time step is unity.

3.7 Issues with the CMFD Solver

One of the most significant issues facing the TML is ensuring the convergence of the CMFD solver, which does not converge if the matrix system is ill-conditioned. This problem is particularly difficult to handle when considering that the group-to-group scattering matrices and surface current tallies in MC calculations are dependent on analog tallies. For regions where few analog tallies occur, CMFD matrix entries for Eq. (2.150) are potentially underpopulated and exacerbate the ill-conditioned nature of the system. Obviously, a sufficient number of particles should be run in order to ensure that the analog tallies are non-zero. Hybrid methods for improving the uncertainty in scattering matrices could also be implemented [103]. For small problems, like the C5G7 calculations in 2D using multigroup physics, 1×10^5 histories per cycle were seemingly sufficient to ensure sufficient analog interactions that filled the CMFD matrices.

For the spatially larger DPC calculations in 3D, the CMFD matrices were much more difficult to fill considering the large axial asymmetry in the neutron flux. One could always decrease the resolution of the CMFD spatial mesh to ensure that tally statistics are sufficient. However, Keady recommended in their MC-CMFD analysis that CMFD spatial resolutions be of a higher fidelity than pincell resolution in order to ensure the accuracy and stability of the solution [104]. This conclusion was well-known in the deterministic community and as a rule of thumb, any coarse cell optical thickness $> \sim 2$ mean free paths is considered

problematic. A typical mean free path for a PWR pincell is ~ 1.5 . It is possible that the DPC configuration is particularly sensitive to the CMFD cell optical thickness.

The difficulties encountered with the MC-CMFD coupling might suggest that a problem-specific tradeoff might need to be made between the accuracy and conditioning of the CMFD matrices. In reviewing the CMFD literature for applications with MC solvers (almost exclusively for fission source acceleration), it appears that most studies have only been conducted for 1D, 2D or small 3D geometries [82], [105], [106]. Further, the small 3D geometries that were studied were typically characterized by radial symmetry and low dominance ratios. The robustness of the MC-CMFD coupling remains unclear without further investigation.

3.8 Summary

The TML method was implemented using a modified MC fission source iteration solution at the transport level for the first time. The MC TML implementation showed marked reductions in both stochastic noise and overall simulation time when compared against the standard MC PCQM technique for the C5G7-TD3 transient benchmark problem. It was shown that stochastic noise where the dynamic reactivity is near zero is amplified in the transient power profiles but that this stochastic noise is resolved as the number of histories simulated approaches infinity. Similar stochastic noise resolution was observed with Shaner’s frequency transform method, indicating that the observation is likely generalizable to other transients using the TML.

It was also shown that, for the C5G7-TD3 problem, performance gains diminish significantly after the number of CMFD per MC steps of around ten, where $\geq 95\%$ of the total simulation time is spent on the MC calculations. Zhu found that roughly 50% of the overall TML simulation time was attributed to the MOC solver, with 31% attributed to the CMFD solver (Zhu’s calculations also included thermal hydraulics solves that accounted for 17% of the overall simulation time) [13]. Though it was expected that the MC solver would dominate the overall simulation time, it draws attention to the fact that further acceleration techniques should be oriented towards limiting the number of MC steps needed to simulate the transient without losing accuracy.

An ad-hoc multigroup Doppler broadening was implemented in the MC TML and used to simulate a prompt supercritical step reactivity insertion for a large 3D DPC for used nuclear fuel. These simulations provided insight to some of the difficulties that the TML might experience for simulating large geometries with significant spatial asymmetries in the flux profile. These difficulties are likely problem-specific though and it remains unclear if general conclusions can be drawn to assess whether the CMFD system will be ill-conditioned *a priori*.

While the MC TML shows speedup when compared to the PCQM, the method is still roughly an order of

magnitude slower than the MOC TML for the C5G7-TD3 transient. Additionally, the TML implementation in Shift only supports multigroup physics. Monte Carlo solves using continuous-energy physics will lead to even more computationally expensive transients. Acceleration schemes are therefore necessary to make the algorithm more tractable for practical applications. The next chapter discusses parallel-in-time integration and correlated sampling as potential mechanisms to accelerate the MC TML implementation.

Chapter 4

Parallel-in-Time Acceleration for the TML

In this chapter, parallel-in-time algorithms are investigated, albeit in serial, as potential acceleration schemes for the Transient Multi-Level (TML) implementation discussed in Chapter 3. While the TML shows reductions in stochastic noise and improvements in overall simulation time compared to other quasi-static techniques, many expensive Monte Carlo (MC) evaluations are still necessary to evaluate practical reactor transients. Therefore, acceleration schemes are necessary to make the algorithm more applicable to practical reactor problems.

A novel method is presented here for accelerating the propagation of the amplitude function between subsolver levels of the TML method. A serial parareal implementation is verified using analytic solutions to the point kinetics equations for an asymmetric reactivity ramp as well as benchmarked against a control rod ejection simulated in the Purdue Advanced Reactor Core Simulator (PARCS) code. The serial implementation is then migrated into the Shift TML solver and tested between two Coarse Mesh Finite Difference (CMFD) levels. Finally, a correlated sampling technique is proposed, to generate coarse-level input to the TML subsolvers so that parallel-in-time acceleration can be applied between MC levels.

4.1 Parareal Algorithm

In neutron transport, parallel algorithms have traditionally been limited to the spatial, directional, and energy domains for deterministic solvers, and particle history for MC solvers. Though parallel-in-time algorithms have existed for several decades, it is not very intuitive to consider parallelization of the temporal domain because of its sequential nature. As the number of available processing units continues to grow though, and performance improvements from parallelization of other domains of phase space begin to saturate, parallel-in-time algorithms are becoming increasingly attractive for large practical engineering calculations. Parallel-in-time algorithms present an opportunity to extract previously-overlooked performance improvements from computing infrastructures with very many available processing units.

Parallel-in-time algorithms are designed to solve time-dependent initial value problems. There are many ways to solve initial value problems with methods ranging from fully-discrete differencing schemes to multi-stage and multi-step semi-discrete differencing schemes. While temporal differencing schemes share many similarities with differencing schemes for other dimensions of phase space, such as the spatial domain, they are uniquely troubled by that fact that the solution later in time is affected or even determined by the solution earlier in time and not the other way around. This makes parallel-in-time integration techniques inherently difficult as information from previous points in time must be used to infer information for future points in time, i.e., the process is intuitively serial. As a consequence, parallel-in-time integration techniques were largely, though not entirely, ignored for practical engineering applications until the last two decades or so.

As parallelization in other domains of phase space begin to saturate and the number of available processing units continues to grow with each progressive supercomputer, parallelization of the temporal domain has received growing attention with many different approaches [107]. One of these parallel-in-time integration schemes, known as the *parareal* algorithm, was first proposed by Lions, Maday, and Turinici in 2001 [36]. It is by far the most-studied parallel-in-time algorithm, mainly because of its relative simplicity compared to competing algorithms. For these reasons, the parareal algorithm was selected for this exploratory study of parallel-in-time acceleration of the TML scheme.

The parareal algorithm is an iterative application of two separate time integrators or “propagators” to solve the same differential equation(s). For example, backward Euler and four-stage Runge-Kutta (RK4) might be an example of propagators to solve a system of Ordinary Differential Equations (ODEs). One of these propagators, commonly denoted $\mathcal{F}$, is of high accuracy and high computational expense. The other, commonly denoted $\mathcal{G}$, is computationally cheap but less accurate than $\mathcal{F}$. Between the previously-given examples, RK4 would be selected as the $\mathcal{F}$ propagator and backward Euler would be selected as the $\mathcal{G}$ operator because of their relative accuracies (fourth-order versus second-order, respectively) and computational costs. However, there are several ways to define the “high” and “low” accuracy to distinguish between the $\mathcal{F}$ and $\mathcal{G}$ propagators. Most commonly, high accuracy refers to the order of accuracy of the temporal differencing scheme, fourth-order versus second-order Runge-Kutta, for example. Alternatively, high and low accuracy can refer to the resolution of the discretized phase space. For example, a space-time differencing scheme might use a more refined spatial mesh for $\mathcal{F}$ than for $\mathcal{G}$, though convergence issues have been documented if proper interpolation is not used for these types of parareal space-time discretizations [108].

Regardless of the definition of high and low accuracy, $\mathcal{F}$ is defined on a finer temporal discretization than $\mathcal{G}$. Additionally, both discretizations are collocated at $t^0 \leq t^n \leq t^{N_p}$, where t^0 and t^{N_p} correspond to the initial and terminal points in time for the problem being solved, and N_p corresponds to the number of available processors or threads, depending on the parallelization strategy used. The collocated temporal

domain is then partitioned according to the number of available processors as shown in Fig. 4.1.

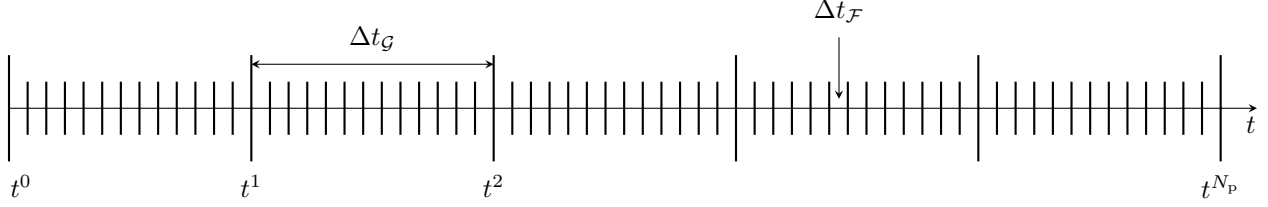


Figure 4.1: Parareal time discretization scheme.

The result of a numerical integration by either $\mathcal{F}$ or $\mathcal{G}$ over time slice $[t^n, t^{n+1}]$ with initial condition $\mathbf{u}^n$ is notated, respectively, as

$$\mathbf{u}^{n+1} = \mathcal{G}(\mathbf{u}^n, t^n, t^{n+1}) , \quad (4.1a)$$

and

$$\mathbf{u}^{n+1} = \mathcal{F}(\mathbf{u}^n, t^n, t^{n+1}) . \quad (4.1b)$$

As an example to clarify the notation used in Eqs. (4.1), consider some arbitrary ODE

$$\frac{du}{dt} = f(u, t) , \quad (4.2)$$

where the function $f(u, t)$ is known. Applying a backward Euler temporal discretization for the coarse propagator, $\mathcal{G}$, would yield the following propagation for a single time step

$$\begin{aligned} u^{n+1} &= u^n + (t^{n+1} - t^n) f(u^{n+1}, t^{n+1}) \\ &= u^n + \Delta t_{\mathcal{G}} f(u^{n+1}, t^{n+1}) , \end{aligned} \quad (4.3)$$

where u^n is the initial condition at initial time t^n for the single time step propagation. The corresponding parareal notation for $\mathcal{G}$ would then be

$$u^{n+1} = \mathcal{G}(u^n, t^n, t^{n+1}) . \quad (4.4)$$

The parareal algorithm, summarized in Algorithm 4.1 and depicted in Fig. 4.2, uses $\mathcal{G}$ (coarse time propagator) to integrate the differential equation over all of the time slices in serial from the beginning to the end of the transient. The algorithm then uses $\mathcal{F}$ (fine time propagator) to integrate all of the decomposed temporal slices $[t^n, t^{n+1}]$ as independent initial value problems in parallel on separate processing units, where

the initial values come from the solution of the previous $\mathcal{G}$ coarse propagations by

$$\mathbf{u}_{k=0}^{n+1} = \mathcal{G}(\mathbf{u}_{k=0}^n, t^n, t^{n+1}), \quad 0 \leq n \leq N_p - 1, \quad (4.5)$$

and k is the parareal outer iteration index. The initial conditions are updated for each parareal iteration after running all N_p propagations of $\mathcal{F}(\mathbf{u}_k^n, t^n, t^{n+1})$ by

$$\mathbf{u}_{k+1}^{n+1} = \underbrace{\mathcal{G}(\mathbf{u}_{k+1}^n, t^n, t^{n+1})}_{\text{prediction}} + \underbrace{\mathcal{F}(\mathbf{u}_k^n, t^n, t^{n+1}) - \mathcal{G}(\mathbf{u}_k^n, t^n, t^{n+1})}_{\text{correction}}, \quad 0 \leq n \leq N_p - 1. \quad (4.6)$$

The final solution $\mathbf{u}_K$ converges towards a serial integration of $\mathcal{F}(\mathbf{u}^0, t^0, t^{N_p})$ as $\mathbf{u}_{k+1} - \mathbf{u}_k \rightarrow 0$, where K is the final parareal iteration index.

Lions et al. showed that the order of accuracy of the $\mathcal{F}$ propagation degrades with increasing k [36]. For a useful application then, the parareal algorithm must converge after just a few k . In addition, computational speedup is only achieved when K is significantly less than the number of temporal slices, $K \ll N_p$. To show this, let $\tau_{\mathcal{F}}$ and $\tau_{\mathcal{G}}$ represent the average computing time required to propagate a single step of $\mathcal{F}$ and $\mathcal{G}$, respectively. Also let $N_{\mathcal{F}}$ and $N_{\mathcal{G}}$ be the fixed number of steps that $\mathcal{F}$ and $\mathcal{G}$ will propagate for each temporal domain slice, respectively. The total computational time to propagate $\mathcal{F}$ over the entire temporal domain in serial will then be

$$c_{\mathcal{F}} = N_p N_{\mathcal{F}} \tau_{\mathcal{F}}. \quad (4.7)$$

The total computational time for the parareal algorithm with K iterations is written as

$$c_{\text{parareal}} = (K + 1) N_p N_{\mathcal{G}} \tau_{\mathcal{G}} + K N_p N_{\mathcal{F}} \tau_{\mathcal{F}}. \quad (4.8)$$

The parareal speedup factor is then

$$\begin{aligned} S_{\text{parareal}} &= \frac{c_{\mathcal{F}}}{c_{\text{parareal}}} \\ &= \frac{1}{(K + 1) \frac{N_{\mathcal{G}} \tau_{\mathcal{G}}}{N_{\mathcal{F}} \tau_{\mathcal{F}}} + \frac{K}{N_p}} \\ &\leq \min \left\{ \frac{N_{\mathcal{F}} \tau_{\mathcal{F}}}{N_{\mathcal{G}} \tau_{\mathcal{G}}}, \frac{N_p}{K} \right\}. \end{aligned} \quad (4.9)$$

Eq. (4.9) illustrates that the $\mathcal{G}$ propagator must be selected carefully so as to be computationally inexpensive to keep the first bound large, but accurate enough so that K is kept small to keep the second bound large.

As was briefly mentioned in Section 1.2.3, the parareal algorithm has seen engineering applications

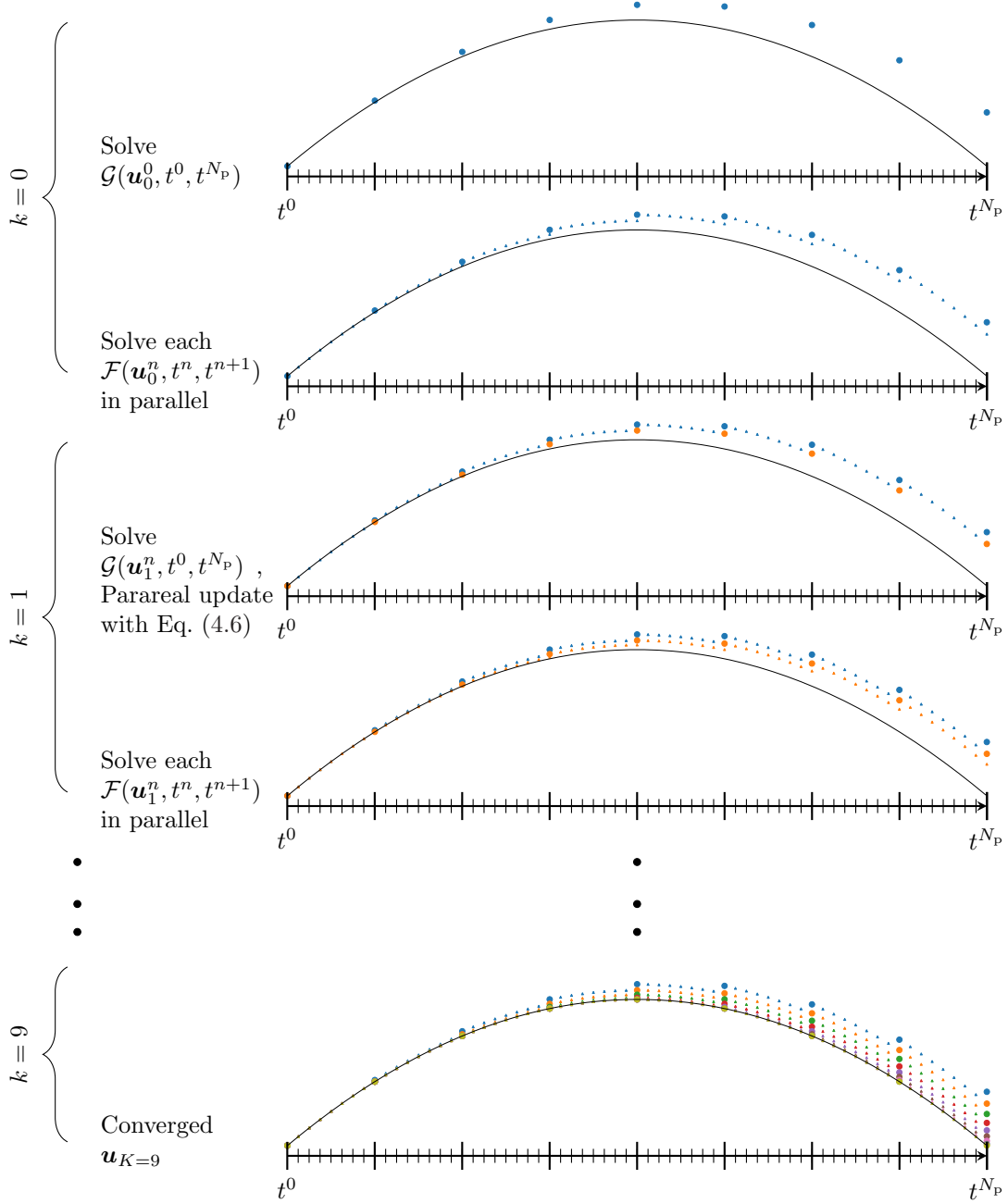


Figure 4.2: Example of parareal algorithm updates to the solution $\mathbf{u}(t)$ from both the coarse ($\mathcal{G}$) and fine $\mathcal{F}$ propagators. The analytical solution is shown as the thin black line.

for several Partial Differential Equations (PDEs), include Navier-Stokes, the heat equation, and molecular dynamics. Limited applications to neutron kinetics have been investigated. Until very recently the only applications included using the same 3D two-(energy)-group neutron diffusion solver for both $\mathcal{F}$ and $\mathcal{G}$ with rediscritized temporal domains [23], [35]. Soon to be published is a hybrid MC/deterministic strategy for simulating 1D one-group neutron diffusion [24] where a finite element diffusion solver is used as $\mathcal{G}$ and a MC

Algorithm 4.1 Parareal algorithm with coarse time propagator $\mathcal{G}$ and fine time propagator $\mathcal{F}$.

```

1:  $u_0^0 \leftarrow$  initial condition
2:  $\epsilon \leftarrow \infty$ 
3: for all  $0 \leq n \leq N_p - 1$  do
4:    $u_0^{n+1} \leftarrow$  Eq. (4.5)
5: end for
6: while  $\epsilon >$  tolerance do
7:   for all  $0 \leq n \leq N_p - 1$  do ▷ Each  $n$  is executed in parallel
8:      $u_{k+1}^{n+1} \leftarrow$  Eq. (4.6)
9:   end for
10:   $\epsilon \leftarrow \text{Norm}(\mathbf{u}_{k+1} - \mathbf{u}_k)$ 
11:   $k \leftarrow k + 1$ 
12: end while

```

diffusion model is used as $\mathcal{F}$. All of the previously-studied strategies iterate on the solution vector over the entire discretized phase space: $\mathbf{u}(x, y, z, E, t)$ for [23], [35] and $\mathbf{u}(x, t)$ for [24]. The results shown in this chapter explore the idea of coupling the $\mathcal{F}$ and $\mathcal{G}$ propagators as point kinetics parameter generators, where the solution vector would iterate over the time-dependent amplitudes and delayed neutron precursor densities, i.e.,

$$\mathbf{u}(t) = \begin{bmatrix} p(t) \\ \zeta_1(t) \\ \vdots \\ \zeta_J(t) \end{bmatrix}. \quad (4.10)$$

$\mathbf{u}(p(t), \rho(t), \zeta_j(t))$.

Conveniently, the TML method discussed in Chapter 3 is comprised of three hierarchical propagators capable of generating globally-integrated point kinetics parameters at varying levels of physics fidelity. This chapter asks the question: can the parareal algorithm be used for reactor point kinetics applications between the various physics fidelities of the TML? It will be shown that the parareal iteration between the amplitude solutions converge between both standalone Exact Point Kinetics Equations (EPKE) solvers and between the CMFD and EPKE levels of the TML. It will then be shown that, for some transients, parareal speedup factors can be achieved between two CMFD discretizations that are competitive with parallelization across the energy group.

4.1.1 Serial Implementation for Point Kinetics

While a parallel implementation of the parareal algorithm is not included in this analysis, a serial implementation was completed that shows quick convergence (within two to three outer parareal iterations) and projected parallel speedups up to ~ 4 between two point kinetics solvers. These findings provided justification

for extending the serial parareal implementation to the TML method in Section 4.2 to investigate potential parallel-in-time performance gains. To verify that the serial parareal implementation was performing as expected, a simple coupling between two EPKE propagators was implemented in C++ and can be found at <https://github.com/esgonzalez/epke-solver>. For this simple system of coupled ODEs, the parareal iteration procedure is the same as that outlined by Algorithm 4.1. Both the coarse propagator ($\mathcal{G}$) and the fine propagator ($\mathcal{F}$) use the same backward Euler temporal discretization (discussed in Section 2.4.2), and the solution vector is defined as in Eq. (4.10). The parareal iteration infrastructure does not require modification of the EPKE solver beyond an interface for setting and getting the time-dependent point kinetics solution vector $\mathbf{u}(t)$.

To begin the algorithm, the EPKEs are propagated through the entire transient in serial on the coarse propagator time scale $\Delta t_{\mathcal{G}}$. Next, the EPKE is propagated through of the coarse time step subdomains on the fine propagator time scale $\Delta t_{\mathcal{F}}$ using the power and precursor concentrations from the coarse $\mathcal{G}$ propagation as the initial condition of every temporal subdomain. Now, another coarse propagation through the entire transient is computed with $\mathcal{G}$ in serial. This second coarse propagation though is updated at each time step with Eq. (4.6) to incorporate the correction from the fine propagation with $\mathcal{F}$. Finally, the parareal fine propagations and subsequent parareal updates with Eq. (4.6) are repeated until the point kinetics power $p(t)$ and delayed neutron concentration $\zeta_j(t)$ profiles converge.

4.1.1.1 Slow Asymmetric Reactivity Ramp

The first test problem was a slow asymmetric reactivity ramp analogous perhaps to a slow control rod withdrawal followed by an insertion at a slower rate. An analytical solution can be derived using the prompt jump approximation if one delayed neutron precursor group is used (see Appendix B). The transient event was simulated with an initial amplitude condition of $p^0 = 1$ with one delayed neutron precursor ($\lambda = 0.49405 \text{ s}^{-1}$ and $\beta^{\text{eff}} = 0.0076$) where the initial concentration is specified by Eq. (2.55c). A nearly instantaneous effective neutron generation time of $\Lambda = 2.6 \times 10^{-15} \text{ s}$ was used to simulate the prompt jump approximation that $\Lambda \rightarrow 0$. The reactivity profile shown in Fig. 4.3 was used to drive the transient. Other than the reactivity, all other point kinetics parameters were held constant throughout the transient event.

The amplitude profiles for the serial parareal solution of the asymmetric reactivity ramp problem are shown in Fig. 4.4 for a coarse time step $\Delta t_{\mathcal{G}} = 1 \text{ s}$. The corresponding effective delayed neutron precursor concentration profile $\zeta(t)$ is shown in Fig. 4.5. For both EPKE to EPKE coupling test problems, it is assumed that there will be only one coarse integration per processing unit $N_{\mathcal{G}} = 1$, meaning that for the six second transient $N_{\text{p}} = 6$. For the results presented in Fig. 4.4, the number of fine integrations per processing unit is

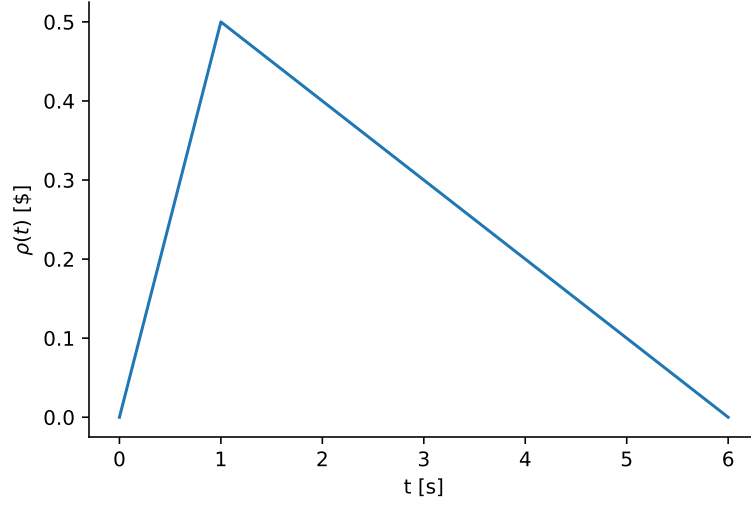


Figure 4.3: Parareal reactivity profile for asymmetric reactivity ramp with one delayed neutron precursor group.

arbitrarily set to be $N_{\mathcal{F}} = 50$.

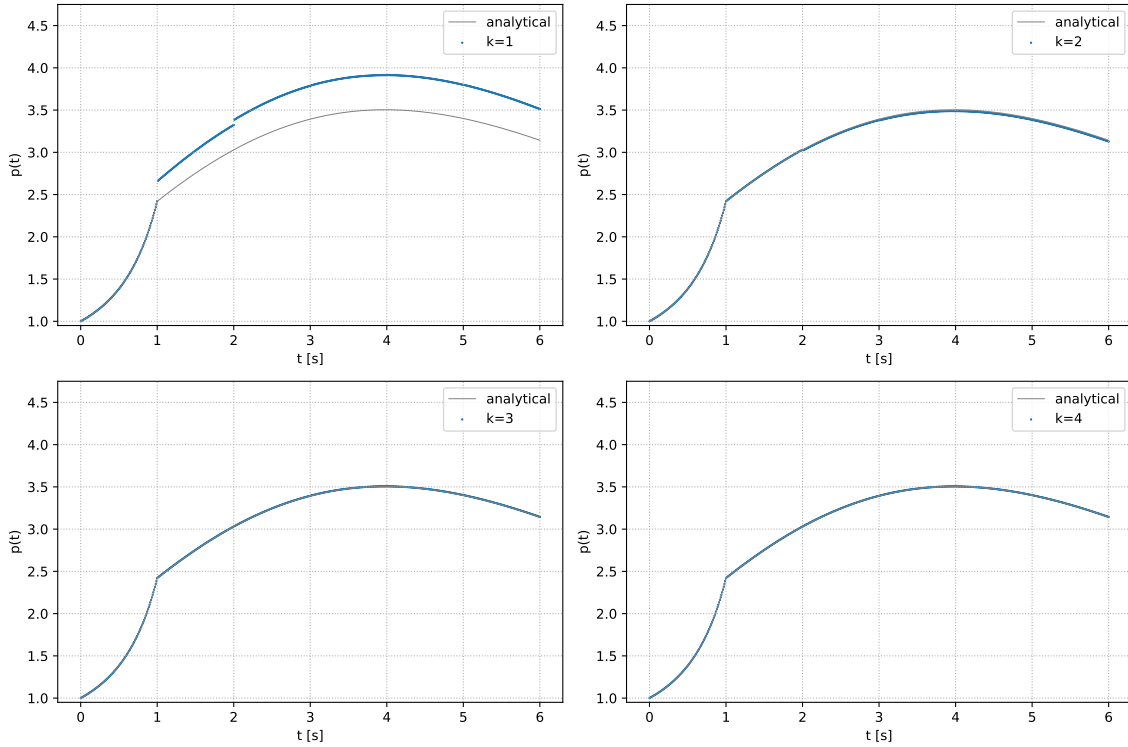


Figure 4.4: Parareal power profile for asymmetric reactivity ramp with one delayed neutron precursor group.

Figure 4.6 shows a convergence analysis of the parareal problem (left) and the projected parareal speedup

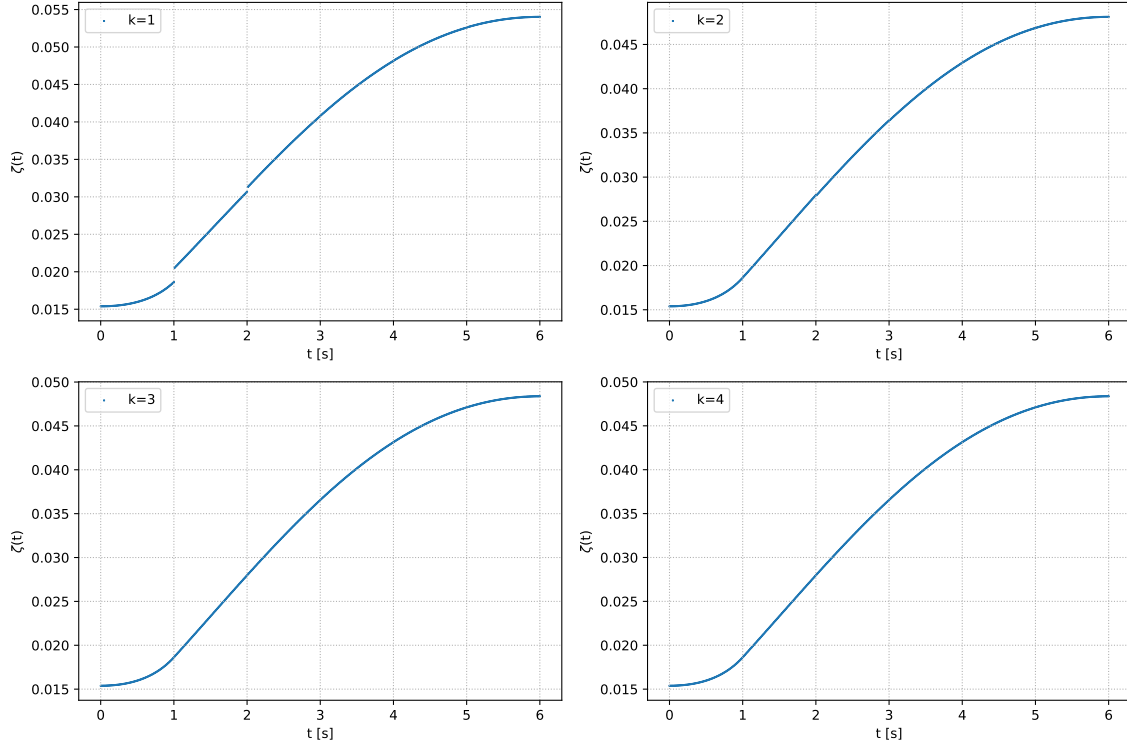


Figure 4.5: Parareal delayed neutron precursor concentration profile for asymmetric reactivity ramp.

(right) using Eq. (4.9) for a range of ratios $N_{\mathcal{F}}/N_{\mathcal{G}}$. To assess convergence, the L_2 norm of the difference between the parareal solution after k iterations $p(t)_k$ and the analytical solution $p(t)_{\text{analytical}}$ divided by the number of fine time steps $N_p N_{\mathcal{F}}$ was used. As expected, for increasing k , the L_2 norm converges towards zero until leveling off. For increasing $N_{\mathcal{F}}/N_{\mathcal{G}}$, the number of parareal iterations needed to reach convergence also increases. To estimate the projected parareal speedup, the serial computation time to propagate $\mathcal{F}(p^0, t^0, t^N)$ was computed using C++ embedded timers and $\tau_{\mathcal{F}}/\tau_{\mathcal{G}}$ was assumed to be unity.

For this particular transient, parareal speedup $1 < S_{\text{parareal}} \leq 2$ is achieved for $N_{\mathcal{F}}/N_{\mathcal{G}} > 5$. This speedup is not particularly impressive, but it was anticipated because the first limiting condition from Eq. (4.9) is directly dependent on the ratio of $\tau_{\mathcal{F}}/\tau_{\mathcal{G}}$ being large. In other words, for larger speedup factors, the computational expense of advancing the fine propagator a single time step should be much more expensive than the computational expense of advancing the coarse propagator a single time step. Unless the parallel implementation has issues scaling (which has yet to be studied for reactor transient applications), the ratio $N_{\mathcal{F}}/N_{\mathcal{G}}$ can be made arbitrarily large for arbitrarily large speedup depending on the degree of accuracy desired by the engineer.

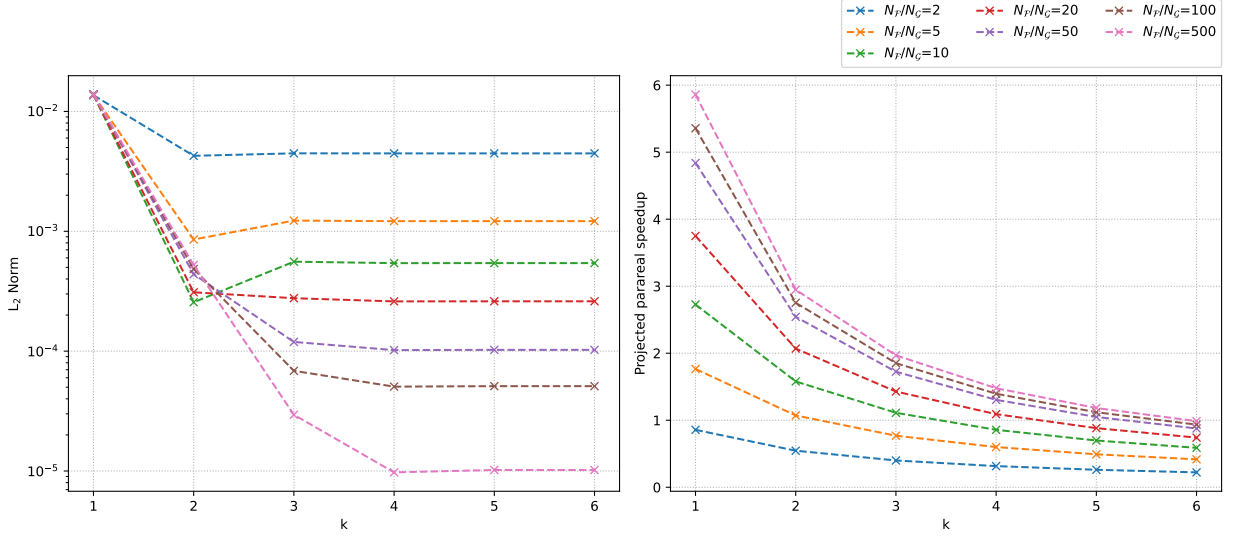


Figure 4.6: Convergence (left) and relative computational time (right) between serial parareal implementation and serial backward Euler implementation for reactivity ramp problem at various ratios of $\Delta t_{\text{coarse}}/\Delta t_{\text{fine}}$.

4.1.1.2 Fast Control Rod Ejection

A second problem was tested with the serial EPKE coupling. This time, a prompt supercritical reactivity excursion was modeled where the imposed reactivity profile was obtained from a PARCS [109] simulation of a control rod ejection and can be seen in Fig. 4.7. Six delayed neutron precursor groups were used to simulate the transient. The corresponding effective decay constants and delayed neutron fractions are given in Table 4.1. Instead of comparing against an analytical solution, direct comparison of the parareal solution with the PARCS amplitude profile is presented in Fig. 4.8 for a coarse time step of $\Delta t_G = 5$ ms. The prompt supercritical transient requires a much smaller time step than the previous slow reactivity ramp. The corresponding delayed neutron precursor profiles for each of the six precursor groups are shown in Fig. 4.9. It was assumed again that there is one processing unit available for each coarse time step. For a one second transient, this gives $N_p = 200$ and $N_G = 1$. The number of fine time propagations per processing unit was set to be $N_F = 20$.

Table 4.1: Six-group delayed neutron parameters used for control rod eject transient.

j	1	2	3	4	5	6
λ_j	0.0128	0.0318	0.119	0.3181	1.4027	3.9286
$\beta_j(\%)$	0.025 84	0.152	0.139 08	0.307 04	0.1102	0.025 84

The convergence plot shown in Fig. 4.10 appears to follow the same trends as were described for the

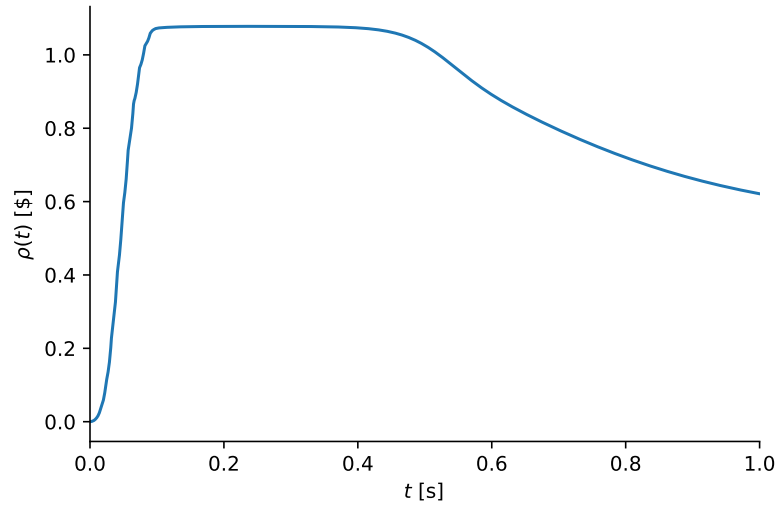


Figure 4.7: Parareal reactivity profile for control rod ejection simulated with PARCS.

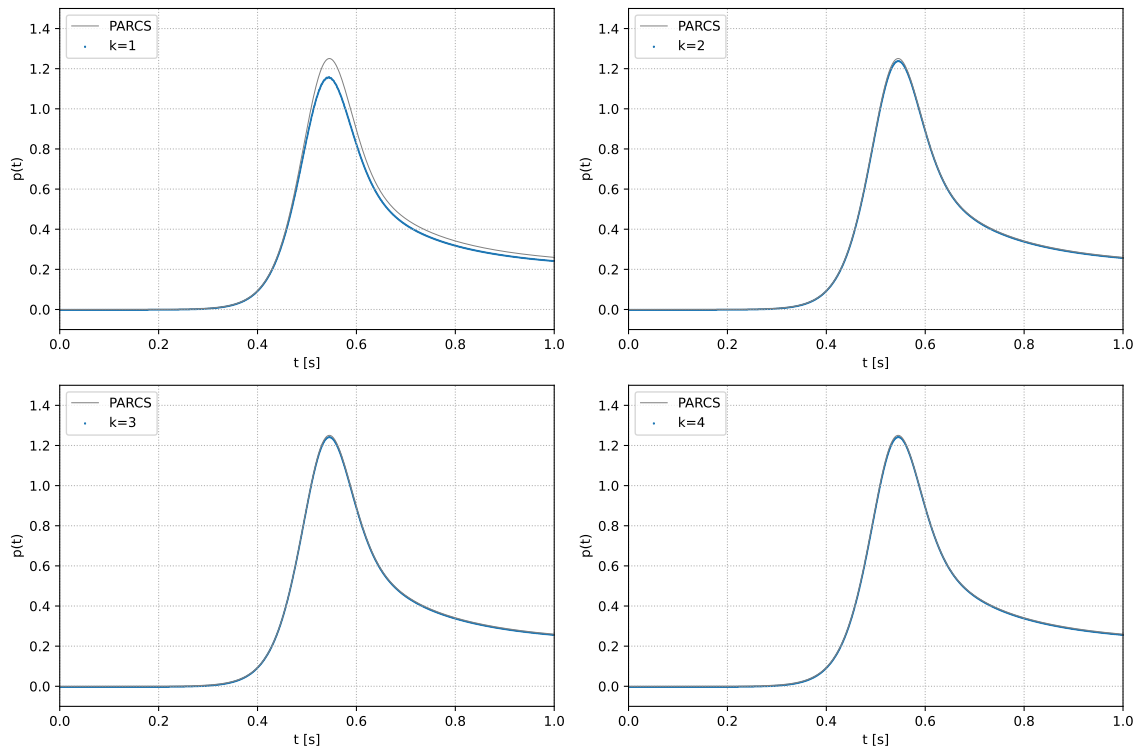


Figure 4.8: Parareal power profile for control rod ejection problem.

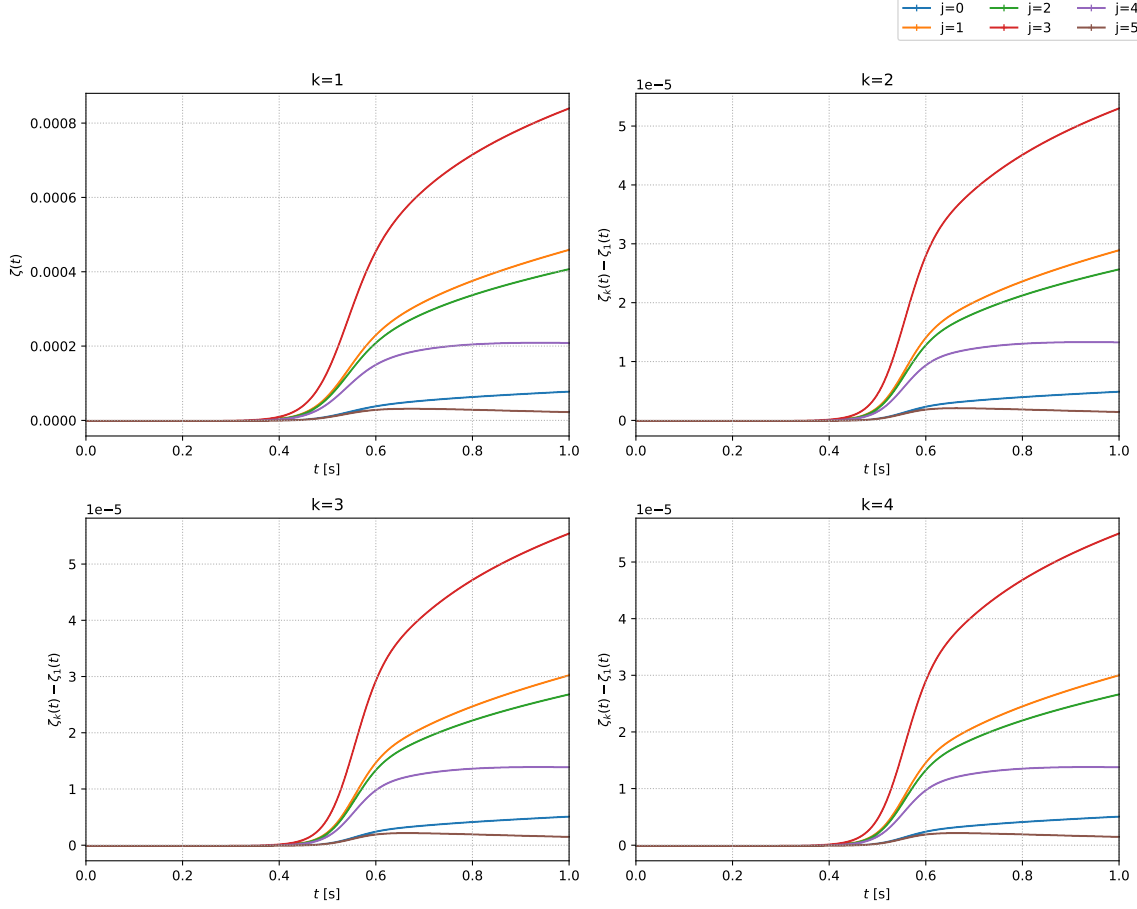


Figure 4.9: Parareal delayed neutron precursor concentration profiles for control rod ejection problem. The top-right and bottom plots show the difference between the precursor concentration profiles at k versus $k = 1$.

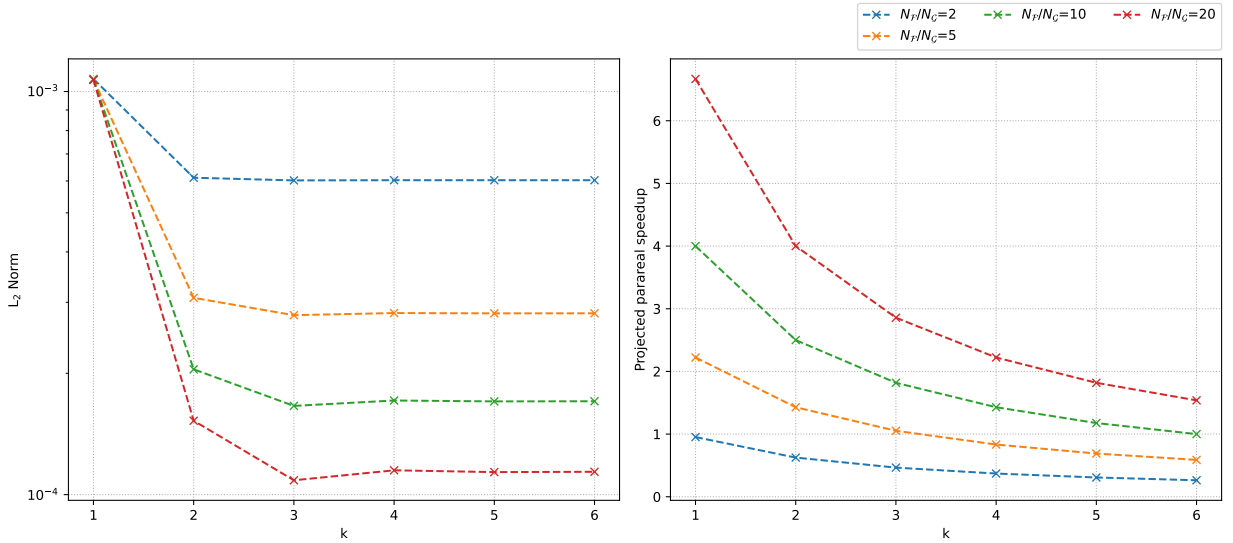


Figure 4.10: Convergence (left) and relative computational time (right) between serial parareal implementation and serial backward Euler implementation for control rod ejection problem at various ratios of $\Delta t_{\text{coarse}}/\Delta t_{\text{fine}}$.

previous reactivity ramp problem. For this transient, the PARCS reference solution limited the study to a fine time step resolution of $\Delta t_{\mathcal{F}} = 250 \mu\text{s}$. Also of note is that the projected parareal speedup estimates are noticeably higher for this transient than for the slow reactivity ramp, largely in part to the assumption that there is a one to one ratio between the number of available processing units and number of coarse time steps throughout the transient, i.e., 200 versus 6. This might also be a consequence of including more delayed neutron precursor groups in the calculation (six for the control rod ejection, versus one for the reactivity ramp). While the projected speedup values are only estimates and assume no additional overhead from communication between processing units, the observation that convergence appears to happen for very few parareal iterations is certainly promising. These results provide justification for the next section, where the parareal infrastructure is implemented in Shift to study parallel-in-time acceleration schemes of the TML.

4.2 Serial Implementation for TML Subsolders

After the parareal algorithm was confirmed to have been working in serial for the two standalone EPKE solvers (a very simple system of coupled ODEs), it was implemented into the Shift TML framework. The parareal iteration infrastructure is largely self-contained and only requires passing of the time-dependent solution $\mathbf{u}(t)$ computed by the various solvers nested within the TML. As a result, minimal structural changes are required to implement the parareal algorithm beyond those already required for the TML. For this work, the iterated solution in all of the proposed parareal coupling schemes is the EPKE amplitude function from the quasi-static factorization of the CMFD scalar flux. Therefore, the solution vector $\mathbf{u}(t)$ is defined by the unknown quantities from the EPKEs given by Eq. (4.10).

The MC, CMFD, and EPKE solvers are all independently capable generating estimates of point kinetics quantities of interest ($p(t)$, $\rho(t)$, and $\zeta_j(t)$) throughout the transient. While there is a clear hierarchy in the physics fidelities of the various solvers, it is difficult to say with certainty that one solver's estimates for the true point kinetics quantities of the system are more accurate than the other solvers' because they all use different time steps. For example, the MC solver is able to produce very accurate estimates of integrated quantities at any given point in time, but if too large of a time step is taken, the fission source distribution might diverge and point kinetics estimates cannot be trusted. On the other hand, EPKE solutions are able to be propagated forward in time at very small time scales, but do not retain any of the spatial or energy dependence of the system's true flux distribution throughout the transient.

The first parareal coupling tested was between the CMFD ($\mathcal{G}_{\text{CMFD}}$) and EPKE ($\mathcal{F}_{\text{EPKE}}$) levels. As a reminder, Fig. 4.12 shows the flow of information between the TML levels that was presented in Chapter 3. The order in which the various solvers are evaluated throughout the transient for the TML without parareal

coupling is depicted by the circled numbers. In contrast, the same flow of information and corresponding order of solver evaluations for the parareal coupling are depicted in Fig. 4.11.

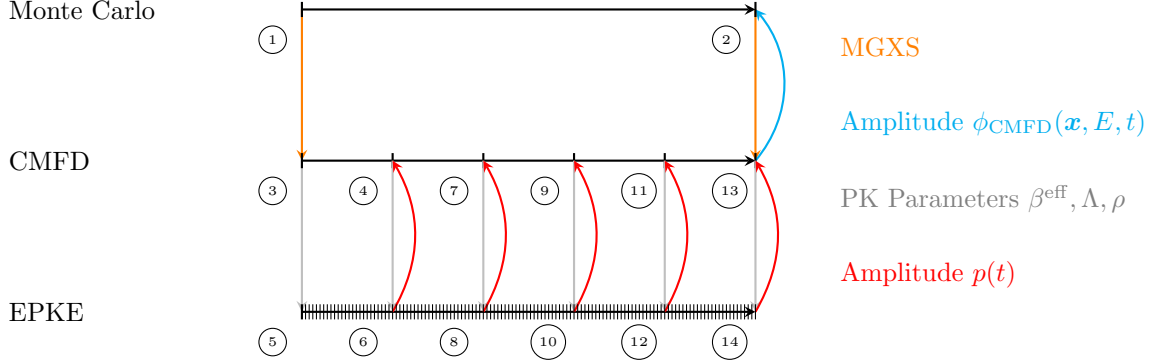


Figure 4.11: Flow of information between unmodified TML levels. The circled numbers indicated the order in which the various solvers are evaluated throughout the transient.

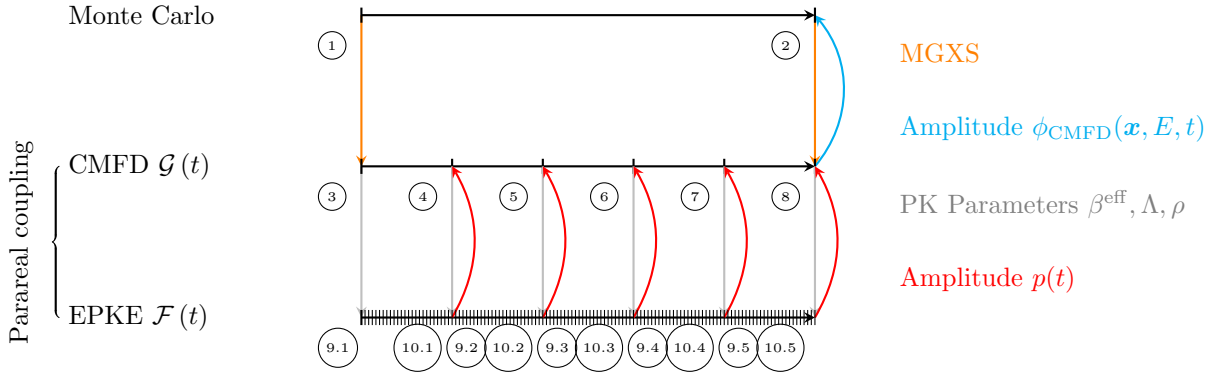


Figure 4.12: Flow of information between parareal coupling of CMFD and EPKE levels. The circled numbers indicated the order in which the various solvers are evaluated throughout the transient.

With this parareal coupling, the algorithm for a single MC time step begins by evaluating the MC solver at $t = t^n$ (①) and propagating forward the solution a single time step on the Δt_{MC} time scale (②). Up until this point, the algorithm is identical to the TML method described in Chapter 3. From here, the CMFD solution is propagated forward in time on the Δt_{CMFD} time scale in serial (③ – ⑧) without being “corrected” by the EPKE amplitude function. From these propagated CMFD solutions, point kinetics parameters are computed on the Δt_{CMFD} time scale and denoted as $\mathcal{G}_{CMFD}(\mathbf{u}^n, t^n, t^{n+1})$. Next, the EPKE amplitude solver is propagated forward in time on the Δt_{EPKE} time scale for all of the CMFD sub-intervals simultaneously (⑨ – ⑩ in parallel) and denoted as $\mathcal{F}_{EPKE}(\mathbf{u}^n, t^n, t^{n+1})$. After the parallel EPKE propagations, the parareal outer update is computed with Eq. (4.6) and the outer iteration index is advanced.

The results for the parareal coupling are presented in Fig. 4.13 for the C5G7-TD3-4 transient with an MC time step of $\Delta t_{MC} = 0.5$ s, CMFD time step of $\Delta t_{CMFD} = 0.1$ s, and EPKE time step of $\Delta t_{EPKE} = 4$ ms.

For all C5G7-based results shown in this chapter there were 1×10^5 particle histories simulated per fission source iteration cycle in the MC solves, with 150 inactive cycles and 250 active cycles. Table 4.2 shows L_∞ , L_2 , and L_1 norm comparisons for the fractional fission rate profiles for each parareal iteration. The norms improve up until five parareal iterations. After this point, there were no appreciable reductions in error between the serial TML calculation and the serial parareal coupling. While it is encouraging that parareal convergence is possible between various solver levels, this selection of $\mathcal{G}_{\text{CMFD}}$ and $\mathcal{F}_{\text{EPKE}}$ does not provide any computational speedup because here $\mathcal{G}_{\text{CMFD}}$ is more expensive than $\mathcal{F}_{\text{EPKE}}$.

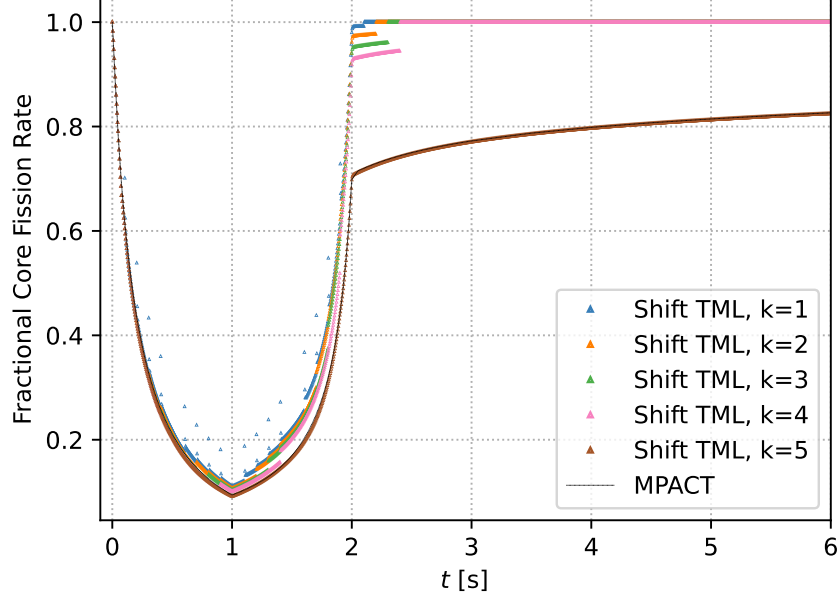


Figure 4.13: Parareal fission rate with coupling between CMFD and EPKE levels.

Table 4.2: Norm comparisons (given in %) to MPACT solution for parareal coupling between CMFD and EPKE levels.

Norm	k				
	1	2	3	4	5
$L_\infty = \max_n \left[\frac{p_{n,\text{parareal}} - p_{n,\text{MPACT}}}{p_{n,\text{MPACT}}} \right]$	41.77	38.92	35.65	33.87	4.90
$L_2 = \frac{1}{N} \left[\sum_n \left(\frac{p_{n,\text{parareal}} - p_{n,\text{MPACT}}}{p_{n,\text{MPACT}}} \right)^2 \right]^{\frac{1}{2}}$	2.48	2.20	1.92	1.65	0.23
$L_1 = \frac{1}{N} \sum_n \left \frac{p_{n,\text{parareal}} - p_{n,\text{MPACT}}}{p_{n,\text{MPACT}}} \right $	20.8	17.73	15.22	12.83	2.38

4.2.1 Introducing an Additional 1-Group CMFD Level

In the previous section, the CMFD propagations from the TML were used as the coarse time propagator $\mathcal{G}_{\text{CMFD}}$ and the EPKE propagations from the TML were used as the fine time propagator $\mathcal{F}_{\text{EPKE}}$. This

selection of coarse and fine time propagators does not provide parallel-in-time speedup over the serial TML because the cost of propagating the CMFD equations forward in time is more computationally expensive than the cost of propagating the EPKEs. For parareal speedup to be possible, the coarse propagator must be less expensive than the fine propagator. One way to accomplish this, is by adding an additional level between the MC and CMFD levels that is less computationally expensive than the standard CMFD propagations.

For the C5G7 transient problems being studied, seven energy groups are used to solve both the MC and CMFD multigroup calculations. One choice for this additional TML level is to use the same CMFD solver with a collapsed energy group structure, i.e., the energy dependence is integrated out of the CMFD equations. Shen saw performance improvement to the serial TML by introducing an additional CMFD level with a collapsed energy structure (one energy group) to the TML between the multigroup CMFD and EPKE levels [14]. In reality, the CMFD solution method implemented in Shift is parallelized across energy groups. Therefore, for supercomputing applications in which processing units are plentiful, speedup of a parallel parareal implementation would be reduced. However, for a computing environment limited by the number of available processors, this choice for parareal coupling could prove advantageous if the number of parareal iterations is very small.

A modified flow of information for the additional TML level is depicted in Fig. 4.14. Just as with the previous parareal coupling depicted in Fig. 4.12, the algorithm for a single MC time step begins by evaluating the MC solver at $t = t^n$ ((1)) and propagating forward the solution a single time step on the Δt_{MC} time scale ((2)). This time though, the coarse propagator, $\mathcal{G}_{CMFD-1G}$, includes both the CMFD-1G and EPKE levels computed in serial ((3)-(6)). In effect, the first three levels in Fig. 4.14 are computed in the same sequence as the unmodified TML with no parareal coupling. The parareal solution vector is still defined by the unknown quantities from the EPKEs given by Eq. (4.10).

The fine propagator, $\mathcal{F}_{CMFD-7G}$ includes both the CMFD-7G and another EPKE level. However, now all of the CMFD-7G subintervals (and corresponding EPKE amplitude propagations) are computed in parallel ((7)-(10)). For each subinterval, the initial conditions necessary for the EPKE amplitude propagations come from the coarse time EPKE amplitude solution at the $\mathcal{G}_{CMFD-1G}$ propagator. Obviously, the time step being used between the CMFD-1G and CMFD-7G propagators will differ ($\Delta t_{CMFD-1G} > \Delta t_{CMFD-7G}$). In addition, the quality of the point kinetics parameters generated for the EPKE amplitude propagations will differ between the two CMFD levels as well, where the parameters generated by the CMFD-7G solver are expected to be more accurate. After the parallel $\mathcal{F}_{CMFD-7G}$ propagations have completed, the EPKE amplitude vector $\mathbf{u}(t)$ is updated via Eq. (4.6).

Figure 4.15 shows the results for the coupling between CMFD levels with $\Delta t_{MC} = \Delta t_{CMFD-1G} = 0.5$ s, $\Delta t_{CMFD-7G} = 0.1$ s, and $\Delta t_{EPKE} = 4$ ms. Also plotted is the serial TML solution without parareal coupling

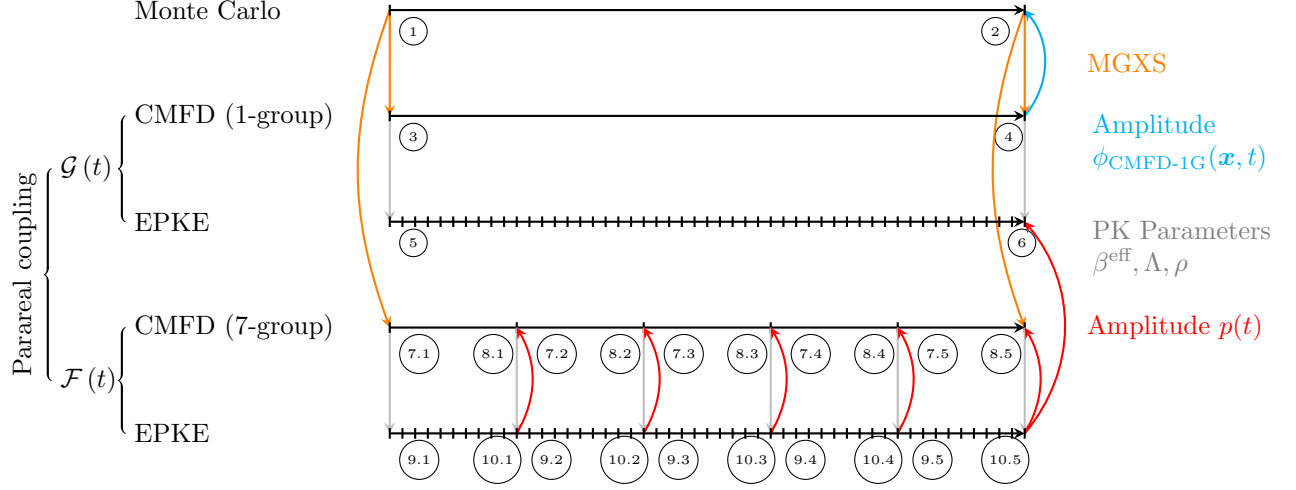


Figure 4.14: Parareal flow of information for coupling between CMFD-7G and CMFD-1G levels with $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 5$.

or the additional CMFD-1G level for the same selection of solver time steps and a serial TML solution with refined time steps $\Delta t_{\text{MC}} = 0.25 \text{ s}$, $\Delta t_{\text{CMFD}} = 0.05 \text{ s}$, and EPKE time step of $\Delta t_{\text{EPKE}} = 2 \text{ ms}$. The scheme shows very fast convergence towards the reference solution after a single parareal iteration. This is likely because the shape function with just spatial resolution calculated by the CMFD-1G solver is sufficient to define the point kinetics parameters used for the EPKE amplitude propagation. Appreciable reductions in error were not observed beyond $k = 2$. Of note, the parareal solution for this simulation does not converge towards the serial TML solution. Instead, the parareal iterations converge towards the reference solution with the refined solver time steps. Figure 4.16 shows the same C5G7-TD3-4 simulation with an increased ratio of $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 10$. This time, with an increased number of CMFD-7G solver, the serial and reference solutions align.

The CMFD-7G to CMFD-1G coupling was also tested for a positive reactivity ramp for the C5G7 geometry to determine whether the same convergence from the negative reactivity ramp with the C5G7-TD3-4 transient could be achieved. For this new transient, the moderator density was increased linearly for one second from its initial fractional density of 1 to 1.03. Then from $1 \text{ s} \leq t \leq 2 \text{ s}$, the reactivity was linearly decreased back down to its initial value. The same number of histories per cycle, inactive and active fission cycles were used as in the previous C5G7 transients presented in this chapter. The MC and CMFD-1G time steps were selected to be $\Delta t_{\text{MC}} = \Delta t_{\text{CMFD-1G}} = 0.25 \text{ s}$ with the ratio of the CMFD-1G to CMFD-7G time steps $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 10$. The results are presented in Fig. 4.17 and display a similar rapid convergence after only a single parareal iteration $k = 1$.

Figure 4.18 shows the projected speedup of the CMFD levels with this selection of parareal coupling

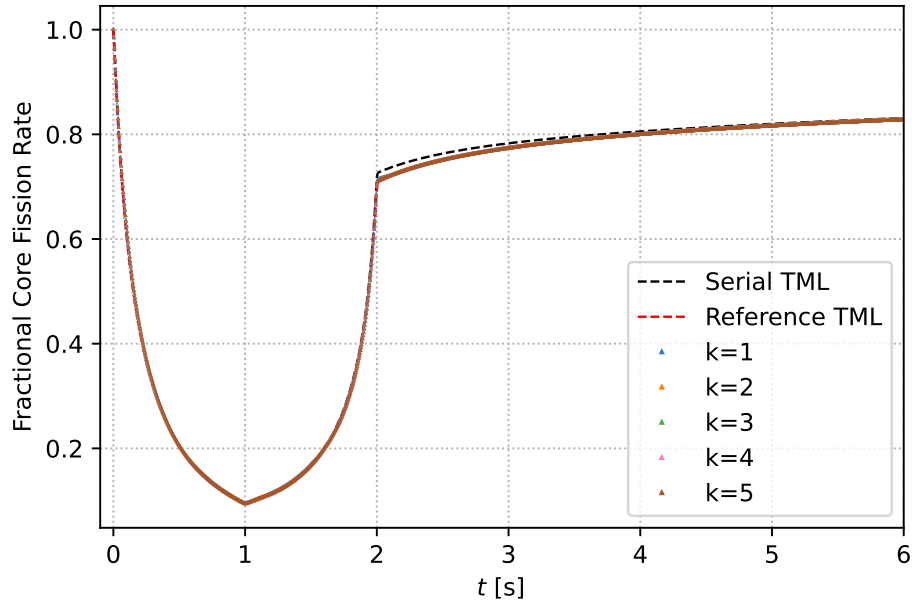


Figure 4.15: Fractional core fission rate for C5G7-TD3-4 problem with serial parareal between CMFD-7G and CMFD-1G levels and $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 5$.

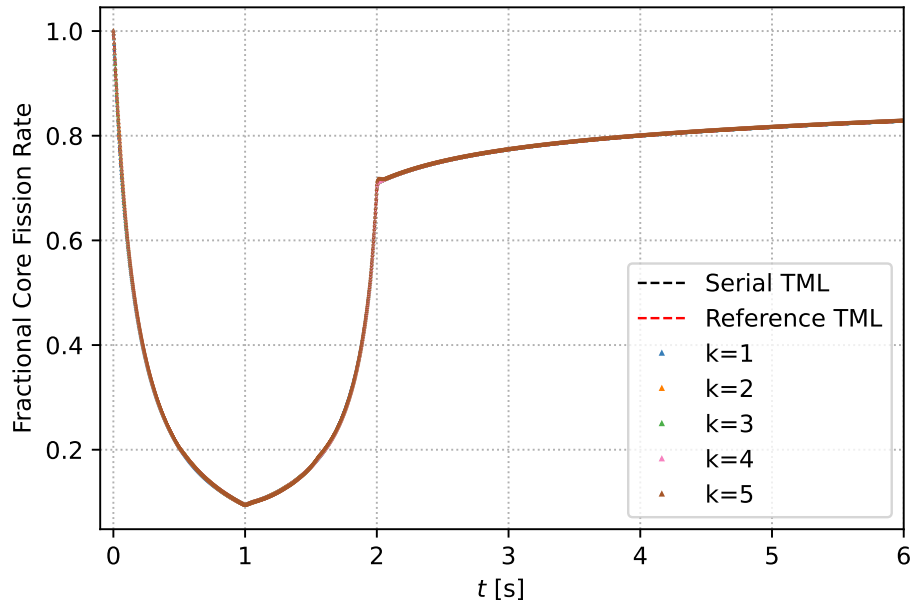


Figure 4.16: Fractional core fission rate for C5G7-TD3-4 problem with serial parareal coupling between CMFD-7G and CMFD-1G levels and $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 10$.

assuming overhead for the calculation is zero. The plot on the left shows speedup calculations assuming that there are sufficient processing units available to use parallelization across both energy groups and time

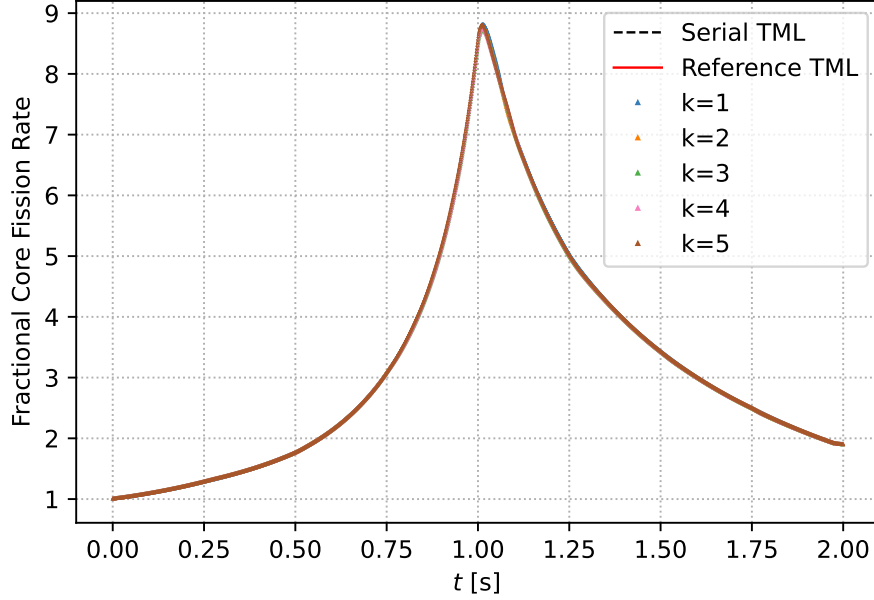


Figure 4.17: Fractional core fission rate for C5G7 positive reactivity ramp with serial parareal coupling between CMFD-7G and CMFD-1G levels and $\Delta t_{\text{CMFD-7G}}/\Delta t_{\text{CMFD-1G}} = 10$.

steps, i.e., for supercomputing applications. The plot on the right shows projected speedups assuming that parallelization across energy groups is not used for the CMFD-7G calculation. For the case on the left, speedups of ~ 1.5 to ~ 3.5 are projected for $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 5$ and 10, respectively, if convergence after a single parareal iteration $k = 1$ is assumed. For the case presented on the right, speedups of ~ 4 to ~ 8 are projected for $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 5$ and 10, respectively. Note that a speedup of 7 or greater would prove to be more efficient than only using parallelization across the energy groups. Given that the parareal solution converges quickly to the correct solution and that there is no shortage of processing units, it seems that there is no reason not to use a parareal implementation to improve overall computational cost of the subsolvers in the TML hierarchy.

A summary of the parareal couplings tested throughout this chapter is shown in Table 4.3. The projected speedups are given in the last column, assuming that the available processing units are devoted to parallelization of the temporal domain. The largest potential speedups are observed after adding the additional CMFD level with a collapsed energy structure between the MC level and the CMFD level with seven energy groups. For this coupling, the coarse propagator $\mathcal{G}_{\text{CMFD-1G}}$ is less computationally expensive (by a factor of seven) than the fine propagator $\mathcal{F}_{\text{CMFD-7G}}$. While these results are promising, it was shown in Chapter 3 that the overall simulation time is dominated by the MC level. The next section discusses one possible scheme for generating coarse-level MC solutions with correlated sampling to be used as the coarse propagator, $\mathcal{G}_{\text{MC-CS}}$.

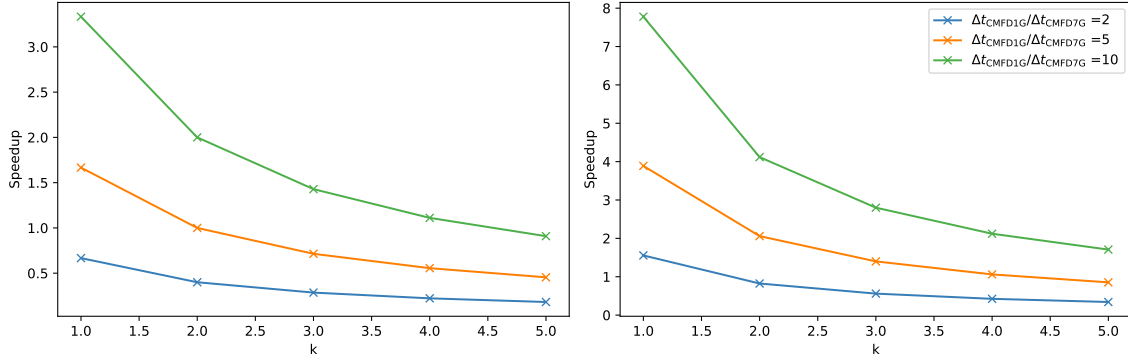


Figure 4.18: Projected speedup of parallel parareal coupling between CMFD-7G and CMFD-1G levels for various values of $\Delta t_{\text{CMFD-7G}}/\Delta t_{\text{CMFD-1G}}$. The left plot shows speedup if the CMFD-7G solution is additionally parallelized across energy groups, while the right shows speedup if no additional parallelization is used.

Table 4.3: Summary of serial parareal problems simulated.

Transient	$\mathcal{G}$	$\mathcal{F}$	Projected Speedup
Slow asymmetric ramp	EPKE	EPKE	1 – 3
Fast rod ejection	EPKE	EPKE	1.5 – 4
C5G7 TD3-4	CMFD-7G	EPKE	No speedup
C5G7 TD3-4	CMFD-1G	CMFD-7G	1.5 – 7.9
C5G7 positive ramp	CMFD-1G	CMFD-7G	1.5 – 7.9

4.2.2 Coupling Between Monte Carlo Levels

Projected speedups of the CMFD level are promising and warrant further investigation with a parallel implementation of the parareal schemes proposed. However, it was shown in Chapter 3 that practical applications of the MC TML implementation are burdened by the reality that upwards of 95% of the computational expense is attributed to the MC solver. In order for the parareal algorithm to make meaningful reductions in computational cost, it would need to be coupled between two different MC levels. Direct parareal couplings between two Monte Carlo solutions are not ideal, as stochastic noise coming from the coarse and fine propagators make convergence of the final solution difficult to ascertain. Additionally, convergence for a coupled system of integro-differential equations (that include the hyperbolic streaming operator) is not guaranteed by the parareal algorithm.

One of the advantages of the quasi-static approximation that the TML built upon is that it offloads a lot of the computational burden that comes with advancing the solution forward in time to the deterministic CMFD and EPKE subsolvers as *amplitude* functions. If parareal couplings are then focused on accelerating the computation of amplitude functions, as they were in the previous sections, the complexity of the coupled

system is reduced to a diffusion problem (if using $\mathbf{u}(t)$ from the CMFD level) or a point kinetics problem (if using $\mathbf{u}(t)$ from the EPKE level). For a parareal coupling scheme where the parareal solution vector corresponds to the unknown quantities from the CMFD equations, the parareal solution vector would correspond to

$$\mathbf{u}(\mathbf{x}, E, t) = \begin{bmatrix} \phi(\mathbf{x}, E, t) \\ C_1(\mathbf{x}, t) \\ \vdots \\ C_J(\mathbf{x}, t) \end{bmatrix}. \quad (4.11)$$

There are several ways that one might go about choosing a coarse-level MC solver that is less computationally expensive than the current top-level MC solver. If the MC level in the traditional TML is equipped to handle continuous-energy physics ($\mathcal{F}_{\text{MC-CE}}$), then a multigroup MC solution might suffice as the coarse-level MC solution ($\mathcal{G}_{\text{MC-MG}}$). Alternatively, one could simply reduce the number of particle histories per fission source iteration cycle or reduce the number of iteration cycles to reduce the computation cost of the coarse-level MC solution ($\mathcal{F}_{\text{MC-more histories}}$ versus $\mathcal{G}_{\text{MC-fewer histories}}$).

The rest of this chapter investigates the feasibility of using a correlated sampling technique to provide estimates of perturbed reaction rate, surface current, and point kinetics tallies as the coarse-level MC solution. These approximate time-propagated Monte Carlo tallies of integrated quantities can then be used as input to the CMFD and EPKE subsolvers of the TML method to propagate the solution forward in time with a coarse MC time step as $\mathcal{G}_{\text{MC-CS}}$. The fine time TML propagations, computed using standard direct MC evaluations ($\mathcal{F}_{\text{MC-direct}}$) on temporal subintervals, could then be computed in parallel with a refined MC time step to correct the $\mathcal{G}_{\text{MC-CS}}$ amplitude solution.

The proposed algorithm for a single MC time step uses the full TML as the coarse propagator. For the coarse $\mathcal{G}_{\text{MC-CS}}$ propagations, all MC evaluations will be computed at the same time, i.e., the MC tallies at (2) will be estimated via correlated sampling from the direct MC simulation at (1). The input parameters for the CMFD and EPKE subsolvers will then be interpolated to propagate the amplitude solvers forward in time ((3)-(12)). After these $\mathcal{G}_{\text{MC-CS}}$ propagations are complete, the fine $\mathcal{F}_{\text{MC-direct}}$ propagations ((13)-(18)) will be computed in parallel. Finally, the CMFD amplitude function given by Eq. (4.11) will be iterated using the standard parareal update given by Eq. (4.6).

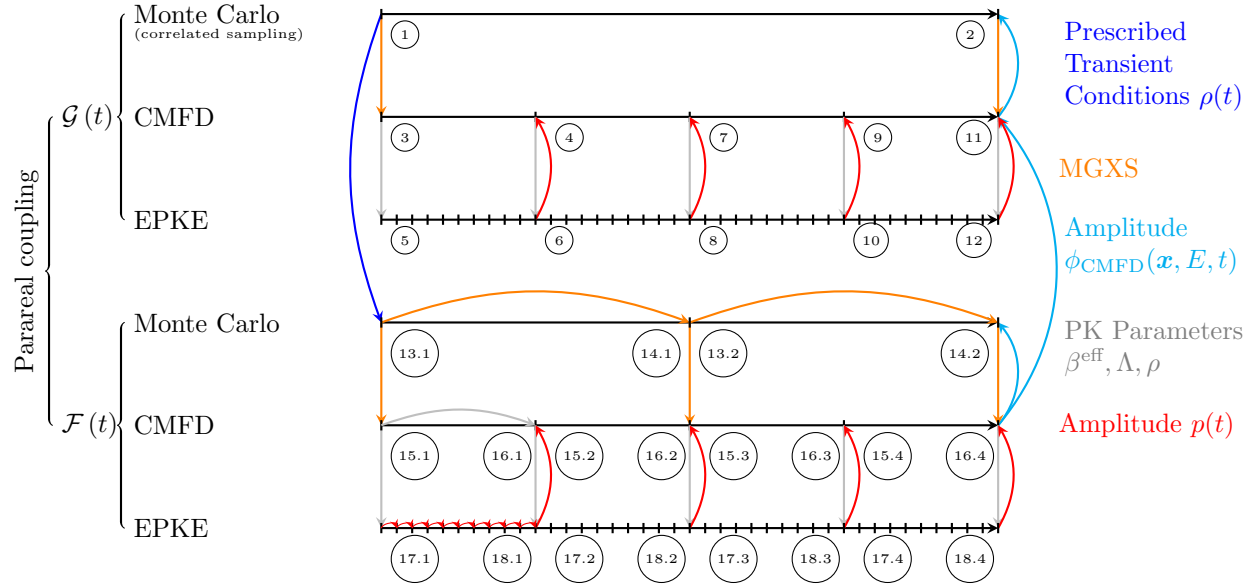


Figure 4.19: Proposed flow of information for coupling between two MC levels.

4.3 Correlated Sampling

Correlated sampling is a Monte Carlo perturbation technique for estimating the effects that small changes to system parameters (perturbations) have on tallies of integrated quantities of interest. Perturbations might include small changes to material densities, nuclear data, or geometry features. Sometimes dozens or even hundreds of perturbations might be necessary to design a new experiment or reactor. Obviously, it would be very time-consuming and inefficient if a new simulation has to be run for each perturbation. Additionally, for the MC method, small changes to system parameters are often difficult to quantify because of the statistical uncertainties that might appear between two independent and uncorrelated results. For example, suppose the Monte Carlo estimate for the leakage probability from the surface of some fictitious sphere of fissile material is 0.1 ± 0.05 . Now suppose that the density of the fissile material in the simulation is increased by 2% and the new Monte Carlo estimate of the leakage probability is 0.07 ± 0.04 . For this hypothetical scenario, more-costly MC simulations would need to be run with a larger number of particle histories before any engineering decisions could be made because the uncertainties for both calculations overlap with the mean value estimates.

Correlated sampling offers a unique solution to this problem through perturbation estimates of the Monte Carlo tallies of interest. The correlated sampling technique, for a single MC simulation, produces $N_{\text{pert}} + 1$ Monte Carlo results, where N_{pert} is the number of perturbations specified, and the extra result corresponds to that of the unperturbed system. All $N_{\text{pert}} + 1$ of these results use the same set of pseudo-random numbers

and therefore are subject to the same random walk sequence. The perturbations are estimated by keeping track of adjusted weight ratios throughout the random walk. These weight ratios account for differences in the streaming and collision probabilities between the perturbed and unperturbed systems.

4.3.1 Derivation of the Direct and Indirect Perturbation Effects

The mathematics derivation for the correlated sampling technique begins by letting R be a response of interest during the particle transport process that is dependent on model parameter α

$$R(\alpha) = \int_h s_h(\alpha) p_h(\alpha) dh. \quad (4.12)$$

In Eq. (4.12), $R(\alpha)$ is expressed as an integral over the continuous random walk space, where h is the index of an individual random walk. The integrand is the product of a scoring function $s_h(\alpha)$ and the Probability Density Function (PDF) of an individual random walk occurring, $p_h(\alpha)$. The PDF for an individual random walk can further be expanded as the product of the N_c tracks/steps/collisions that compose the random walk, each occurring with probability p_i

$$p_h(\alpha) = \prod_{i=1}^{N_c} p_i(\alpha). \quad (4.13)$$

Now let $R'(\alpha')$ be the response corresponding to a perturbation in the model parameter $\Delta\alpha = \alpha' - \alpha$

$$\begin{aligned} R'(\alpha') &= \int_{h'} s'_{h'}(\alpha') p'_{h'}(\alpha') dh' \\ &= \int_{h'} s'_{h'}(\alpha') \prod_{i'=1}^{N'_c} p'_{i'}(\alpha') dh'. \end{aligned} \quad (4.14)$$

Note that h' in the perturbed system is a distinct random walk from h in the unperturbed system. Additionally, both the scoring function, $s'_{h'}$, and the PDF for an individual random walk, $p'_{h'}$, are dependent on the perturbed model parameter, α' . To obtain an unbiased estimate of the change in the response $\Delta R = R' - R$, two separate sequences of random walks need to be simulated. Instead, the correlated sampling technique uses only the unperturbed random walk to estimate both the perturbed and unperturbed responses. Therefore,

Eq. (4.14) is approximated as

$$\begin{aligned}
R'(\alpha') &\approx \int_h s'_h(\alpha') \prod_{i=1}^{N_c} p'_i(\alpha) \, dh \\
&= \int_h s'_h(\alpha') \prod_{i=1}^{N_c} \underbrace{\frac{p'_i(\alpha')}{p_i(\alpha)}}_{a_i} p_i(\alpha) \, dh \\
&= \int_h s'_h(\alpha') \prod_{i=1}^{N_c} a_i p_i(\alpha) \, dh \\
&= \int_h s'_h(\alpha') \underbrace{\prod_{i=1}^{N_c} a_i}_{a_h} \underbrace{\prod_{i=1}^{N_c} p_i(\alpha)}_{p_h(\alpha)} \, dh \\
&= \int_h s'_h(\alpha') a_h p_h(\alpha) \, dh, \tag{4.15}
\end{aligned}$$

where a_h is multiplicative adjustment factor for random walk h and a_i is the adjustment factor for the i th transition of h . This adjustment factor keeps track of the ratio of particle weights between the perturbed and unperturbed systems throughout the biased random walk. At the start of a random walk in the unperturbed system, the adjustment factor is unity. As the random walk progresses, the adjustment factor is multiplicatively updated by the ratio of the probabilities of the phase space transitions between the perturbed and unperturbed systems (these probabilities are given explicitly later on in this section). Recall from Section 2.10 that scoring functions for MC tallies are additive and can be written as

$$s_h(\alpha) = \sum_i^{N_c} s_i(\alpha), \tag{4.16}$$

where $s_i(\alpha)$ is a score for the i th transition. This also applies to the perturbed scoring function, which can be expanded as

$$\begin{aligned}
s'_h(\alpha') &= \sum_i^{N_c} s'_i(\alpha') \\
&= \sum_i^{N_c} (s_i(\alpha) + \Delta s_i) \\
&= s_h(\alpha) + \Delta s_h. \tag{4.17}
\end{aligned}$$

Using Eqs. (4.15) and (4.17), the change in the response function ΔR is expressed as

$$\begin{aligned}
\Delta R &= R'(\alpha') - R(\alpha) \\
&\approx \int_h s'_h(\alpha') a_h p_h(\alpha) dh - \int_h s_h(\alpha) p_h(\alpha) dh \\
&= \int_h (s'_h(\alpha') a_h - s_h(\alpha)) p_h(\alpha) dh \\
&= \int_h \left(\frac{s'_h(\alpha')}{s_h(\alpha)} a_h s_h(\alpha) - s_h(\alpha) \right) p_h(\alpha) dh \\
&= \int_h \left(\frac{s'_h(\alpha')}{s_h(\alpha)} a_h - 1 \right) s_h(\alpha) p_h(\alpha) dh \\
&= \int_h \left[\underbrace{\left(1 + \frac{\Delta s_h}{s_h(\alpha)} \right)}_{\text{direct effect}} \underbrace{a_h}_{\text{indirect effect}} - 1 \right] s_h(\alpha) p_h(\alpha) dh.
\end{aligned} \tag{4.18}$$

Now, aside from the adjustment factor a_h and the perturbed scoring contribution Δs_h , Eq. (4.18) is written only in terms of quantities defined for the unperturbed system. Implementing the correlated sampling technique therefore only requires the tracking of two additional terms throughout the random walks. These two additional terms are used to define the direct and indirect effects (see underbraces from Eq. (4.18)) that a perturbation to α has on $R(\alpha)$.

The direct effect accounts for the direct change that $\Delta\alpha$ has on the scoring function between the perturbed and unperturbed states. The perturbed scoring function is additive and is easily tabulated at every scoring event as the difference in the perturbed and unperturbed scores $\Delta s_h = s'_h - s_h$. For the C5G7-TD3 transients, where the moderator number density, N_{mod} , is modified as a function of time, the macroscopic cross sections are directly affected ($\Sigma_{\text{mod}} = N_{\text{mod}}\sigma_{\text{mod}}$). For a track length tally corresponding to the scattering reaction rate in the moderator, the direct effect would therefore computed as

$$\Delta s_h(\Delta N_{\text{mod}}) = w \Delta N_{\text{mod}} \sigma_{\text{s,mod}} \ell, \tag{4.19}$$

where w is the particle weight, $\Delta N_{\text{mod}} = N_{\text{mod}}(t) - N_{\text{mod}}(0)$, $\sigma_{\text{s,mod}}$ is the microscopic scattering cross section in the moderator, and ℓ is the track length the neutron traveled in the tally volume.

The indirect effect accounts for changes that $\Delta\alpha$ has on the probability of the random walk, and is tracked through the adjustment factor, a_h . To derive the adjustment factor contributions throughout the random walk, consider its definition as the ratio of the perturbed and unperturbed state transition probabilities

$$a_h = \prod_i^{N_c} \frac{p'_i}{p_i}. \tag{4.20}$$

In MC particle transport routines, particles experience state transitions by streaming and colliding with nuclei within the host medium. The probabilities of these individual streaming and collision events define the probability of an individual random walk. First, consider the PDF for a particle streaming event. At the start of a particle history, after the particle state information is populated by a sourcing routine, the particle will be transported a distance ℓ to either an intersection point with a geometric surface ($\ell = \ell_{\text{surf}}$) or a collision site before the surface intersection point ($0 \leq \ell = \ell_{\text{col}} < \ell_{\text{surf}}$). The PDF for a particle streaming a distance ℓ in an infinite medium is given by Eq. (2.151) as

$$P(\ell) = \Sigma_t e^{-\Sigma_t \ell} d\ell. \quad (4.21)$$

If the particle's trajectory crosses a surface before the sampled collision would occur ($\ell_{\text{surf}} < \ell_{\text{col}}$), the particle is transported to the nearest surface intersection instead, with probability

$$P(\ell_{\text{surf}}) = e^{-\Sigma_t \ell_{\text{surf}}}. \quad (4.22)$$

The mixed PDF for the transport kernel is then

$$T_i(\ell) = (\Sigma_t e^{-\Sigma_t \ell} + \delta(\ell - \ell_{\text{surf}}) e^{-\Sigma_t \ell}) d\ell, \quad 0 \leq \ell \leq \ell_{\text{surf}}, \quad (4.23)$$

where $\delta(\ell - \ell_{\text{surf}})$ is the Dirac delta function. If a surface crossing event occurs, then $\ell = \ell_{\text{surf}}$, and the Dirac delta function evaluates to infinity and causes Eq. (4.23) to be dominated as Eq. (4.22). Otherwise, the Dirac delta function evaluates to zero, and Eq. (4.23) evaluates to Eq. (4.21). The ratio of transition probabilities between the perturbed and unperturbed system is therefore given as

$$\frac{T'_i}{T_i} = \frac{(\Sigma'_t e^{-\Sigma'_t \ell} + \delta(\ell - \ell_{\text{surf}}) e^{-\Sigma'_t \ell_{\text{surf}}}) d\ell}{(\Sigma_t e^{-\Sigma_t \ell} + \delta(\ell - \ell_{\text{surf}}) e^{-\Sigma_t \ell_{\text{surf}}}) d\ell}, \quad 0 \leq \ell \leq \ell_{\text{surf}}, \quad (4.24)$$

and can be written as two separate cases: the first being triggered when traveling to a collision, and the second being triggered when traveling to a geometric surface

$$\frac{T'_i}{T_i} = \begin{cases} \frac{\Sigma'_t}{\Sigma_t} e^{-(\Sigma'_t - \Sigma_t)\ell}, & 0 \leq \ell < \ell_{\text{surf}}, \\ e^{-(\Sigma'_t - \Sigma_t)\ell}, & \ell = \ell_{\text{surf}}. \end{cases} \quad (4.25)$$

For the PDF of the collision kernels, first consider that when a collision z occurs, the particle transitions from incident energy E' and direction $\hat{\Omega}'$ to outgoing energy E and direction $\hat{\Omega}$. The probability of this

collision is written as

$$K_i = \frac{\Sigma_z(\mathbf{x}, E)}{\Sigma_t(\mathbf{x}, E)} \mathbb{P}_z(E' \rightarrow E, \hat{\boldsymbol{\Omega}}' \cdot \hat{\boldsymbol{\Omega}}) , \quad (4.26)$$

where $\mathbb{P}_z(E' \rightarrow E, \hat{\boldsymbol{\Omega}}' \cdot \hat{\boldsymbol{\Omega}})$ is the probability of the energy and directional transition occurring. For elastic scattering, this probability is determined by classical mechanics and is a function of the atomic mass of the colliding nucleus. Therefore, assuming that the system perturbation $\Delta\alpha$ does not change the atomic mass of the host medium, $\mathbb{P}_z(E' \rightarrow E, \hat{\boldsymbol{\Omega}}' \cdot \hat{\boldsymbol{\Omega}})$ will be unchanged in the perturbed system. The adjustment factor for the collision kernel is then

$$\frac{K'_i}{K_i} = \begin{cases} \frac{\Sigma'_z/\Sigma'_t}{\Sigma_z/\Sigma_t}, & 0 \leq \ell < \ell_{\text{surf}}, \\ 1, & \ell = \ell_{\text{surf}}. \end{cases} \quad (4.27)$$

Finally, the i th transition probability can therefore be written as the product of the probability of the streaming and collision kernels, T_i and K_i

$$p_i = T_i K_i , \quad (4.28)$$

and Eq. (4.20) can be decomposed as

$$\begin{aligned} a &= \prod_i^{N_c} \frac{T'_i K'_i}{T_i K_i} \\ &= \prod_i^{N_c} \frac{T'_i}{T_i} \frac{K'_i}{K_i} . \end{aligned} \quad (4.29)$$

Figure 4.20 details the modifications to the typical Monte Carlo particle branching history (with implicit capture turned on) in blue text.

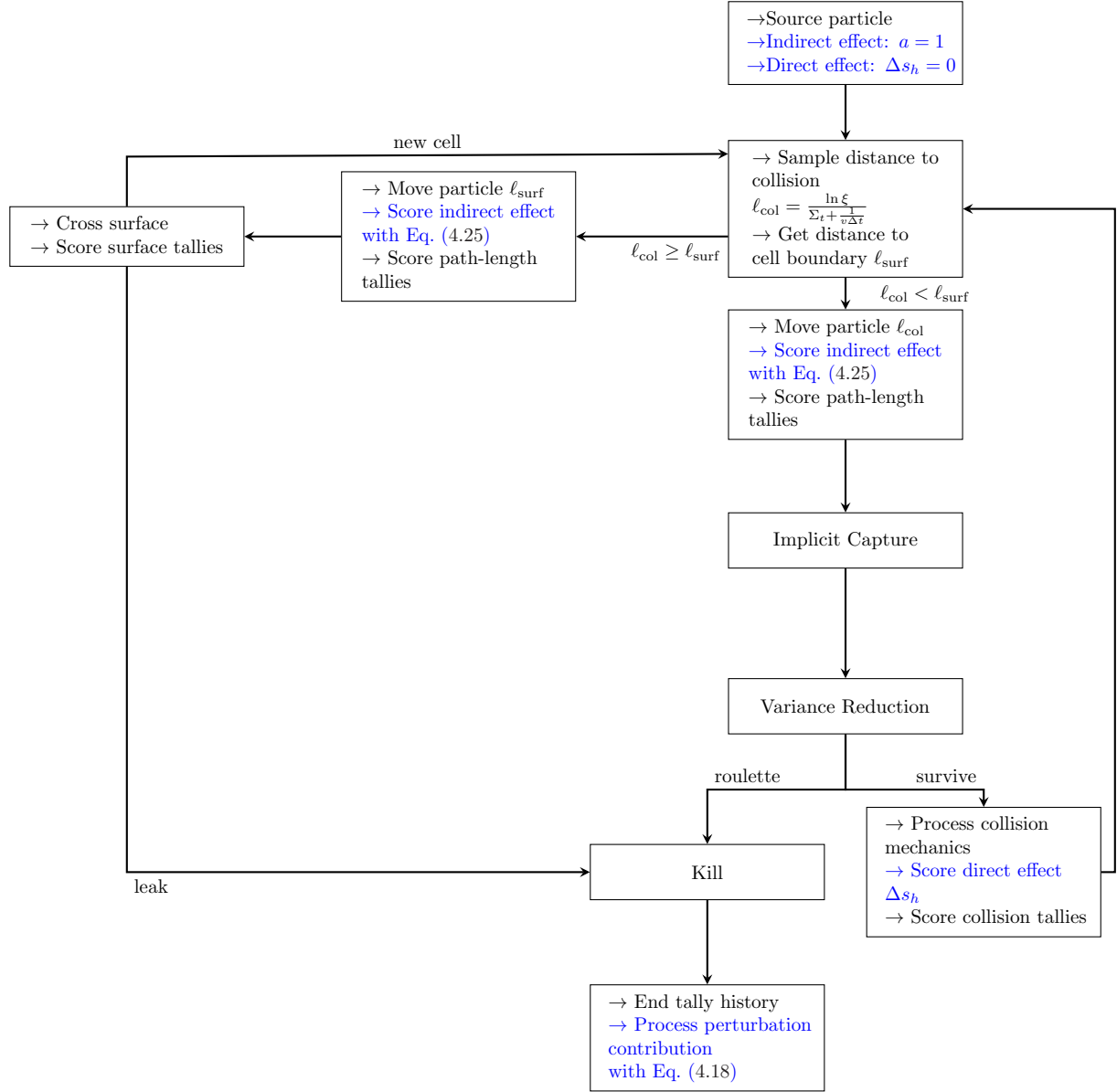


Figure 4.20: Simplified neutron history branching for correlated sampling. Changes from a standard Monte Carlo history are highlighted in blue text.

4.3.2 Perturbed Fission Source Weighting

The transient MC method described in Chapter 3 relies on fission source iteration calculations, where the fission source (the physical distribution of banked neutrons in space and energy) is updated by transferring the particle state information (position, direction, energy, weight) of fission neutrons between cycles. Correlated sampling techniques have traditionally been focused on estimating perturbation effects for fixed source calculations, though there is a growing body of literature for extending perturbation techniques to static (no time dependence) fission source iteration calculations. When the distribution of source particles is changing throughout the calculation, additional adjustment factors (the ratio of particle weights between the perturbed and unperturbed systems) must be tracked across fission generations. For example, if the fission cross section is increased in the perturbed system, more fissions will occur and the additional particle weight from the resulting fission neutrons will need to be tracked across fission generations.

Most of the proposed techniques for applying correlated sampling to fission source iteration calculations are centered around accurately estimating the eigenvalue k_{eff} . While this quantity is certainly of interest for the TML calculations, the main motivation for investigating these correlated sampling techniques is to use these tally estimates as input for the CMFD and EPKE subsolvers of the TML in a parareal scheme, where the correlated sampling TML propagations would serve as the coarse propagator ($\mathcal{G}_{\text{MC-CS}}$). As a result, correlated sampling estimates of general MC tallies will be needed (reaction rates, surface currents, etc.) beyond just tallies for k_{eff} . Additionally, these perturbed source calculations will need to be extended to consider perturbation effects for time-propagation (discussed in Section 4.3.5).

Nakagawa et al. and Kitada et al. both proposed methods for tracking indirect adjustment factors across fission generations [46], [47], but the fluctuations in the propagated adjustment factors increase rapidly with increasing generations. This led to the need of spatially-averaged propagated adjustment factors, where the proper spatial resolution is problem specific and difficult to pinpoint. Nagaya and Mori developed a technique for estimating the perturbed fission source effect by passing a normalized initial weight $q^{*(l)}$ to fission source particles between generations [48]. The method was shown to greatly reduce the fluctuations in the propagated adjustment factors across fission generations and very accurately produces estimates for perturbed values of k_{eff} for both global and local perturbations. Nagaya and Mori's method for estimating changes to k_{eff} basically splits the perturbed estimate into the sum of two quantities

$$\Delta k = \Delta k_{\text{NPS}} + \Delta k_{\text{PS}}, \quad (4.30)$$

where Δk_{NPS} denotes a correlated sampling estimate without the perturbed source effect (i.e., the regular correlated sampling described previously), and Δk_{PS} denotes the contribution exclusively capturing the perturbed fission source effect. For the i 'th collision, the estimate of Δk_{PS} is computed as the product

$$\Delta k_{\text{PS},i} = q^{*(l)} a_i s_i, \quad (4.31)$$

where $s_i = \frac{w \nu \Sigma_f}{\Sigma_t}$ is a score for the k_{eff} collision estimator, and a_i is the adjustment factor introduced in Eq. (4.15). The normalized weights that are passed between fission generations are computed as the sum of unnormalized weights at each fission site. At each fission site f , the unnormalized weight is computed as

$$w_f^{*(l)} = a_i \left(1 + q^{*(l-1)} \right), \quad 1 \leq l \leq L. \quad (4.32)$$

At the end of every fission source iteration, the total unnormalized weight of all fission neutrons produced in generation l , is computed as

$$W^{*(l)} = \frac{\sum_f \nu_f w_f^{*(l)}}{\sum_f \nu_f} = \frac{\sum_f \nu_f w_f^{*(l)}}{\nu^l}, \quad (4.33)$$

where ν_f is the number of fission neutrons produced at the f 'th fission site and ν^l is the number of fission neutrons produced during fission source iteration l . This total unnormalized weight is then used to compute the normalized perturbed fission source weight that will be passed to the next generation

$$q^{*(l)} = \frac{w_f^{*(l)}}{W^{*(l)}} - 1. \quad (4.34)$$

4.3.3 Extension of Perturbed Fission Source Weighting to Other Tally Ratios

To the author's knowledge, Nagaya and Mori's method has only been applied to compute Δk_{eff} estimates and not to other MC tallies. However, Kiedrowski et al. showed that Nagaya and Mori's perturbed fission source weighting method for differential operator sampling can be extended to other MC tallies. The premise of the remainder of this chapter is based on the idea that correlated sampling technique can similarly be generalized to additional MC tallies and used to project perturbations forward in time to generate input parameters for TML subsolvers. Since are only unique up to a scalar multiple, only ratios of MC flux-dependent tallies can be used as input to subsolvers across different fission source iteration solutions. By definition, k_{eff} is the ratio of two phase space-integrated quantities. Fortunately, all of the parameters input to the TML subsolvers can also be defined as or derived from ratios of MC tallies. For example, recall the macroscopic cross sections

Algorithm 4.2 Routine for computing initial perturbed source weight for every fission source iteration.

```

1:  $q^{*(0)} \leftarrow 0$ 
2: for all  $1 \leq l \leq L$  do  $\triangleright L$  is total # of fission source iterations
3:    $f = 0$ 
4:    $\nu_h^l = 0$ 
5:   for all  $1 \leq h \leq \nu_h^{l-1}$  do
6:      $\Delta s_h \leftarrow 0$ 
7:      $q^{*(l)} \leftarrow \text{Eq. (??)}$   $\triangleright$  Perturbed source adjustment factor is set from previous fission source iteration
8:     for all  $1 \leq i \leq N_c$  do
9:        $\Delta k_{\text{PS},i} \leftarrow \text{Eq. (4.31)}$ 
10:      if fission event sampled then  $\triangleright$  Each fission neutron gets assigned bank index  $f$ 
11:         $w_f^{*(l)} \leftarrow \text{Eq. (4.32)}$ 
12:         $\nu_h^l += \nu_f$ 
13:         $W^{*(l)} += \nu_f w_f^{*(l)}$ 
14:         $f += 1$ 
15:      end if
16:    end for
17:     $\Delta k \leftarrow \text{Eq. (4.30)}$ 
18:  end for
19:   $W^{*(l)} / = \nu^l$ 
20: end for
21:  $q^{*(l)} \leftarrow \text{Eq. (4.38)}$ 

```

used as input for the transient CMFD solver from Section 3.2.2.1

$$\overline{\overline{\Sigma}}_{t,l,m,n}^g \equiv \frac{\left\langle \overline{\overline{\Sigma}}_{t,l,m,n}^g \overline{\phi}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle}{\left\langle \overline{\phi}_{l,m,n}^g \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle}, \quad (4.35a)$$

$$\overline{\overline{\Sigma}}_{s,l,m,n}^{h \rightarrow g} \equiv \frac{\left\langle \overline{\overline{\Sigma}}_{s,l,m,n}^{h \rightarrow g} \overline{\phi}_{l,m,n}^h \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle}{\left\langle \overline{\phi}_{l,m,n}^h \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle}, \quad (4.35b)$$

$$\overline{\overline{\nu}}_{f,l,m,n}^{h \rightarrow g} \equiv \frac{\left\langle \overline{\overline{\nu}}_{f,l,m,n}^{h \rightarrow g} \overline{\phi}_{l,m,n}^h \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle}{\left\langle \overline{\phi}_{l,m,n}^h \Delta_l^u \Delta_m^v \Delta_n^w \right\rangle}. \quad (4.35c)$$

The total macroscopic cross section energy group g and voxel (l, m, n) given by (4.35a) is the ratio of the total reaction rate for that subset of phase space to the total collision density (or track length generated) for that subset of phase space. Both of these quantities are straightforward to calculate with Monte Carlo solvers by either collision or path-length tallies. The same idea applies to computing the group-to-group scattering rate defined in Eq. (4.35b) and the fission reaction rate defined in Eq. (4.35c). Further, the special CMFD diffusion coefficients $\tilde{D}$ and $\hat{D}$ from Eqs. (2.142) and (2.144) can be defined as the ratio of either volume-integrated transport-corrected diffusion coefficients, or the ratio of surface-integrated leakage tallies to volume-averaged scalar flux tallies. Finally, the point kinetics parameters from Eq. (2.54) can be computed

as the ratio of two volume-integrated MC tallies.

In order to extend Nagaya and Mori's method to other tally ratios, a simple generalization is proposed: change the scoring contribution s_i in Eq. (4.31) from that of a k_{eff} score ($s_i = \frac{w\nu\Sigma_f}{\Sigma_t}$) with the appropriate score for the response of interest. For example, for a simple flux track length tally, $s_i = w\ell$. Most of the infrastructure for implementing Nagaya and Mori's method is concentrated to six transport routine injection points:

1. *sourcing a fission particle*: The initial perturbed source adjustment factor is assigned, given its progenitor index.
2. *sampling fission sites*: Unnormalized weight contributions are computed with Eq. (4.32)
3. *particle streaming events*: Indirect effect adjustment factor contributions are computed with Eq. (4.25).
4. *particle collision events*: Indirect effect adjustment factor contributions are computed with Eq. (4.27).
5. *end of particle history*: Perturbed and non-perturbed source contributions to the correlated sampling perturbation tally are computed with Eqs. (4.31) and (4.18), respectively.
6. *end of fission source iteration cycle*: The perturbed source adjustment factor is renormalized with Eq. (4.38).

Figure 4.21 depicts the changes to a Monte Carlo neutron history that are required to implement the technique into a standard Monte Carlo fission source iteration routine.

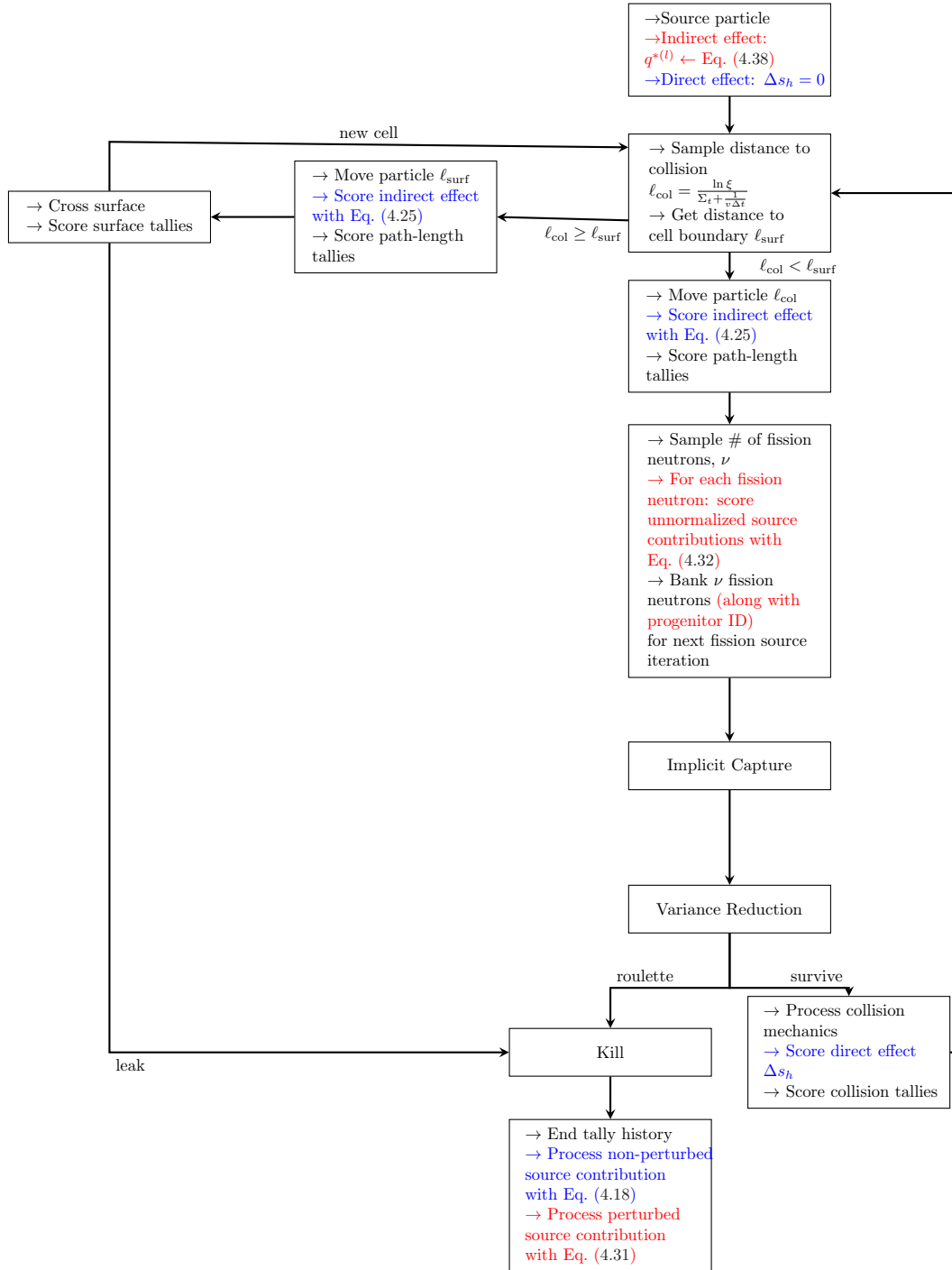


Figure 4.21: Simplified neutron history branching for correlated sampling. Changes for the non-perturbed fission source effect from a standard Monte Carlo history are highlighted in blue text. Changes for the perturbed fission source effect from a standard Monte Carlo history are highlighted in red text.

4.3.4 Godiva Perturbations

Nagaya and Mori present a few test cases for validating that their method works for perturbation estimates of Δk_{eff} . The first test cases evaluate the method for density perturbations of the canonical Godiva critical configuration [110], a bare uranium metal sphere with a radius of 8.741 cm. The physical experiment had a density of 18.74 g cm^{-3} and composition 93.71 wt% ^{235}U , 5.24 wt% ^{238}U , and 1.05 wt% ^{234}U . For the simulated perturbed states, densities of the Godiva material were increased to 20.0, 21.0, 23.5, and 26.0 g cm^{-3} . First, the series of density perturbations we applied uniformly over the entire Godiva sphere, seen in Fig. 4.22 (left). Second, to test the accuracy of the method for strong local perturbations, the same density perturbations were applied only to an inner 6 cm radius of the Godiva configuration, seen in Fig. 4.22 (right).

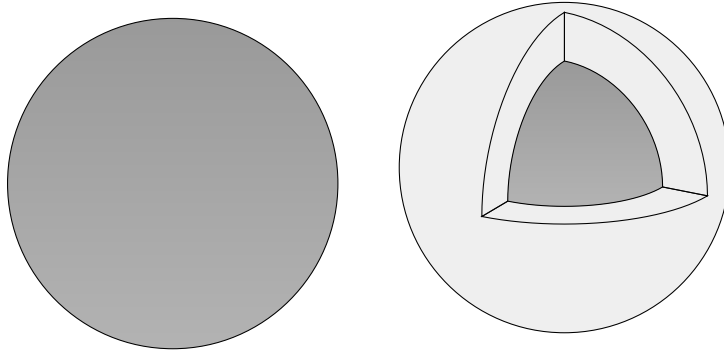


Figure 4.22: Godiva uniform (left) and central (right) perturbations where the dark gray regions correspond to the perturbed material density.

The perturbed fission source correlated sampling technique was first implemented in an in-house continuous energy MC code. Infrastructure for the Nagaya and Mori's perturbed fission source technique for differential operator sampling already existed in this code and was leveraged to ease the analogous correlated sampling implementation. Figure 4.23 shows the correlated sampling with perturbed fission source results for the uniform Godiva density perturbations. As a verification that the implementation is working as intended, the relative change in k_{eff} is shown in the top left plot. The results agree well with those reported by Nagaya and Mori.

The remaining plots from Fig. 4.23 show the same method applied to various combinations of volume- and surface-integrated MC tally ratios: the volume-integrated scattering reaction rate to volume-integrated flux (top right), the surface-integrated leakage for an internal spherical surface with a radius of 6 cm to volume-integrated scattering reaction rate (bottom right), and the surface-integrated leakage for the exterior Godiva surface to volume-integrated scattering reaction rate (bottom left). Direct MC calculations at the

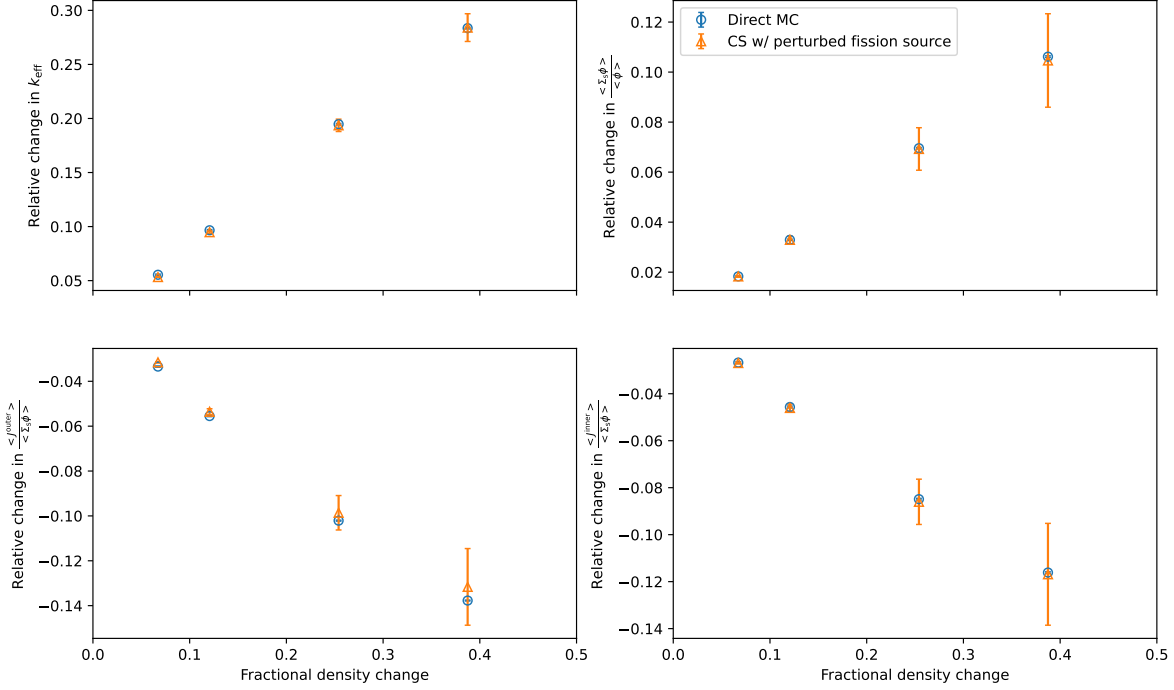


Figure 4.23: Correlated sampling results with perturbed fission source for Godiva configuration with various density perturbations applied uniformly to the entire Godiva geometry. The top plots describe the change in k_{eff} (left) and the change in the scattering reaction rate for the entire geometry relative the scalar flux (right). The bottom plots describe the change in the outer surface current out of the Godiva geometry relative to the scattering reaction rate for the entire geometry (left) and the change in the surface current out of an internal spherical surface with a 6 cm radius relative to the scattering reaction rate for the entire geometry (right).

perturbed and unperturbed states are used as reference calculations. Therefore, to generate the data in Fig. 4.23, five independent MC calculations were run: one for the initial unperturbed state with the a material density of 18.74 g cm^{-3} , and one for each of the four density perturbations prescribed by Nagaya and Mori.

For both the direct MC calculations and the correlated sampling MC calculation, 1×10^4 histories per fission source iteration cycle were run with 50 inactive cycles and 100 active cycles. Note that all of the results shown in Fig. 4.23 (aside from the direct reference calculations), were generated from a single MC eigenvalue calculation. All of the presented tally ratios perturbations are well-approximated by the perturbed fission source correlated sampling technique for the uniform density perturbation.

Figure 4.24 shows the same reaction ratio combinations and density perturbations simulated for the central Godiva perturbation problem. This time, only estimates for k_{eff} agree well with the direct MC reference calculations, with less than 6% difference between the mean estimates from the two methods. The ratio of the surface-integrated leakage at the material interface to the volume-integrated scattering reaction rate does fall within the standard deviation error bars, but the error bars are quite forgiving for this quantity ($\sim 50\%$ relative uncertainty for all of the density perturbations). It would appear from the results presented here

then, that the perturbed fission source correlated sampling technique is not a good approximation for strong localized perturbations.

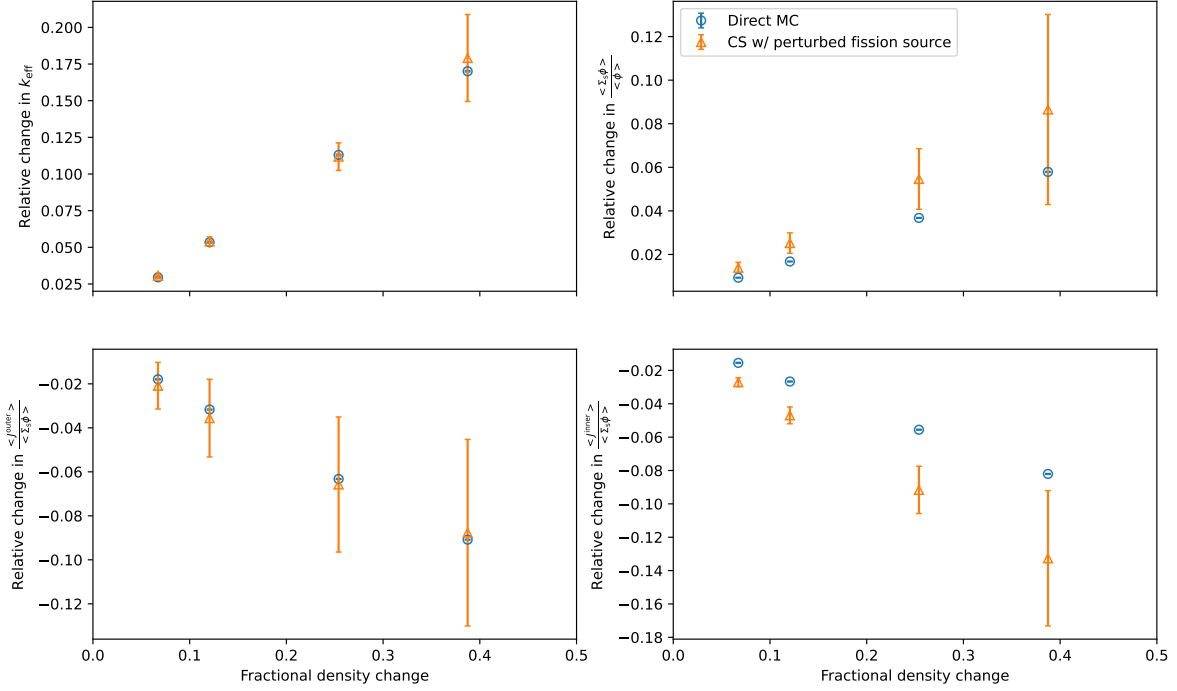


Figure 4.24: Correlated sampling results with perturbed fission source for Godiva configuration with various density perturbations to the 6 cm inner Godiva core. The top plots describe the change in k_{eff} (left) and the change in the scattering reaction rate for the entire geometry relative to the scalar flux (right). The bottom plots describe the change in the outer surface current out of the Godiva geometry relative to the scattering reaction rate for the entire geometry (left) and the change in the surface current out of the 6 cm interface between the two Godiva regions relative to the scattering reaction rate for the entire geometry (right).

For the C5G7-TD3 transients presented in the previous chapters, time-dependent density perturbations are applied to the neutron moderator within the reactor core. Before migrating the correlated sampling technique to the Shift MC code, a homogenized two-region UO_2 pincell from the C5G7 geometry specifications [92] was modeled with the in-house MC code using reflecting boundary conditions. The correlated sampling results are shown in Fig. 4.25. This time, k_{eff} was analyzed in the top left plot, the ratio of the surface-integrated leakage at the fuel-moderator interface to the volume-integrated scattering reaction rate in the fuel was analyzed in the bottom plot, and the ratio of the volume-integrated scattering reaction rate to the volume-integrated scalar flux was analyzed in the top right plot.

All reaction ratio perturbation estimates agree with the direct MC reference solutions. For k_{eff} , the three smallest density perturbations have relative errors of less than 5% between the mean estimates of the correlated sampling and direct MC calculations, and 11% for the largest density perturbation. For the scattering reaction rate relative to the scalar flux plot (top right), the means between the direct MC

and correlated sampling calculations agree within 1%. Finally, for the outer surface current relative to the scattering reaction rate plot, the means between the direct and correlated sampling calculations agree within 2%.

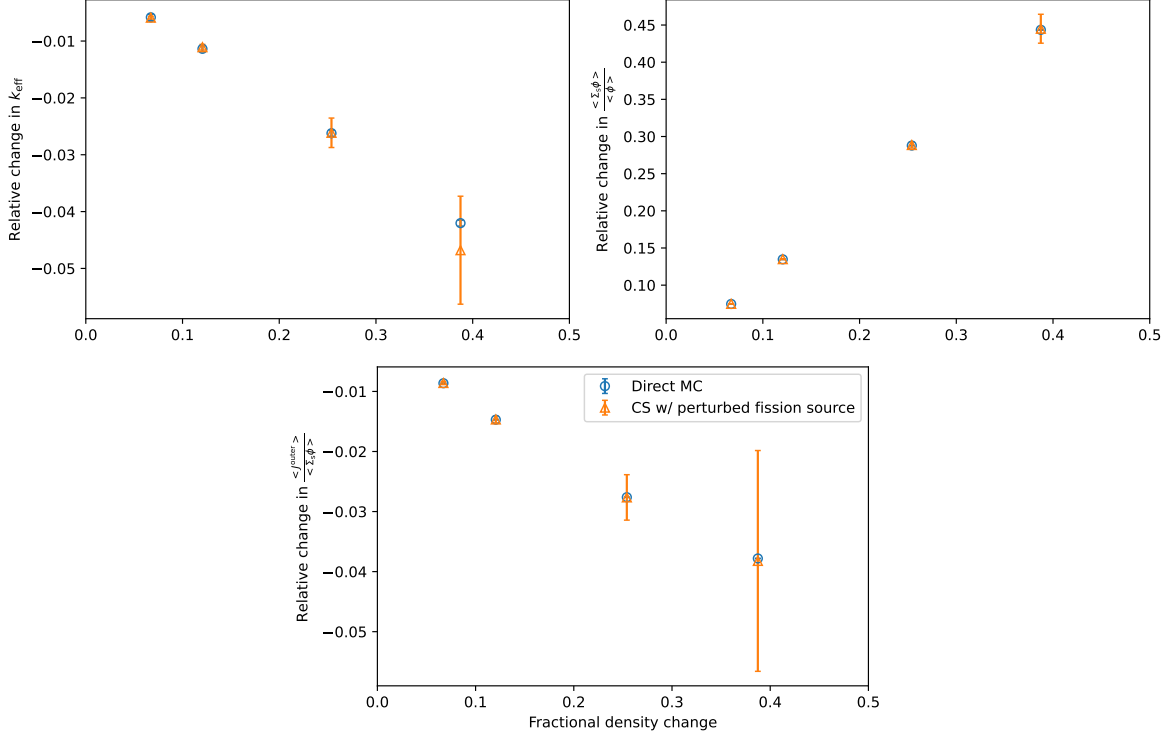


Figure 4.25: Correlated sampling results with perturbed fission source for pin cell configuration with various moderator density perturbations. The top left plot describes the change in k_{eff} , the top right plot describes the change in the scattering reaction rate in the fuel relative the scalar flux in the fuel, and the bottom plot describes the change in the surface current out of the fuel relative to the scattering reaction rate in the fuel.

4.3.5 Potential Use in a Multi-Level Scheme

The fission source perturbation correlated sampling technique discussed above was implemented in Shift to estimate changes in the static k -eigenvalue, Δk_{eff} . Extension of the technique has not yet been implemented for ratios of other MC tallies but should not be difficult to accomplish now that most of the required infrastructure exists and no further modifications to the transport routine should need to be made. All tracking of the direct effect, indirect effect, and indirect fission source renormalization are implemented in a standalone correlated sampling data structure. Most of the remaining work to extend the implementation will be concentrated in communicating information from Shift's internal tallying data structures to this new correlated sampling data structure. In its current state, the correlated sampling data structure only supports cell-based tallies. Additional work will need to be done to extend the data structure to support mesh tallies.

Figure 4.26 shows the result of using the static Δk_{eff} estimate as a surrogate for the time-dependent dynamic reactivity to advance the C5G7-TD3-4 transient in a Predictor-Corrector Quasi-static Method (PCQM) coupling. While computing the direct MC solution with time steps of Δt_{MC} and $\Delta t_{\text{EPKE}} = 4$ ms, density perturbations are projected forward in time at an intermediate time step of $\Delta t_{\text{CS}} = 0.1$ s. Only three direct MC evaluations were used to solve this transient for a total simulation time of 29.37 CPU hrs, as opposed to the 351.1 CPU hrs needed to propagate the TML solution shown in black. The correlated sampling estimates for the dynamic reactivity are not expected to agree exactly with the direct TML propagation for several reasons, including that:

1. Dynamic reactivity is supposed to be computed as an adjoint-weighted quantity. Here, it is substituted with the static definition of reactivity and without importance weighting.
2. The remaining point kinetics parameters (β^{eff} , ζ_j , Λ , etc.) are not being updated at the Δt_{CS} time scale, but rather at the direct MC time scale.
3. The uncertainties in the correlated sampling technique are resolved with increasing fission source iteration cycles. Only 250 active cycles were used to produce Fig. 4.26.
4. There is no special consideration given for the neutrons propagated between time steps.

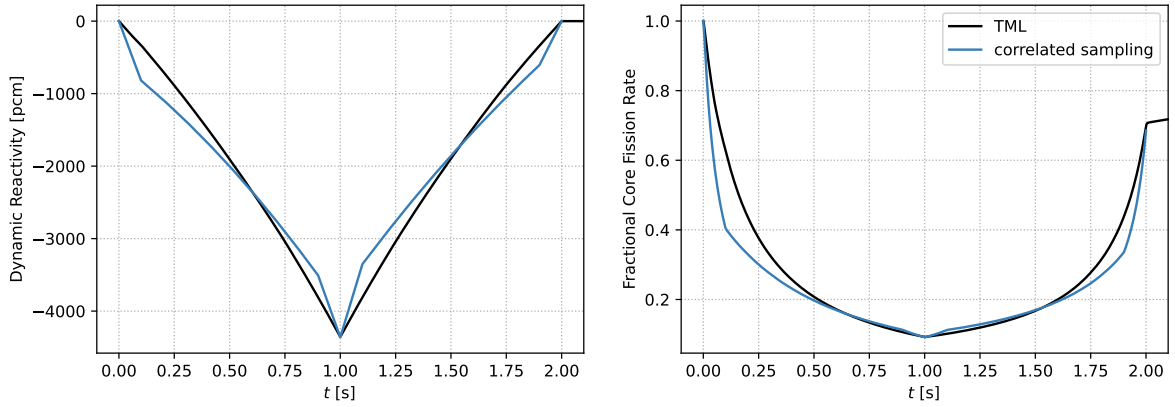


Figure 4.26: Correlated sampling results for C5G7-TD3-4 transient.

Addressing the fourth bullet point to add time consideration to the density perturbations will require special modifications to the correlated sampling technique. To the author's knowledge, time-dependent MC neutron transport with the correlated sampling technique has not been studied and should be treated as an avenue of future work.

4.3.5.1 Direct Effect

The only conceptual difference for the direct effect is that parameter perturbations are now a function of time. With regards to prescribed transients, no further modifications to the Shift implementation should be necessary beyond those already implemented for estimating Δk_{eff} with Nagaya and Mori’s method given a series of material density perturbations. For example, with the C5G7-TD3 transient, the time-dependent parameter perturbations correspond to the fractional moderator density profile prescribed for the transient, $\Delta\alpha(t) = \frac{\rho_{\text{mod}}(t)}{\rho_{\text{mod}}(0)}$. For transients with feedback, new parameter perturbations will need to be evaluated in between direct MC time steps as the transient progresses.

4.3.5.2 Indirect Effect on the Perturbed Random Walk

For time consideration, the indirect effect has to take into account that the transport and collision have clearly changed because the total cross section has been augmented with an additional “time absorption” reaction. Mathematically, this is simply adding an additional $\frac{1}{v\Delta t}$ factor to the total cross section during transport and tallying. The transport probability T_i then becomes

$$T_i(\ell) = \left[\left(\Sigma_t + \frac{1}{v\Delta t} \right) e^{-(\Sigma_t + \frac{1}{v\Delta t})\ell} + \delta(\ell - \ell_{\text{surf}}) e^{-(\Sigma_t + \frac{1}{v\Delta t})\ell} \right] d\ell, \quad 0 \leq \ell \leq \ell_{\text{surf}}. \quad (4.36)$$

Similarly, the collision probability K_i becomes

$$K_i = \frac{\Sigma_z(\mathbf{x}, E, t)}{\Sigma_t(\mathbf{x}, E, t) + \frac{1}{v\Delta t^n}} \mathbb{P}_z \left(E' \rightarrow E, \hat{\Omega}' \cdot \hat{\Omega} \right). \quad (4.37)$$

Both of these kernels are already implemented (without the $\frac{1}{v\Delta t}$ terms) for both the fixed source and fission source iteration schemes depicted in Figs. 4.20 and 4.21.

4.3.5.3 Indirect Effect on the Perturbed Fission Source

In a time-propagated correlated sampling scheme, the unperturbed fission source distribution at time index n and fission source iteration index l is used to predict the perturbed fission source distribution at time index $n + 1$ and fission source iteration index l . The weight contribution at the f th fission site of the l th fission source iteration is still computed analogously to Eq. (4.32) as

$$q^{*(l,n+1)} = \frac{w_f^{*(l,n+1)}}{W^{*(l,n+1)}} - 1. \quad (4.38)$$

However, the fission source weight renormalization factor $W^{*(l,n+1)}$ now has to account for each of the source terms present in Eq. (3.11), reproduced here with the time indices advanced by one ($n-1 \rightarrow n$ and $n \rightarrow n+1$)

$$\begin{aligned}
\left(M + \frac{1}{v(E)\Delta t^{n+1}}\right) \psi^{l+1,n+1}(\mathbf{x}, \hat{\Omega}, E) &= \underbrace{\frac{\psi^n(\mathbf{x}, \hat{\Omega}, E)}{v(E)\Delta t^{n+1}}}_{\textcircled{1}} + \underbrace{\mathbf{F}_p \psi^{l,n+1}(\mathbf{x}, \hat{\Omega}, E)}_{\textcircled{2}} \\
&+ \underbrace{\sum_{j=1}^J \mathbf{S}_{d,j} \psi^n(\mathbf{x}, \hat{\Omega}, E) f_{1,j}}_{\textcircled{3}} \\
&+ \underbrace{\sum_{j=1}^J \mathbf{F}_{d,j} \psi^n(\mathbf{x}, \hat{\Omega}, E) f_{2,j}}_{\textcircled{4}} + \underbrace{\sum_{j=1}^J \mathbf{F}_{d,j} \psi^{l,n+1}(\mathbf{x}, \hat{\Omega}, E) f_{3,j}}_{\textcircled{5}}.
\end{aligned} \tag{4.39}$$

These additional source terms include contributions from prompt fission (source term $\textcircled{2}$), delayed fission (source terms $\textcircled{4}$, $\textcircled{5}$), and census-captured neutrons (source term $\textcircled{1}$). Since the transient MC implementation in Shift uses implicit capture for both fission and census capture events, at every collision there are ν fission neutrons and one census neutron produced. The perturbed fission source weight renormalization factor is then

$$W^{*(l,n+1)} = \frac{\sum_f w_f^{*(l,n+1)} \left\{ \nu_f^n \left[p_{\textcircled{2}}^n + p_{\textcircled{4}}^n + p_{\textcircled{5}}^n \right] + p_{\textcircled{1}}^n \right\}}{\sum_f \left\{ \nu_f^n \left[p_{\textcircled{2}}^n + p_{\textcircled{4}}^n + p_{\textcircled{5}}^n \right] + p_{\textcircled{1}}^n \right\}}, \tag{4.40}$$

where, $p_{\textcircled{1}}^n$ is the probability that a neutron is census captured, $p_{\textcircled{2}}^n$ is the probability that a prompt fission neutron is born, $p_{\textcircled{4}}^n$ is the probability that a delayed neutron is born and numerically weighted to be sourced at t^{n+1} , and $p_{\textcircled{5}}^n$ is the probability that a delayed neutron is born and numerically weighted to be sourced at t^n . Each of these probabilities is defined below as

$$p_{\textcircled{1}}^n = \frac{1}{v(E)\Delta t^n} \frac{1}{\Sigma_t^n(\mathbf{x}, E) + \frac{1}{v(E)\Delta t^n}}, \tag{4.41a}$$

$$p_{\textcircled{2}}^n = (1 - \beta), \tag{4.41b}$$

$$p_{\textcircled{4}}^n = \sum_{j=1}^J \beta_j f_{2,j}^n, \tag{4.41c}$$

$$p_{\textcircled{5}}^n = \sum_{j=1}^J \beta_j f_{3,j}^n. \tag{4.41d}$$

For time indices $n > 0$, additional neutron histories are sourced from the time step source bank that is passed from t^{n-1} . The normalized fission source weights for these time-source neutrons need to be known to properly renormalize the fission source weights in subsequent fission source iterations. Since the fission source distribution from the end of the active fission source iteration cycles at t^n is used as the initial fission source distribution guess for $l = 0$ at t^{n+1} , the vector of initial perturbed source weights can be passed between time steps $\mathbf{q}^{*(0,n)} = \mathbf{q}^{*(L,n-1)}$. The length of each of the $\mathbf{q}$ vectors corresponds to the number of fission sites sampled at the previous fission source iteration index $l - 1$. Figure 4.27 depicts the transfer of normalized fission source weights between fission source and time indices.

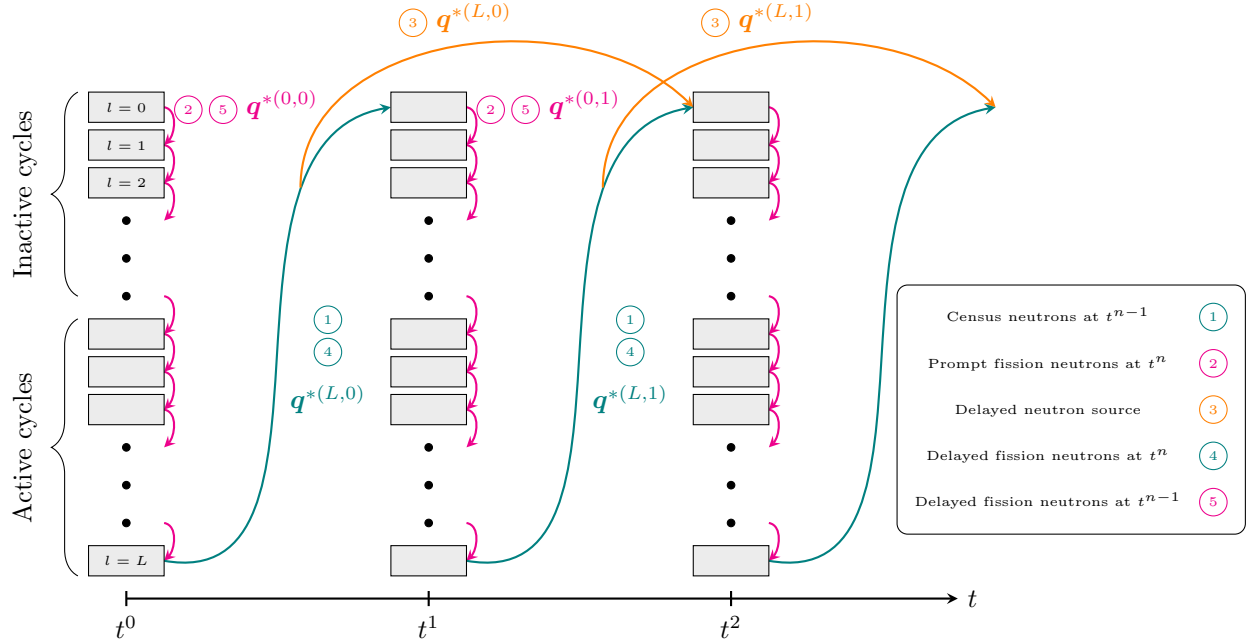


Figure 4.27: Transfer of source terms and perturbed fission source weights between fission source iterations and time steps for a time-propagated correlated sampling scheme.

For the time-propagated perturbed fission source effect, no modifications to the Shift transport routine should be necessary beyond those already completed to implement Nagaya and Mori's correlated sampling scheme for estimating static perturbations to k_{eff} . The remaining modifications will largely be concentrated within the correlated sampling tally data structure for computing the source weight renormalization factor described by Eq. (4.40). Additionally, the particle state information for neutrons transferred between time steps will need to have a member variable added to denote the time and fission source iteration indices where the particle was born.

4.4 Summary

In this chapter, the parareal algorithm was implemented in serial and investigated as an accelerating mechanism for quasi-static reactor kinetics calculations. The serial implementation was first tested as a coupling between two EPKE solvers, where the $\mathcal{G}$ and $\mathcal{F}$ propagators were differentiated by rediscrctized temporal domains. A slow asymmetric reactivity ramp with an analytic solution, and a fast control-rod ejection with precomputed reactivity and power profiles from the PARCS solver, were used to verify the implementation converges to the correct solution within four parareal iterations. The serial algorithm was then migrated to the Shift TML implementation and tested with a coupling between the CMFD and EPKE levels for the $\mathcal{G}_{\text{CMFD}}$ and $\mathcal{F}_{\text{EPKE}}$ solvers, respectively. While the solution did converge after five parareal iterations, this particular coupling is not ideal as the order of the $\mathcal{G}_{\text{CMFD}}$ and $\mathcal{F}_{\text{EPKE}}$ is reversed in terms of computational cost and would not provide significant savings when implemented in parallel. In order to test the correct hierarchy of $\mathcal{G}$ and $\mathcal{F}$, an additional CMFD level with one energy group and a rediscrctized temporal domain. This implementation converged after a single parareal iteration for both $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 5$ and 10. The projected speedups are promising and range from 1.5 to 8, depending on the ratio of $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}}$ and whether parallelization across the energy group is used for the CMFD solver.

While the parareal coupling results are promising, the timing results from Chapter 3 showed that meaningful reductions in TML simulation time would need to be focused on minimizing the number of MC evaluations needed to solve the transient problem. The latter half of this chapter investigated correlated sampling as a way to obtain estimates for MC tallies used as input to the TML subsolvers throughout the transient. In this way, a coarse estimate of the predictor tallies can be propagated forward in time as a $\mathcal{G}$ propagator and direct MC calculations can correct the solution as a $\mathcal{F}$ propagator. It was shown with density perturbations to the Godiva geometry that Nagaya and Mori’s method for estimating reactivity perturbations with correlated sampling can be generalized to both volume-integrated and surface-integrated tallies for spatially-uniform perturbations. The same was not observed for strong locally-varying perturbations. Fortunately, perturbations to the moderator density for a pincell calculation matched very well to the direct MC reference calculations. To conclude the chapter, Nagaya and Mori’s correlating sampling technique was applied in the Shift MC code to estimate k_{eff} perturbations. Speculations are given on what modifications to the algorithm would be necessary to include indirect effects from propagating the perturbation forward in time. The next chapter of this dissertation discusses summarizes all of the results presented and postulates what future avenues of research might look like.

Chapter 5

Conclusions and Future Work

The primary objective of this work was to enable high-fidelity time-dependent neutron transport calculations, specifically Monte Carlo (MC) transport calculations, for practical use in nuclear reactors or for used nuclear fuel transient criticality analyses. To accomplish this, hybrid multi-level calculations were implemented in the Shift MC solver to make time-dependent MC calculations more computationally tractable. Additionally, parallel-in-time integration methods were explored (in serial) as potential mechanisms to accelerate the convergence of lower-level quasi-static solvers. Finally, a correlated sampling technique with fission source weight normalization for estimating static reactivity perturbations was extended to general MC tallies. The technique is proposed as a potential generator of time-propagated MC tallies so that they might be used as input for the lower-level quasi-static solvers in future parallel-in-time integration schemes. The remainder of this chapter will summarize each of these endeavors in more detail and propose future avenues of work.

5.1 Monte Carlo Transient Multi-Level Method

In Chapter 3, the procedure for extending a static k -eigenvalue solver to support transient calculations was discussed. The implementation procedure was simplified from those of previous works by adding a single source bank (as opposed to four distinct source banks) to transfer census-captured and delayed neutrons in between MC time steps. The Transient Multi-Level (TML) method, a nested predictor-corrector quasi-state approach that solves the Coarse Mesh Finite Difference (CMFD) equations at the intermediate time step and the Exact Point Kinetics Equations (EPKEs) at the fine time step, was then implemented in a MC solver for the first time. The MC TML was verified against a deterministic implementation of the TML for the C5G7-TD3 transient benchmark, where the two codes were in agreement.

The TML showed improvements over the standard Predictor-Corrector Quasi-static Method (PCQM) with reductions in both the number of MC evaluations needed to solve the transient and the prevalence of stochastic noise in the power profile. Overall, the TML method showed roughly an order of magnitude

improvement over the PCQM for both the total CPU hours and maximum absolute error in the power profile for the transients examined. Zhu observed similar performance improvements with their Method of Characteristics (MOC) TML implementation [13]. It was shown, for the transient scenarios studied, that improvements in the L_2 convergence of the fractional core fission rate profile diminished for ratios of $\Delta t_{MC}/\Delta t_{CMFD} > 10$. In this practical regime of MC to CMFD time step selection, the MC solver was responsible for $\geq 95\%$ of the total computational cost. While expected, this result confirms that meaningful TML acceleration schemes should reduce the number of MC evaluations needed to solve a transient problem.

The MC TML implementation was then extended to support psuedo multigroup Doppler broadening with a functional dependence on temperature to test the method’s applicability to rapid transients dominated by prompt feedback. The feedback-dominated transients were tested for the small C5G7 geometry and a large Dual-Purpose Canister (DPC) configuration with significant asymmetries in both the radial and axial directions. The method was able to simulate the step reactivity insertions with millisecond resolution at the MC level in both cases but significant work was required on the user-end for the DPC geometry to determine the appropriate spatial resolution necessary for the CMFD calculations. On the one hand, the spatial resolution has to be sufficiently refined to accurately represent the spatial dependence of the flux. On the other hand though, small CMFD mesh elements correspond to difficult-to-resolve tally statistics from the MC solver. The MC tally statistics at the bottom of the DPC geometry were particularly troublesome, as very few analog interactions take place where the fission distribution is low. To date, most of the CMFD applications for MC fission source acceleration have been studied for either 1D geometries, 2D geometries or small 3D geometries with radial symmetry and low dominance ratios. There remains a gap in the literature for MC CMFD applications specifically developed for large geometries with large variations in the spatial fission distribution. The robustness of the MC TML algorithm remains undetermined for this reason.

5.1.1 Future Work

The power pulse transient simulated for the DPC geometry revealed that the CMFD solver faces spatial refinement difficulties when incorporated with a MC solver. Nelson developed hybrid methods for reducing the uncertainty of MC-generated scattering matrices [103]. Future avenues of work might consider leveraging these methods to remedy this issue or implementing variance reduction techniques to bias particles towards regions with low analog interactions. Future MC TML implementations might also consider adaptive CMFD mesh refinement across MC time steps for transients with significant changes to the spatial fission distribution. Alternatively, future work might consider replacing the CMFD solver with a more MC-friendly solver altogether that is not heavily reliant on analog tallies.

For higher-fidelity analyses, the TML implementation in Shift should be extended to support continuous-energy physics. Most of the infrastructure to support this feature already exists in Shift. The only significant barrier in enabling these continuous-energy transient calculations is the pre-processing and internal memory storage of the delayed neutron precursor fission spectra. At the time of implementation and writing, only cumulative delayed neutron data was available in Shift and not data specified for individual delayed neutron precursor groups. Continuous energy physics would also enable the use of Shift’s on-the-fly Doppler sampling [100] so that Doppler feedback coefficients can be calculated or fuel temperature feedback can be more realistically simulated. This would additionally open the door for multi-physics calculations to be accomplished with Shift as the neutronics solver. Shen has already extended the TML implementation in the Michigan Parallel Characteristics Transport (MPACT) solver to stabilize multi-physics calculations and similar techniques could potentially be translated to the MC implementation [61].

An additional improvement to the Shift implementation include enabling adjoint-weighted tallies [111]. This would allow for more-accurate point kinetics parameters (when computed with continuous-energy physics) and would eliminate the need for the external SP_3 adjoint eigenvalue solution. Adjoint-weighted tallies would also help enable more accurate time-propagated correlated sampling estimates to propagate the EPKE amplitude function forward in future parallel-in-time schemes (discussed at the end of Chapter 4).

5.2 Parallel-in-Time Acceleration of the TML Subsolvers

In Chapter 4, the parareal algorithm and its applicability to multi-fidelity neutron kinetics problems was investigated. A serial implementation of the algorithm was first applied as a coupling between two exact point kinetics solvers, where the difference in fidelity between $\mathcal{G}$ and $\mathcal{F}$ came from a rediscritized temporal domain. For this coupling, two prescribed-transient verification problems were tested: a slow asymmetric reactivity ramp with one delayed neutron precursor group (for which an analytical solution was derived), and a fast control-rod ejection with precomputed reactivity and power profiles from the Purdue Advanced Reactor Core Simulator (PARCS) solver. Both of these test problems converged towards the reference solution within four parareal iterations. The serial algorithm was then implemented with the Shift TML as a coupling between the CMFD and EPKE levels for the $\mathcal{G}$ and $\mathcal{F}$ solvers, respectively. While the solution did converge after five parareal iterations, this particular coupling is not ideal as the order of the $\mathcal{G}$ and $\mathcal{F}$ is reversed in terms of computational cost and would not provide significant savings when implemented in parallel. In order to test the correct hierarchy of $\mathcal{G}$ and $\mathcal{F}$, an additional CMFD level with one energy group and a rediscritized temporal domain was investigated. This implementation converged after a single parareal iteration for both $\Delta t_{\text{CMFD-1G}}/\Delta t_{\text{CMFD-7G}} = 5$ and 10. For these time step ratios, projected speedups with a

parallel implementation (ignoring potential computational overhead) ranged from 1.5 to 8, depending on the number of available processing units and whether or not parallelization across the energy group is used when solving the CMFD-7G matrices.

5.2.1 Correlated Sampling as a potential Coarse-Estimate MC Solver

The latter half of Chapter 4 investigated correlated sampling as a way to obtain estimates for MC tallies propagated through time. These tally estimates could then be used as input for the TML subsolvers to propagate the solution forward in time. These correlated sampling propagation could then potentially be used as a $\mathcal{G}$ propagator and direct MC calculations can correct the solution as a $\mathcal{F}$ propagator. One specific correlated sampling technique, developed by Nagaya and Mori, was expanded beyond its intended exclusive use as an estimate for static reactivity perturbations to general static MC tally perturbations. It was confirmed using density perturbations to the Godiva geometry and the moderator in a pincell geometry that the technique can be generalized to both volume-integrated and surface-integrated tallies for spatially-uniform perturbations. Finally, the correlating sampling technique was applied in the Shift MC code the estimate k_{eff} perturbations. Speculations were given on what modifications to the algorithm would be necessary to include indirect effects from propagating the perturbation forward in time.

5.2.2 Future Work

First and foremost, the parareal algorithm needs to be implemented in parallel to either confirm or reject the project speedup values given by Eq. (4.9) for the transients presented in Chapter 4. Successful implementations have been seen for both multi-processing and multi-threading applications, but it remains unclear how to optimally allocate processors when parallelizing across multiple dimensions of phase space [39]. Modifications to the parallelization infrastructure of a production-level transport code designed for high-performance computing proved to be significantly more involved than anticipated. Therefore, future endeavors of exploratory parareal implementations should consider the use of less-complicated research-based transport codes.

The parareal implementation between the TML subsolvers used the solution from the EPKE solver as the solution vector $\mathbf{u}(t) = p_{\text{EPKE}}(t)$ to converge on the propagation of the CMFD-level amplitude function. Future implementations should explore the convergence properties of using a solution vector with dependence on higher orders of phase space such, as the one-energy-group CMFD scalar flux $\mathbf{u}(\mathbf{x}, t) = \phi_{\text{CMFD}}(\mathbf{x}, t)$.

The transients simulated in 4 were all prescribed transients and therefore did not consider any forms of reactor feedback. It was alluded to in Section 1.2.1 that one of the drawbacks of the PCQM is that it does not allow for iterative communication between multiphysics solvers. The parareal algorithm for TML

subsolver levels presents an opportunity to recapture that iterative communication. Beyond multiphysics coupling though, parareal implementation should be tested for more rigorous problems that include reactor feedback, and transport-level coupling before they are of broader use to the reactor kinetics community.

It was shown in Chapter 3 that anywhere from 85% to 99% of the overall simulation time for the MC TML implementation was credited to the MC solver. This means that meaningful improvements in the computational performance of the Shift implementation will have to be focused on reducing the number of MC evaluations needed to solve the transient. For parareal applications, this means that the $\mathcal{G}$ and $\mathcal{F}$ solvers should be coupled between two MC levels as anywhere between 85-99% of the computational time is spent at the MC level. A coarse MC solver might be accomplished by rediscrctizing the time domain with a coarser time step and passing the reactor state conditions and time census/fission source iteration banks to the parallel independent initial value problems. Alternatively, the novel approach proposed in Section 4.3.5 could be investigated, where correlated sampling is used to estimate time-propagated perturbations for reaction rate and surface crossing tallies that can be used as input for the TML subsolvers.

References

- [1] L. L. Price, A. A. Alsaed, P. V. Brady, M. B. Gross, E. Hardin, M. Nole, J. L. Prouty, K. Banerjee, and G. G. Davidson, "Postclosure criticality consequence analysis - scoping phase," Apr. 2019. DOI: [10.2172/1762311](https://doi.org/10.2172/1762311). [Online]. Available: <https://www.osti.gov/biblio/1762311>.
- [2] L. L. Price, A. Alsaed, A. C. Barela, P. V. Brady, F. Gelbard, M. Gross, M. Nole, J. L. Prouty, K. Banerjee, S. Bhatt, G. G. Davidson, Z. Fang, R. Howard, S. R. Johnson, S. L. Painter, M. Swinney, and E. Gonzalez, "Preliminary analysis of postclosure DPC criticality consequences," Dec. 2020. DOI: [10.2172/1616378](https://doi.org/10.2172/1616378). [Online]. Available: <https://www.osti.gov/biblio/1616378>.
- [3] A. F. Henry, "The application of reactor kinetics to the analysis of experiments," *Nuclear Science and Engineering*, vol. 3, no. 1, pp. 52–70, 1958. DOI: [10.13182/NSE58-1](https://doi.org/10.13182/NSE58-1). eprint: <https://doi.org/10.13182/NSE58-1>. [Online]. Available: <https://doi.org/10.13182/NSE58-1>.
- [4] A. F. Henry and N. J. Curlee, "Verification of a method for treating neutron space-time problems," *Nuclear Science and Engineering*, vol. 4, no. 6, pp. 727–744, 1958. DOI: [10.13182/NSE4-727](https://doi.org/10.13182/NSE4-727). eprint: <https://doi.org/10.13182/NSE4-727>. [Online]. Available: <https://doi.org/10.13182/NSE4-727>.
- [5] J. B. Yasinsky and A. F. Henry, "Some numerical experiments concerning space-time reactor kinetics behavior," *Nuclear Science and Engineering*, vol. 22, no. 2, pp. 171–181, 1965. DOI: [10.13182/NSE65-A20236](https://doi.org/10.13182/NSE65-A20236). eprint: <https://doi.org/10.13182/NSE65-A20236>. [Online]. Available: <https://doi.org/10.13182/NSE65-A20236>.
- [6] K. O. Ott and D. A. Meneley, "Accuracy of the quasistatic treatment of spatial reactor kinetics," *Nuclear Science and Engineering*, vol. 36, no. 3, pp. 402–411, 1969.
- [7] J. C. Gehin, "A quasi-static polynomial nodal method for nuclear reactor analysis," PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1992.
- [8] S. Dulla, E. Mund, and P. Ravetto, "Accuracy of a predictor-corrector quasi-static method for space-time reactor dynamics," vol. 2006, Jan. 2006.
- [9] S. Dulla, E. H. Mund, and P. Ravetto, "The quasi-static method revisited," *Progress in Nuclear Energy*, vol. 50, 2008.
- [10] C. Patricot, A.-M. Baudron, and O. Fandeur, "The multi-physics improved quasi-static method - application to a neutronics-thermomechanics coupling," *Progress in Nuclear Energy*, vol. 101, pp. 352–359, 2017, Special Issue on the Physics of Reactors International Conference PHYSOR 2016: Unifying Theory and Experiments in the 21st Century, ISSN: 0149-1970. DOI: <https://doi.org/10.1016/j.pnucene.2017.03.021>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0149197017300665>.

- [11] Z. M. Prince and J. C. Ragusa, “Multiphysics reactor-core simulations using the improved quasi-static method,” *Annals of Nuclear Energy*, vol. 125, pp. 186–200, 2019, ISSN: 0306-4549. DOI: <https://doi.org/10.1016/j.anucene.2018.10.056>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S030645491830584X>.
- [12] G. Kooreman and D. P. Griesheimer, “Predictor-corrector quasi-static method for tightly-coupled reactor multiphysics transient calculations,” *EPJ Web Conf.*, vol. 247, p. 07011, 2021. DOI: [10.1051/epjconf/202124707011](https://doi.org/10.1051/epjconf/202124707011). [Online]. Available: <https://doi.org/10.1051/epjconf/202124707011>.
- [13] A. Zhu, Y. Xu, and T. Downar, “A multilevel quasi-static method for pin-resolved transport transient reactor analysis,” *Nuclear Science and Engineering*, vol. 182, 2016.
- [14] Q. Shen, B. Kochunas, Y. Xu, S. Choi, and T. Downar, “Transient multilevel scheme with one-group CMFD acceleration,” *Nuclear Science and Engineering*, vol. 0, no. 0, pp. 1–25, 2021. DOI: [10.1080/00295639.2020.1866388](https://doi.org/10.1080/00295639.2020.1866388). eprint: <https://doi.org/10.1080/00295639.2020.1866388>. [Online]. Available: <https://doi.org/10.1080/00295639.2020.1866388>.
- [15] B. L. Sjenitzer and J. E. Hoogenboom, “Dynamic Monte Carlo for nuclear reactor kinetics calculations,” *Nuclear Science and Engineering*, vol. 175, 2013.
- [16] B. L. Sjenitzer, J. E. Hoogenboom, J. J. Escalante, and V. S. Espinoza, “Coupling of dynamic Monte Carlo with thermal-hydraulic feedback,” *Ann. of Nucl. Energy*, vol. 76, pp. 27–39, 2015.
- [17] M. W. Hackemack, J. C. Ragusa, D. P. Griesheimer, and J. M. Pounders, “A Monte Carlo implementation of the predictor-corrector quasi-static method,” in *M&C 2013*, (Sun Valley, ID), American Nuclear Society, May 2013.
- [18] Y. Jo, B. Cho, and N. Z. Cho, “Nuclear reactor transient analysis by continuous-energy Monte Carlo calculation based on predictor-corrector quasi-static method,” *Nuclear Science and Engineering*, vol. 183, 2016.
- [19] Y. Ban, T. Endo, and A. Yamamoto, “A unified approach for numerical calculation of space-dependent kinetic equation,” *Journal of Nuclear Science and Technology*, vol. 49, no. 5, pp. 496–515, 2012. DOI: [10.1080/00223131.2012.677126](https://doi.org/10.1080/00223131.2012.677126). eprint: <https://doi.org/10.1080/00223131.2012.677126>. [Online]. Available: <https://doi.org/10.1080/00223131.2012.677126>.
- [20] S. C. Shaner, “Development of high fidelity methods for 3d Monte Carlo transient analysis of nuclear reactors,” PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2018.
- [21] M. A. Kreher and B. Forget, “Thermal feedback coupling in a transient Monte Carlo high-order/low-order scheme,” May 2022.
- [22] T. M. Pandya, S. R. Johnson, G. G. Davidson, T. M. Evans, and S. P. Hamilton, “Shift: A massively parallel Monte Carlo radiation transport package,” in *Joint International Conference on Mathematics and Computation, Supercomputing in Nuclear Applications and the Monte Carlo Method*, Nashville, TN: American Nuclear Society, Jan. 2015. [Online]. Available: <https://www.osti.gov/biblio/1185847>.
- [23] A. M. Baudron, J. J. Laautard, Y. Maday, M. K. Riahi, and J. Salomon, “Parareal in time 3d numerical solver for the LWR benchmark neutron diffusion transient model,” *Journal of Computational Physics*, vol. 279, no. 0, pp. 67–79, 2014.

- [24] J. Dabaghi, Y. Maday, and A. Zoia, “A hybrid parareal Monte Carlo algorithm for parabolic problems,” *Journal of Computational and Applied Mathematics*, vol. 420, p. 114 800, 2023, ISSN: 0377-0427. DOI: <https://doi.org/10.1016/j.cam.2022.114800>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377042722004071>.
- [25] B. R. Wienke and R. E. Hiromoto, “Parallel Sn transport algorithms,” *Transport Theory and Statistical Physics*, vol. 15, no. 1-2, pp. 49–59, 1986. DOI: [10.1080/00411458608210444](https://doi.org/10.1080/00411458608210444). eprint: <https://doi.org/10.1080/00411458608210444>. [Online]. Available: <https://doi.org/10.1080/00411458608210444>.
- [26] Y. Y. Azmy, “Multidimensional nodal transport methods for multiple-instruction multiple-data distributed memory machines,” Jan. 1988. [Online]. Available: <https://www.osti.gov/biblio/5221854>.
- [27] M. Yavuz and E. W. Larsen, “Spatial domain decomposition for neutron transport problems,” *Transport Theory and Statistical Physics*, vol. 18, no. 2, pp. 205–219, 1989. DOI: [10.1080/00411458908204321](https://doi.org/10.1080/00411458908204321). eprint: <https://doi.org/10.1080/00411458908204321>. [Online]. Available: <https://doi.org/10.1080/00411458908204321>.
- [28] W. R. Martin, T.-C. Wan, T. S. Abdel-Rahman, T. N. Mudge, and K. Miura, “Monte Carlo photon transport on shared memory and distributed memory parallel processors,” *The International Journal of Supercomputing Applications*, vol. 1, no. 3, pp. 57–74, 1987. DOI: [10.1177/109434208700100306](https://doi.org/10.1177/109434208700100306). eprint: <https://doi.org/10.1177/109434208700100306>. [Online]. Available: <https://doi.org/10.1177/109434208700100306>.
- [29] T. A. Brunner, T. J. Urbatsch, T. M. Evans, and N. A. Gentile, “Comparison of four parallel algorithms for domain decomposed implicit Monte Carlo,” *Journal of Computational Physics*, vol. 212, no. 2, pp. 527–539, 2006, ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2005.07.009>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999105003311>.
- [30] J. A. Ellis, T. M. Evans, S. P. Hamilton, C. Kelley, and T. M. Pandya, “Optimization of processor allocation for domain decomposed Monte Carlo calculations,” *Parallel Computing*, vol. 87, pp. 77–86, 2019, ISSN: 0167-8191. DOI: <https://doi.org/10.1016/j.parco.2019.06.001>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167819119301243>.
- [31] F. B. Brown and W. R. Martin, “Monte Carlo methods for radiation transport analysis on vector computers,” *Progress in Nuclear Energy*, vol. 14, no. 3, pp. 269–299, 1984, ISSN: 0149-1970. DOI: [https://doi.org/10.1016/0149-1970\(84\)90024-6](https://doi.org/10.1016/0149-1970(84)90024-6). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0149197084900246>.
- [32] S. P. Hamilton, S. R. Slattery, and T. M. Evans, “Multigroup Monte Carlo on GPUs: Comparison of history- and event-based algorithms,” *Annals of Nuclear Energy*, vol. 113, pp. 506–518, 2018, ISSN: 0306-4549. DOI: <https://doi.org/10.1016/j.anucene.2017.11.032>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306454917304231>.
- [33] S. P. Hamilton and T. M. Evans, “Continuous-energy Monte Carlo neutron transport on GPUs in the shift code,” *Annals of Nuclear Energy (Oxford)*, vol. 128, Jan. 2019. DOI: [10.1016/j.anucene.2019.01.012](https://doi.org/10.1016/j.anucene.2019.01.012).

- [34] S. P. Hamilton, T. M. Evans, K. E. Royston, and E. D. Biondo, “Domain decomposition in the GPU-accelerated shift Monte Carlo code,” *Annals of Nuclear Energy*, vol. 166, p. 108 687, 2022, ISSN: 0306-4549. DOI: <https://doi.org/10.1016/j.anucene.2021.108687>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306454921005636>.
- [35] A. M. Baudron, J. J. Laautard, Y. Maday, and O. Mula, *The parareal in time algorithm applied to the kinetic neutron diffusion equation*, ser. Domain Decomposition Methods in Science and Engineering XXI. Springer, 2014, ISBN: 978-3-319-05789-7.
- [36] J. L. Lions, Y. Maday, and G. Turinici, “A ‘parareal’ in time discretization of PDE’s,” *Comptes Rendus de l’Académie des Sciences - Series I - Mathematics*, vol. 332, pp. 661–668, 2001.
- [37] D. Guibert and D. Tromeur-Dervout, “Parallel deferred correction method for cfd problems,” in *Parallel Computational Fluid Dynamics 2006*, J. Kwon, A. Ecer, N. Satofuka, J. Periaux, and P. Fox, Eds., Amsterdam: Elsevier Science B.V., 2007, pp. 131–138, ISBN: 978-0-444-53035-6. DOI: <https://doi.org/10.1016/B978-044453035-6/50019-5>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780444530356500195>.
- [38] E. J. Bylaska, J. Q. Weare, and J. H. Weare, “Extending molecular simulation time scales: Parallel in time integrations for high-level quantum chemistry and complex force representations,” *The Journal of Chemical Physics*, vol. 139, no. 7, p. 074 114, 2013. DOI: [10.1063/1.4818328](https://doi.org/10.1063/1.4818328). eprint: <https://doi.org/10.1063/1.4818328>. [Online]. Available: <https://doi.org/10.1063/1.4818328>.
- [39] R. Speck, D. Ruprecht, M. Emmett, M. Boltén, and R. Krause, “A space-time parallel solver for the three-dimensional heat equation,” *CoRR*, vol. abs/1307.7867, 2013. arXiv: [1307.7867](https://arxiv.org/abs/1307.7867). [Online]. Available: <http://arxiv.org/abs/1307.7867>.
- [40] A. Henthaler, B. Southworth, D. Nordsletten, O. Röhrle, R. Falgout, and J. Schroder, “Multilevel convergence analysis of multigrid-reduction-in-time,” *SIAM Journal on Scientific Computing*, vol. 42, A771–A796, Jan. 2020. DOI: [10.1137/19M1238812](https://doi.org/10.1137/19M1238812).
- [41] F. Danieli and S. MacLachlan, *Multigrid reduction in time for non-linear hyperbolic equations*, 2021. arXiv: [2104.09404](https://arxiv.org/abs/2104.09404) [math.NA].
- [42] M. Gander, F. Kwok, and H. Zhang, “Multigrid interpretations of the parareal algorithm leading to an overlapping variant and mgrit,” *Computing and Visualization in Science*, vol. 19, Jul. 2018. DOI: [10.1007/s00791-018-0297-y](https://doi.org/10.1007/s00791-018-0297-y).
- [43] E. E. Lewis and W. F. Miller, “Computational methods of neutron transport,” Jan. 1984. [Online]. Available: <https://www.osti.gov/biblio/5538794>.
- [44] H. Rief, “Stochastic perturbation analysis applied to neutral particle transport,” in *Advances in Nuclear Science and Technology*. Boston, MA: Springer US, 1996, pp. 69–140, ISBN: 978-0-306-47810-9. DOI: [10.1007/0-306-47810-2_2](https://doi.org/10.1007/0-306-47810-2_2). [Online]. Available: https://doi.org/10.1007/0-306-47810-2_2.
- [45] B. C. Kiedrowski, “Review of early 21st-century Monte Carlo perturbation and sensitivity techniques for k-eigenvalue radiation transport calculations,” *Nuclear Science and Engineering*, vol. 185, no. 3, pp. 426–444, 2017. DOI: [10.1080/00295639.2017.1283153](https://doi.org/10.1080/00295639.2017.1283153). eprint: <https://doi.org/10.1080/00295639.2017.1283153>. [Online]. Available: <https://doi.org/10.1080/00295639.2017.1283153>.

- [46] M. Nakagawa and T. Asaoka, "Improvement of correlated sampling Monte Carlo methods for reactivity calculations," *Journal of Nuclear Science and Technology*, vol. 15, no. 6, pp. 400–410, 1978. DOI: [10.1080/18811248.1978.9735527](https://doi.org/10.1080/18811248.1978.9735527). eprint: <https://www.tandfonline.com/doi/pdf/10.1080/18811248.1978.9735527>. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1080/18811248.1978.9735527>.
- [47] T. Kitada, A. Yamane, and T. Takeda, "Improvement of the correlated sampling method in Monte Carlo perturbation theory," Sep. 1996, A–212.
- [48] Y. Nagaya and T. Mori, "Impact of perturbed fission source on the effective multiplication factor in Monte Carlo perturbation calculations," *Journal of Nuclear Science and Technology*, vol. 42, no. 5, pp. 428–441, 2005. DOI: [10.1080/18811248.2005.9726411](https://doi.org/10.1080/18811248.2005.9726411). eprint: <https://www.tandfonline.com/doi/pdf/10.1080/18811248.2005.9726411>. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1080/18811248.2005.9726411>.
- [49] K. O. Ott and R. J. Neuhold, *Introductory nuclear reactor dynamics*. American Nuclear Society, 1985, ISBN: 978-0-894-48029-4.
- [50] E. Greenspan, "Developments in perturbation theory," in *Advances in Nuclear Science and Technology*, E. J. Henley and J. Lewins, Eds., Academic Press, 1976, pp. 181–268, ISBN: 978-0-12-029309-4. DOI: <https://doi.org/10.1016/B978-0-12-029309-4.50011-8>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780120293094500118>.
- [51] M. A. Cooper and E. W. Larsen, "Automated weight windows for global Monte Carlo particle transport calculations," *Nuclear Science and Engineering*, vol. 137, no. 1, pp. 1–13, 2001.
- [52] T. L. Becker, "Hybrid Monte Carlo/deterministic methods for radiation shielding problems," PhD thesis, University of Michigan, Ann Arbor, MI, 2009.
- [53] J. C. Wagner and A. Haghighat, "Automated variance reduction of Monte Carlo shielding calculations using the discrete ordinates adjoint function," *Nuclear Science and Engineering*, vol. 128, pp. 186–208, 1998.
- [54] J. C. Wagner, D. E. Peplow, and S. W. Mosher, "Fw-cadis method for global and regional variance reduction of monte carlo radiation transport calculations," *Nuclear Science and Engineering*, vol. 176, pp. 37–57, 2014.
- [55] M. Becker, "A generalized formulation of point nuclear reactor kinetics equations," *Nuclear Science and Engineering*, vol. 31, no. 3, pp. 458–464, 1968. DOI: [10.13182/NSE68-A17588](https://doi.org/10.13182/NSE68-A17588). [Online]. Available: <https://doi.org/10.13182/NSE68-A17588>.
- [56] B. C. Kiedrowski, "Adjoint weighting for continuous-energy Monte Carlo radiation transport," PhD thesis, University of Wisconsin, Madison, WI, Jan. 2009.
- [57] B. Mitchell, "Taylor series methods for the solution of the point reactor kinetics equations," *Annals of Nuclear Energy*, vol. 4, no. 4, pp. 169–176, 1977, ISSN: 0306-4549. DOI: [https://doi.org/10.1016/0306-4549\(77\)90062-7](https://doi.org/10.1016/0306-4549(77)90062-7). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0306454977900627>.
- [58] J. Sanchez, "On the numerical solution of the point reactor kinetics equations by generalized runge-kutta methods," *Nuclear Science and Engineering*, vol. 103, no. 1, pp. 94–99, 1989. DOI: [10.13182/NSE89-A23663](https://doi.org/10.13182/NSE89-A23663). eprint: <https://doi.org/10.13182/NSE89-A23663>. [Online]. Available: <https://doi.org/10.13182/NSE89-A23663>.

- [59] Y. Cai, X. Peng, Q. Li, and K. Wang, “A numerical solution to the nonlinear point kinetics equations using magnus expansion,” *Annals of Nuclear Energy*, vol. 89, pp. 84–89, 2016, ISSN: 0306-4549. DOI: <https://doi.org/10.1016/j.anucene.2015.11.021>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306454915005587>.
- [60] J. A. W. da Nobrega, “Time integration methods for reactor kinetics,” Dec. 1971. DOI: [10.2172/4644788](https://doi.org/10.2172/4644788). [Online]. Available: <https://www.osti.gov/biblio/4644788>.
- [61] Q. Shen, “Robust and efficient methods in transient whole-core neutron transport calculations,” PhD thesis, University of Michigan, Ann Arbor, MI, 2021.
- [62] W. H. Reed, “Finite difference techniques for the solution of the reactor kinetics equations,” PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1969.
- [63] J. S. Hendricks and A. F. Henry, “Finite difference solution of the time dependent neutron group diffusion equations,” Aug. 1975. DOI: [10.2172/4140727](https://doi.org/10.2172/4140727). [Online]. Available: <https://www.osti.gov/biblio/4140727>.
- [64] K. Fuchs, *Efficiency for Very Slow Assembly*, LA-596. 1952. [Online]. Available: <https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-00596>.
- [65] L. Nordheim, “Pile kinetics,” War Department, Corps of Engineers, Office of the District Engineer, Manhattan District, Tech. Rep. MDDC-35, 1946. [Online]. Available: https://books.google.com/books?id=SquhaRFvX%5C_YC.
- [66] G. E. Hansen, *Burst characteristics associated with the slow assembly of fissionable materials*, LA-1441. 1952. [Online]. Available: <https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-01441>.
- [67] D. E. Cullen and C. R. Weisbin, “Exact doppler broadening of tabulated cross sections,” *Nuclear Science and Engineering*, vol. 60, no. 3, pp. 199–229, 1976. DOI: [10.13182/NSE76-1](https://doi.org/10.13182/NSE76-1). eprint: <https://doi.org/10.13182/NSE76-1>. [Online]. Available: <https://doi.org/10.13182/NSE76-1>.
- [68] B. R. Bassett, “Meshless methods for the neutron transport equation,” PhD thesis, University of Michigan, Ann Arbor, MI, 2018.
- [69] B. R. Herman, B. Forget, K. Smith, and B. N. Aviles, “Improved diffusion coefficients generated from Monte Carlo codes,” in *M&C 2013*, (Sun Valley, ID), American Nuclear Society, May 2013.
- [70] E. M. Gelbard, “Application of spherical harmonics methods to reactor problems,” Bettis Atomic Power Laboratory, Tech. Rep. WAPD-BT-20, 1960. DOI: [10.2172/4127649](https://doi.org/10.2172/4127649).
- [71] —, “Simplified spherical harmonics equations and their use in shielding problems,” Bettis Atomic Power Laboratory, Tech. Rep. WAPD-T-1182, 1961.
- [72] —, “Applications of the simplified spherical harmonics equations in spherical geometry,” Bettis Atomic Power Laboratory, Tech. Rep. WAPD-TM-294, 1962.
- [73] E. W. Larsen, J. E. Morel, and J. M. McGhee, “Asymptotic derivation of the multigroup P1 and simplified PN equations with anisotropic scattering,” *Nuclear Science and Engineering*, vol. 123, no. 3, pp. 328–342, 1996. DOI: [10.13182/NSE123-328](https://doi.org/10.13182/NSE123-328). eprint: <https://doi.org/10.13182/NSE123-328>. [Online]. Available: <https://doi.org/10.13182/NSE123-328>.

- [74] P. S. Brantley and E. W. Larsen, "The simplified P3 approximation," *Nuclear Science and Engineering*, vol. 134, no. 1, pp. 1–21, 2000. DOI: [10.13182/NSE134-01](https://doi.org/10.13182/NSE134-01). eprint: <https://doi.org/10.13182/NSE134-01>. [Online]. Available: <https://doi.org/10.13182/NSE134-01>.
- [75] D. R. Ferguson and K. L. Derstine, "Optimized iteration strategies and data management considerations for fast reactor finite difference diffusion theory codes," *Nuclear Science and Engineering*, vol. 64, no. 2, pp. 593–604, 1977. DOI: [10.13182/NSE77-5](https://doi.org/10.13182/NSE77-5). eprint: <https://doi.org/10.13182/NSE77-5>. [Online]. Available: <https://doi.org/10.13182/NSE77-5>.
- [76] G. R. Cefus and E. W. Larsen, "Stability analysis of coarse-mesh rebalance," *Nuclear Science and Engineering*, vol. 105, no. 1, pp. 31–39, 1990. DOI: [10.13182/NSE88-117](https://doi.org/10.13182/NSE88-117). eprint: <https://doi.org/10.13182/NSE88-117>. [Online]. Available: <https://doi.org/10.13182/NSE88-117>.
- [77] H. J. Kopp, "Synthetic method solution of the transport equation," *Nuclear Science and Engineering*, vol. 17, no. 1, pp. 65–74, 1963. DOI: [10.13182/NSE63-1](https://doi.org/10.13182/NSE63-1). eprint: <https://doi.org/10.13182/NSE63-1>. [Online]. Available: <https://doi.org/10.13182/NSE63-1>.
- [78] E. M. Gelbard and L. A. Hageman, "The synthetic method as applied to the Sn equations," *Nuclear Science and Engineering*, vol. 37, no. 2, pp. 288–298, 1969. DOI: [10.13182/NSE69-A20689](https://doi.org/10.13182/NSE69-A20689). eprint: <https://doi.org/10.13182/NSE69-A20689>. [Online]. Available: <https://doi.org/10.13182/NSE69-A20689>.
- [79] R. E. Alcouffe, "Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations," *Nuclear Science and Engineering*, vol. 64, no. 2, pp. 344–355, 1977. DOI: [10.13182/NSE77-1](https://doi.org/10.13182/NSE77-1). eprint: <https://doi.org/10.13182/NSE77-1>. [Online]. Available: <https://doi.org/10.13182/NSE77-1>.
- [80] A. Yamamoto, "Generalized coarse-mesh rebalance method for acceleration of neutron transport calculations," *Nuclear Science and Engineering*, vol. 151, no. 3, pp. 274–282, 2005. DOI: [10.13182/NSE151-274](https://doi.org/10.13182/NSE151-274). eprint: <https://doi.org/10.13182/NSE151-274>. [Online]. Available: <https://doi.org/10.13182/NSE151-274>.
- [81] K. S. Smith, "Nodal method storage reduction by nonlinear iteration," *Transactions of the American Nuclear Society*, vol. 44, pp. 265–266, 1983.
- [82] M. J. Lee, H. G. Joo, D. Lee, and K. Smith, "Coarse mesh finite difference formulation for accelerated Monte Carlo eigenvalue calculation," *Annals of Nuclear Energy*, vol. 65, pp. 101–113, 2014, ISSN: 0306-4549. DOI: <https://doi.org/10.1016/j.anucene.2013.10.025>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306454913005550>.
- [83] P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, and B. Forget, "OpenMC: A state-of-the-art Monte Carlo code for research and development," *Ann. of Nucl. Energy*, vol. 82, pp. 90–97, 2015.
- [84] J. A. Kulesza, T. R. Adams, J. C. Armstrong, S. R. Bolding, F. B. Brown, J. S. Bull, T. P. Burke, A. R. Clark, R. A. Forster III, J. F. Giron, A. S. Grieve, C. J. Josey, R. L. Martz, G. W. McKinney, E. J. Pearson, M. E. Rising, C. J. Solomon Jr., S. Swaminarayan, T. J. Trahan, S. C. Wilson, and A. J. Zukaitis, "MCNP® Code Version 6.3.0 Theory & User Manual," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-22-30006, Rev. 1, Sep. 2022. DOI: [10.2172/1889957](https://doi.org/10.2172/1889957). [Online]. Available: <https://www.osti.gov/biblio/1889957>.
- [85] T. T. P. Team, "The Trilinos Project Website," Tech. Rep.

- [86] W. A. Wieselquist, R. A. Lefebvre, and M. A. Jessee, "SCALE code system," Oak Ridge National Laboratory, Oak Ridge, TN, USA, Tech. Rep. ORNL/TM-2005/39.
- [87] The HDF Group, "Hierarchical Data Format, version 5," Tech. Rep., 1997-2022. [Online]. Available: <https://www.hdfgroup.org/HDF5/>.
- [88] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, and K. T. Clarno, "Denovo: A new three-dimensional parallel discrete ordinates code in scale," *Nuclear Technology*, vol. 171, no. 2, pp. 171–200, 2010. DOI: [10.13182/NT171-171](https://doi.org/10.13182/NT171-171). eprint: <https://doi.org/10.13182/NT171-171>. [Online]. Available: <https://doi.org/10.13182/NT171-171>.
- [89] G. G. Davidson, T. M. Pandya, S. R. Johnson, T. M. Evans, A. E. Isotalo, C. A. Gentry, and W. A. Wieselquist, "Nuclide depletion capabilities in the shift Monte Carlo code," *Annals of Nuclear Energy*, vol. 114, pp. 259–276, 2018, ISSN: 0306-4549. DOI: <https://doi.org/10.1016/j.anucene.2017.11.042>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306454917304322>.
- [90] K. O. Ott and J. T. Madell, "Quasistatic treatment of spatial phenomena in reactor dynamics," *Nuclear Science and Engineering*, vol. 26, no. 4, 1966.
- [91] T. E. Booth, "A weight (charge) conserving importance-weighted comb for Monte Carlo," Mar. 1996. [Online]. Available: <https://www.osti.gov/biblio/204256>.
- [92] E. E. Lewis, M. A. Smith, N. Tsoulfanidis, G. Palmiotti, T. A. Taiwo, and R. N. Blomquist, "Benchmark specification for deterministic 2-D/3-D MOX fuel assembly transport calculations without spatial homogenisation (C5G7 MOX)," 2001.
- [93] C. Cavarec, J. Perron, D. Verwaerde, and J. West, "Benchmark calculations of power distribution within assemblies," Nuclear Energy Agency, Tech. Rep., 1994.
- [94] V. Boyarinov, P. Fomichenko, J. Hou, K. Ivanov, A. Aures, W. Zwermann, and K. Velkov, "Deterministic time-dependent neutron transport benchmark without spatial homogenization (C5G7-TD)," *Nuclear Energy Agency Organisation for Economic Co-operation and Development (NEA-OECD), Paris, France*, 2016.
- [95] J. Hou, K. N. Ivanov, V. F. Boyarinov, and P. A. Fomichenko, "OECD/NEA benchmark for time-dependent neutron transport calculations without spatial homogenization," *Nuclear Engineering and Design*, vol. 317, pp. 177–189, 2017.
- [96] B. Kouchunas, D. Jabaay, T. Downar, B. S. Collins, S. G. Stimpson, A. T. Godfrey, K. S. Kim, J. C. Gehin, S. Palmtag, and F. Franceschini, "Validation and application of the 3d neutron transport mpact within casl vera-cs," Jan. 2015. [Online]. Available: <https://www.osti.gov/biblio/1214019>.
- [97] Q. Shen, Y. Wang, D. Jabaay, B. Kochunas, and T. Downar, "Transient analysis of C5G7-TD benchmark with MPACT," *Annals of Nuclear Energy*, vol. 125, pp. 107–120, 2019, ISSN: 0306-4549. DOI: <https://doi.org/10.1016/j.anucene.2018.10.049>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306454918305772>.
- [98] M. A. Kreher, S. Shaner, B. Forget, and K. Smith, "Frequency transform method for transient analysis of nuclear reactors in Monte Carlo," *Nuclear Science and Engineering*, vol. 0, no. 0, Jun. 2022. DOI: [10.1080/00295639.2022.2067739](https://doi.org/10.1080/00295639.2022.2067739).

- [99] M. D. DeHart, Z. Mausolff, Z. Weems, D. Popp, K. Smith, F. Shriver, S. Goluoglu, Z. Prince, and J. Ragusa, “Preliminary results for the OECD/NEA time dependent benchmark using rattlesnake, rattlesnake-iqs and TDKENO,” Aug. 2016. DOI: [10.2172/1364489](https://doi.org/10.2172/1364489). [Online]. Available: <https://www.osti.gov/biblio/1364489>.
- [100] E. D. Biondo, “Implementation of the windowed multipole method in shift,” Jul. 2021. DOI: [10.2172/1818696](https://doi.org/10.2172/1818696). [Online]. Available: <https://www.osti.gov/biblio/1818696>.
- [101] “Final safety analyses report for the hi-storm 100 cask system,” Holtec International, Tech. Rep., 2010.
- [102] W. B. Marshall, K. B. Bekar, J. B. Clarity, M. Dupont, R. A. Lefebvre, and E. M. Saylor, “Keno-vi primer: Performing calculations using scale’s criticality safety analysis sequence (csas6) with fulcrum,” Dec. 2020. DOI: [10.2172/1760129](https://doi.org/10.2172/1760129). [Online]. Available: <https://www.osti.gov/biblio/1760129>.
- [103] A. Nelson, “Improved convergence rate of multi-group scattering moment tallies for Monte Carlo neutron transport codes,” PhD thesis, University of Michigan, Ann Arbor, MI, Sep. 2014.
- [104] K. P. Keady, “Acceleration and stabilization methods for Monte Carlo reactor core k-eigenvalue problems,” PhD thesis, University of Michigan, Ann Arbor, MI, 2016.
- [105] E. Wolters, E. Larsen, and W. Martin, “Hybrid Monte Carlo-CMFD methods for accelerating fission source convergence,” *Nuclear Science and Engineering*, vol. 174, pp. 286–299, Jul. 2013. DOI: [10.13182/NSE12-72](https://doi.org/10.13182/NSE12-72).
- [106] K. P. Keady and E. W. Larsen, “Stability of Monte Carlo k-eigenvalue simulations with CMFD feedback,” *Journal of Computational Physics*, vol. 321, pp. 947–964, 2016, ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2016.06.002>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999116302273>.
- [107] M. J. Gander, “50 years of parallel time integration,” in *Multiple Shooting and Time Domain Decomposition Methods*, T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, Eds., Springer International Publishing, 2015, pp. 69–113, ISBN: 978-3-319-23321-5.
- [108] D. Ruprecht, “Convergence of parareal with spatial coarsening,” *PAMM*, vol. 14, no. 1, pp. 1031–1034, 2014. DOI: <https://doi.org/10.1002/pamm.201410490>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/pamm.201410490>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/pamm.201410490>.
- [109] T. Downar, Y. Xu, and V. Seker, “PARCS v3.0 User Manual,” U.S. NRC, Tech. Rep. 20, 2010.
- [110] R. J. LaBauve, “Bare, highly enriched uranium sphere (godiva), HEU-MET-FAST-001,” Nuclear Energy Agency, OECD, Tech. Rep. NEA/NSC/DOC(95)03/II, Sep. 2001. [Online]. Available: https://books.google.com/books?id=SquhaRFvX%5C_YC.
- [111] B. Kiedrowski, F. Brown, and P. Wilson, “Adjoint-weighted tallies for k-eigenvalue calculations with continuous-energy Monte Carlo,” *Nuclear Science and Engineering*, vol. 168, pp. 226–241, Jul. 2011. DOI: [10.13182/NSE10-22](https://doi.org/10.13182/NSE10-22).

Appendix A

Temporal Integration of the Precursor Equations

To integrate the precursor equations, first introduce an integrating factor of $e^{\lambda_j(\mathbf{x},t)t}$ to Eq. (2.122b) and rearrange to arrive at the form

$$\begin{aligned} \frac{\partial C_j(\mathbf{x},t)}{\partial t} e^{\lambda_j(\mathbf{x},t)t} + \lambda_j(\mathbf{x},t) C_j(\mathbf{x},t) e^{\lambda_j(\mathbf{x},t)t} &= \beta_j(\mathbf{x},t) \frac{1}{k_{\text{eff}}^0} \mathbf{F}\psi(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t) e^{\lambda_j(\mathbf{x},t)t} \\ \frac{\partial}{\partial t} \left(C_j(\mathbf{x},t) e^{\lambda_j(\mathbf{x},t)t} \right) &= \beta_j(\mathbf{x},t) \frac{1}{k_{\text{eff}}^0} \mathbf{F}\psi(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t) e^{\lambda_j(\mathbf{x},t)t}. \end{aligned} \quad (\text{A.1})$$

Equation (A.1) is then integrated over the range $t^{n-1} \leq t \leq t^n$,

$$\int_{t^{n-1}}^{t^n} \left[\frac{\partial}{\partial t} \left(C_j(\mathbf{x},t) e^{\lambda_j(\mathbf{x},t)t} \right) = \beta_j(\mathbf{x},t) \frac{1}{k_{\text{eff}}^0} \mathbf{F}\psi(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t) e^{\lambda_j(\mathbf{x},t)t} \right] dt. \quad (\text{A.2})$$

The left-hand side of Eq. (A.2) becomes

$$\begin{aligned} \int_{t^{n-1}}^{t^n} \frac{\partial}{\partial t} \left(C_j(\mathbf{x},t) e^{\lambda_j(\mathbf{x},t)t} \right) dt &= C_j^n(\mathbf{x}) e^{\lambda_j^n(\mathbf{x})t^n} - C_j^{n-1}(\mathbf{x}) e^{\lambda_j^{n-1}(\mathbf{x})t^{n-1}} \\ &= C_j^n(\mathbf{x}) e^{\lambda_j^n(\mathbf{x})\Delta t^n} - C_j^{n-1}(\mathbf{x}). \end{aligned} \quad (\text{A.3})$$

To integrate the right-hand side of Eq. (A.2), an assumption of the temporal behavior of the delayed neutron production from fission needs to be made. In the following sections, the derivations for both linear and quadratic temporal behavior are presented. First, let the delayed neutron production from fission be represented as

$$g(\mathbf{x}, \hat{\mathbf{\Omega}}, Et) = \beta_j(\mathbf{x}, t) \frac{1}{k_{\text{eff}}^0} \mathbf{F}\psi(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t). \quad (\text{A.4})$$

A.1 Linear Approximation of the Delayed Neutron Source

Let $g(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t)$ be approximated as varying linearly in time and assume the form

$$g(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t) \approx \frac{1}{\Delta t^n} \left(g^n(\mathbf{x}, \hat{\mathbf{\Omega}}, E) - g^{n-1}(\mathbf{x}, \hat{\mathbf{\Omega}}, E) \right) t + g^{n-1}(\mathbf{x}, \hat{\mathbf{\Omega}}, E). \quad (\text{A.5})$$

After substituting Eq. (A.5) into Eq. (A.2), the right-hand side becomes

$$\begin{aligned}
& \int_{t^{n-1}}^{t^n} \beta_j(\mathbf{x}, t) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi(\mathbf{x}, \hat{\Omega}, E, t) e^{\lambda_j(\mathbf{x}, t)t} dt \\
&= \int_{t^{n-1}}^{t^n} g(\mathbf{x}, t) e^{\lambda_j(\mathbf{x}, t)t} dt \\
&= \int_{t^{n-1}}^{t^n} \left[\frac{1}{\Delta t^n} \left(g^n(\mathbf{x}, \hat{\Omega}, E) - g^{n-1}(\mathbf{x}, \hat{\Omega}, E) \right) t + g^{n-1}(\mathbf{x}, \hat{\Omega}, E) \right] e^{\lambda_j(\mathbf{x}, t)t} dt \\
&= g^n(\mathbf{x}, \hat{\Omega}, E) \frac{\lambda_j^n(\mathbf{x}) \Delta t^n e^{\lambda_j^n(\mathbf{x}) \Delta t^n} + 1 - e^{\lambda_j^n(\mathbf{x}) \Delta t^n}}{\lambda_j^n(\mathbf{x}) \Delta t^n} + g^{n-1}(\mathbf{x}, \hat{\Omega}, E) \frac{-\lambda_j^n(\mathbf{x}) \Delta t^n - 1 + e^{\lambda_j^n(\mathbf{x}) \Delta t^n}}{\lambda_j^n(\mathbf{x}) \Delta t^n} \\
&= \beta_j^n(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^n(\mathbf{x}, \hat{\Omega}, E) \frac{\lambda_j^n(\mathbf{x}) \Delta t^n e^{\lambda_j^n(\mathbf{x}) \Delta t^n} + 1 - e^{\lambda_j^n(\mathbf{x}) \Delta t^n}}{\lambda_j^n(\mathbf{x}) \Delta t^n} \\
&+ \beta_j^{n-1}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) \frac{-\lambda_j^n(\mathbf{x}) \Delta t^n - 1 + e^{\lambda_j^n(\mathbf{x}) \Delta t^n}}{\lambda_j^n(\mathbf{x}) \Delta t^n}. \tag{A.6}
\end{aligned}$$

By equating Eqs. (A.3) and (A.6), the temporal discretization of Eq. (2.122b) becomes

$$\begin{aligned}
C_j^n(\mathbf{x}) e^{\lambda_j^n(\mathbf{x}) \Delta t^n} - C_j^{n-1}(\mathbf{x}) &= \beta_j^n(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^n(\mathbf{x}, \hat{\Omega}, E) \frac{\lambda_j^n(\mathbf{x}) \Delta t^n e^{\lambda_j^n(\mathbf{x}) \Delta t^n} + 1 - e^{\lambda_j^n(\mathbf{x}) \Delta t^n}}{\lambda_j^n(\mathbf{x}) \Delta t^n} \\
&+ \beta_j^{n-1}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) \frac{-\lambda_j^n(\mathbf{x}) \Delta t^n - 1 + e^{\lambda_j^n(\mathbf{x}) \Delta t^n}}{\lambda_j^n(\mathbf{x}) \Delta t^n}. \tag{A.7}
\end{aligned}$$

Finally, solving Eq. (A.7) for $C_j^n(\mathbf{x})$ gives

$$C_j^n(\mathbf{x}) = C_j^{n-1}(\mathbf{x}) f_{1,k} + \beta_j^{n-1}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) f_{2,k} + \beta_j^n(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^n(\mathbf{x}, \hat{\Omega}, E) f_{3,k}, \tag{A.8}$$

where

$$f_{1,k} = e^{-\lambda_j^n(\mathbf{x}) \Delta t^n}, \tag{A.9a}$$

$$f_{2,k} = \frac{1 - e^{-\lambda_j^n(\mathbf{x}) \Delta t^n} - \lambda_j^n(\mathbf{x}) \Delta t^n e^{-\lambda_j^n(\mathbf{x}) \Delta t^n}}{\lambda_j^n(\mathbf{x}) \Delta t^n}, \tag{A.9b}$$

$$f_{3,k} = \frac{e^{-\lambda_j^n(\mathbf{x}) \Delta t^n} \left(1 - e^{\lambda_j^n(\mathbf{x}) \Delta t^n} + \lambda_j^n(\mathbf{x}) \Delta t^n e^{\lambda_j^n(\mathbf{x}) \Delta t^n} \right)}{\lambda_j^n(\mathbf{x}) \Delta t^n}. \tag{A.9c}$$

A.2 Quadratic Approximation of the Delayed Neutron Source

Let $g(\mathbf{x}, \hat{\Omega}, E, t)$ be approximated as varying quadratically with Lagrange polynomial interpolation. The standard Lagrange polynomial interpolation has the quadratic form

$$P(t) \approx p^b \frac{(t-a)(t-m)}{(b-a)(b-m)} + p^m \frac{(t-a)(t-b)}{(m-a)(m-b)} + p^a \frac{(t-m)(t-b)}{(a-m)(a-b)}, \tag{A.10}$$

where $P(t)$ is the function approximation, b is the right endpoint, a is the left endpoint, m is the midpoint, and $g(t)$ is the function being approximated. An example interpolation is depicted in Fig. A.1. The

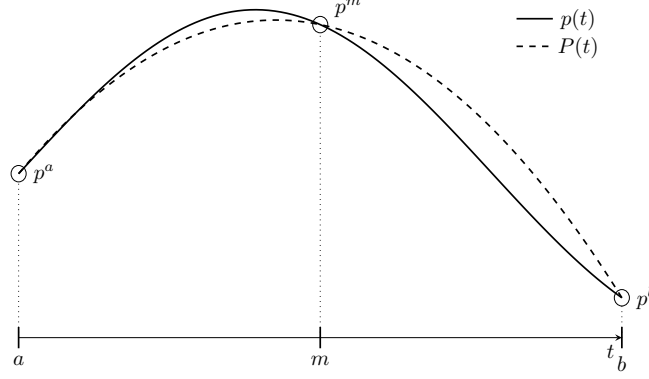


Figure A.1: Quadratic Lagrange polynomial interpolation $P(t)$ of arbitrary function $p(t)$.

three evaluations of $g(\mathbf{x}, \hat{\Omega}, E, t)$ are defined as: $g^b(\mathbf{x}, \hat{\Omega}, E) = \beta_j^n(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^n(\mathbf{x}, \hat{\Omega}, E)$, $g^a(\mathbf{x}, \hat{\Omega}, E) = \beta_j^{n-2}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-2}(\mathbf{x}, \hat{\Omega}, E)$, $g^m(\mathbf{x}, \hat{\Omega}, E) = \beta_j^{n-1}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E)$, $b = t^n$, $a = t^{n-2}$, and $m = t^{n-1}$. This yields the approximation

$$\begin{aligned}
\beta_j(\mathbf{x}, t) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi(\mathbf{x}, \hat{\Omega}, E, t) &\approx \beta_j^n(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^n(\mathbf{x}, \hat{\Omega}, E) \frac{(t - t^{n-2})(t - t^{n-1})}{(t^n - t^{n-2})(t^n - t^{n-1})} \\
&+ \beta_j^{n-1}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) \frac{(t - t^{n-2})(t - t^n)}{(t^{n-1} - t^{n-2})(t^{n-1} - t^n)} \\
&+ \beta_j^{n-2}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-2}(\mathbf{x}, \hat{\Omega}, E) \frac{(t - t^{n-1})(t - t^n)}{(t^{n-2} - t^{n-1})(t^{n-2} - t^n)} \\
&= \beta_j^n(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^n(\mathbf{x}, \hat{\Omega}, E) \frac{\tilde{t}^2 + \tilde{t} \gamma \Delta t^n}{(1 + \gamma)(\Delta t^n)^2} \\
&+ \beta_j^{n-1}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) \left[1 - \frac{\tilde{t}^2 + (\gamma - 1) \Delta t^n \tilde{t}}{\gamma (\Delta t^n)^2} \right] \\
&+ \beta_j^{n-2}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-2}(\mathbf{x}, \hat{\Omega}, E) \frac{\tilde{t}^2 - \tilde{t} \Delta t^n}{(1 + \gamma) \gamma (\Delta t^n)^2}, \tag{A.11}
\end{aligned}$$

where

$$\gamma = \frac{\Delta t^{n-1}}{\Delta t^n}, \tag{A.12a}$$

$$\tilde{t} = t - t^{n-1}. \tag{A.12b}$$

Substituting Eq. (A.11) into the right-hand side of Eq. (A.2), carrying through with the integration, and solving for C_j^n gives

$$\begin{aligned}
C_j^n(\mathbf{x}) &= \Omega_j^0(\tilde{\lambda}_j^n(\mathbf{x})) C_j^{n-1}(\mathbf{x}) + \frac{1}{\lambda_j^n(\mathbf{x})} \beta_j^n(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^n(\mathbf{x}, \hat{\Omega}, E) \Omega_j^n(\tilde{\lambda}_j^n(\mathbf{x})) \\
&+ \frac{1}{\lambda_j^n(\mathbf{x})} \beta_j^{n-1}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-1}(\mathbf{x}, \hat{\Omega}, E) \Omega_j^{n-1}(\tilde{\lambda}_j^n(\mathbf{x})) \\
&+ \frac{1}{\lambda_j^n(\mathbf{x})} \beta_j^{n-2}(\mathbf{x}) \frac{1}{k_{\text{eff}}^0} \mathbf{F} \psi^{n-2}(\mathbf{x}, \hat{\Omega}, E) \Omega_j^{n-2}(\tilde{\lambda}_j^n(\mathbf{x})) \tag{A.13}
\end{aligned}$$

where

$$\Omega_j^0 \left(\tilde{\lambda}_j^n(\mathbf{x}) \right) = E \left(\tilde{\lambda}_j^n(\mathbf{x}) \right), \quad (\text{A.14a})$$

$$\Omega_j^n \left(\tilde{\lambda}_j^n(\mathbf{x}) \right) = \frac{\kappa_2 \left(\tilde{\lambda}_j^n(\mathbf{x}) \right) + \gamma \kappa_1 \left(\tilde{\lambda}_j^n(\mathbf{x}) \right)}{(1 + \gamma)}, \quad (\text{A.14b})$$

$$\Omega_j^{n-1} \left(\tilde{\lambda}_j^n(\mathbf{x}) \right) = \left[\Omega_j^0 \left(\tilde{\lambda}_j^n(\mathbf{x}) \right) - \frac{\kappa_2 \left(\tilde{\lambda}_j^n(\mathbf{x}) \right) + (\gamma - 1) \kappa_1 \left(\tilde{\lambda}_j^n(\mathbf{x}) \right)}{\gamma} \right], \quad (\text{A.14c})$$

$$\Omega_j^{n-2} \left(\tilde{\lambda}_j^n(\mathbf{x}) \right) = \frac{\kappa_2 \left(\tilde{\lambda}_j^n(\mathbf{x}) \right) - \kappa_1 \left(\tilde{\lambda}_j^n(\mathbf{x}) \right)}{(1 + \gamma) \gamma}, \quad (\text{A.14d})$$

$$\kappa_0(x) = 1 - e^{-x}, \quad (\text{A.14e})$$

$$\kappa_1(x) = 1 - \frac{\kappa_0(x)}{x}, \quad (\text{A.14f})$$

$$\kappa_2(x) = 1 - \frac{2\kappa_1(x)}{x}, \quad (\text{A.14g})$$

$$E(x) = e^{-x}, \quad (\text{A.14h})$$

$$\tilde{\lambda}_j^n(\mathbf{x}) = \lambda_j^n(\mathbf{x}) \Delta t^n. \quad (\text{A.14i})$$

Appendix B

Analytic Solution for Asymmetric Reactivity Ramp

Linear reactivity ramps without feedback, such as those seen in Fig. B.1, are some of the few scenarios where analytic solutions for the Exact Point Kinetics Equations (EPKE) can be derived. Consider the EPKEs given in Eqs. (B.1) for a single group of delayed neutron precursors. Assume that point kinetics parameters β^{eff} , Λ , and λ are constant in time and let temporal derivatives be written in dot notation such that $\dot{p}(t) = \frac{dp(t)}{dt}$

$$\dot{p}(t) = \frac{\rho(t) - \beta^{\text{eff}}}{\Lambda} p(t) + \frac{1}{\Lambda} \lambda \zeta(t) , \quad (\text{B.1a})$$

$$\dot{\zeta}(t) = -\lambda \zeta(t) + \beta^{\text{eff}} p(t) . \quad (\text{B.1b})$$

The prompt jump approximation is applied by multiplying Eq. (B.1a) by Λ and letting $\Lambda \rightarrow 0$ to arrive

$$0 = (\rho(t) - \beta^{\text{eff}}) p(t) + \lambda \zeta(t) . \quad (\text{B.2})$$

Differentiating Eq. (B.2) with respect to time gives

$$0 = (\rho(t) - \beta^{\text{eff}}) \dot{p}(t) + p(t) \dot{\rho}(t) + \lambda \dot{\zeta}(t) . \quad (\text{B.3})$$

Next, perform the algebraic operation Eq. (B.1b) + Eq. (B.2) + Eq. (B.3)/ λ and solve for $\dot{p}(t)$ to arrive

$$\dot{p}(t) = \frac{\lambda \rho(t) + \dot{\rho}(t)}{\beta^{\text{eff}} - \rho(t)} p(t) . \quad (\text{B.4})$$

Divide Eq. (B.4) by $p(t)$ and note that $\lambda \rho(t) = \lambda \beta^{\text{eff}} - \lambda (\beta^{\text{eff}} - \rho(t))$ to arrive at

$$\frac{\dot{p}(t)}{p(t)} = \frac{\lambda \beta^{\text{eff}} + \dot{\rho}(t)}{\beta^{\text{eff}} - \rho(t)} - \lambda . \quad (\text{B.5})$$

Next, define the reactivity ramp to start at t_s with duration $\Delta t = t - t_s$ and assume that a linear change in ρ is given by the Taylor Expansion $\rho(t_s + \Delta t) = \rho_s + \dot{\rho} \Delta t$. Then, for $\dot{\rho} \neq 0$, Eq. (B.5) becomes

$$\frac{\dot{p}(t)}{p(t)} + \lambda = \frac{\lambda \beta^{\text{eff}} + \dot{\rho}(t)}{\beta^{\text{eff}} - \rho_s - \dot{\rho} \Delta t} = \frac{C}{\tau - \Delta t} , \quad (\text{B.6})$$

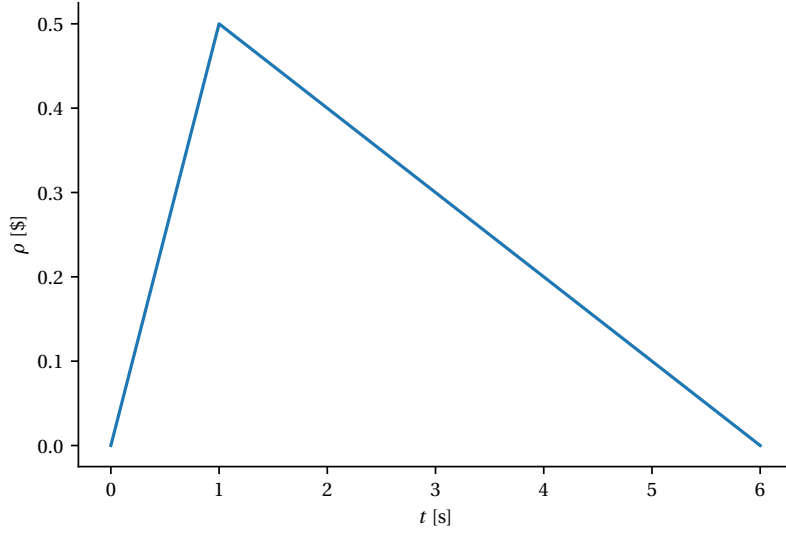


Figure B.1: Asymmetric Reactivity Ramp

where $\tau = \frac{\beta^{\text{eff}} - \rho_s}{\dot{\rho}}$ and $C = \frac{\lambda \beta^{\text{eff}}}{\dot{\rho}} + 1$. Now integrate Eq. (B.6) over the range $t_s \leq t' \leq t$. The left-hand side yields

$$\int_{t_s}^t \frac{\dot{p}(t')}{p(t')} dt' = \int_{p(t_s)}^{p(t)} \frac{1}{p(t')} dp(t') = \ln \frac{p(t)}{p(t_s)}. \quad (\text{B.7})$$

The right-hand side yields

$$\int_0^{\Delta t} \left(\frac{C}{\tau - \Delta t'} \right) d\Delta t' = -C \ln \frac{\tau - \Delta t}{\tau}. \quad (\text{B.8})$$

Substituting Eqs. (B.7) and (B.8) into Eq. (B.6) and solving for $p(t)$ gives

$$p(t) = p(t_s) e^{-\lambda \Delta t} \left(\frac{\tau}{\tau - \Delta t} \right)^{-C}. \quad (\text{B.9})$$

If $\dot{\rho} = 0$ and $\rho_s \neq \beta^{\text{eff}}$, Eq. (B.5) becomes

$$\frac{\dot{p}(t)}{p(t)} = \frac{\lambda \rho_s}{\beta^{\text{eff}} - \rho_s}. \quad (\text{B.10})$$

Solving for $p(t)$ gives

$$p(t) = p(t_s) e^{\frac{\lambda \rho_s \Delta t}{\beta^{\text{eff}} - \rho_s}}, \quad (\text{B.11})$$

and the final solution is then

$$p(t) = \begin{cases} p(t_s) e^{-\lambda \Delta t} \left(\frac{\tau}{\tau - \Delta t} \right)^{-C}, & \dot{\rho} \neq 0, \\ p(t_s) e^{\frac{\lambda \rho_s \Delta t}{\beta^{\text{eff}} - \rho_s}}, & \dot{\rho} = 0. \end{cases} \quad (\text{B.12})$$